Optimization of Cross-Layer Network Data based on Multimedia Application Requirements

by

Tasnim Rahman

A Thesis
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Master of Science
in
Electrical and Computer Engineering
by

August 2019

APPROVED:

Professor Alexander M. Wyglinski, WPI, Advisor

Professor Donald Richard Brown, WPI, Committee Member

Dr. Scott Pudlewski, AFRL, Committee Member

Abstract

This thesis proposes a convex network utility maximization (NUM) problem that can be solved to optimize a cross-layer network based on user and system defined requirements for quality and link capacity of multimedia applications. The problem can also be converged to a distributed solution using dual decomposition. Current techniques do not address the changing system's requirements for the network in addition to the user's requirements for an application when optimizing a cross-layer network, but rather focus on optimizing a dynamic network to conform to a real-time application or for a specific performance. Optimizing the cross-layer network for the changing system and user requirements allows a more accurate optimization of the overall cross-layer network of any given multi-node, ad-hoc wireless application for data transmission quality and link capacity to meet overall mission demands.

Acknowledgements

I would like to express my deepest gratitude to Professor Alexander M. Wyglinski for giving me the opportunity to work on this thesis and for advising me through it. He was supportive in every way and always knew how to assist me whenever I ran into any issues. I thank him for believing in me and for teaching me how to effectively conduct research and write about it. I also want to thank Dr. Scott Pudlewski from the Air Force Research Laboratory (AFRL) who provided a lot of guidance throughout the research process. He took his time to help me understand many of the mathematical concepts that were used in this thesis and gave me resources to use. Many thanks to Ulysses Lee from AFRL who assisted me with reviewing this document and for also teaching me many mathematical concepts. He was very helpful whenever I needed someone to review my derived equations. I would like to show my appreciation to Dr. Michael Medley and Paul Gilgallon from AFRL for reviewing my paper and for all of the emotional support. Finally, thanks to Professor Donald Richard Brown for being on my M.S. Thesis Committee and for advising me to approach Professor Wyglinski for this research endeavor.

Contents

List of Figures

List of Tables

List of Acronyms

AFRL	Air Force Research Laboratory
APA	Adaptive Power Allocation
C-DMRC	Compressive Distortion Minimizing Rate Control
CS	Compressed Sensing
CORE	Common Open Research Emulator
DARPA	Defense Advanced Research Projects Agency
DSA	Dynamic Subcarrier Assignment
EMANE	Extendable Mobile Ad-Hoc Network Emulator
GNUM	Generalized Network Utility Maximization
HEVC	High-Efficiency Video Coder
IP	Internet Protocol
LLC	Logical Link Control
MAC	Media Access Control
MANET	Mobile Ad-hoc Networks
NASA	National Aeronautics and Space Administration
NUM	Network Utility Maximization

NRL	Navy Research Laboratory
OFDM	Orthogonal Frequency-Division Multiplexing
ODG	Objective Difference Grade
OSI	Open Source Information
PEAQ	Perceptual Evaluation of Audio Quality
PSNR	Peak Signal-to-Noise Ratio
ROSA	Routing and Dynamic Spectrum-Allocation
SSIM	Subjective Similarity Index
TCP	Transmission Control Protocol
UAV	Unmanned Aerial Vehicle

Chapter 1

Introduction

1.1 Motivation

With the current research and development in coordinated multi-node network technology, it is important that the network for these applications serve the demands accordingly. The independent devices function within a dynamic and cross-layer network consists of multiple nodes that communicate with each other are often constrained in the operating environment. One example of a multi-node network is the swarm network used in autonomous applications [1]. Figure 1.1 displays a multi-node swarm network where the yellow nodes are mobile, while the orange lines indicate links, which are connections for data transmission between the nodes.

According to NASA, the goal of swarm autonomy capabilities is to "achieve better mission performance, robustness and flexibility at a lower cost" compared to a single system [1] [2]. Additionally, the Defense Advanced Research Projects Agency's (DARPA) has stated that "a priority for DARPA's unmanned systems research is to advance future swarming capabilities, allowing warfighters to outmaneuver adversaries in complex urban environment" [3]. A common application for swarm autonomy in the Air Force that would benefit greatly from the development of nulti-node networking and communications are unmanned aerial vehicles (UAVs), commonly referred to as drones, that operate as a collective team under specific mission demands.

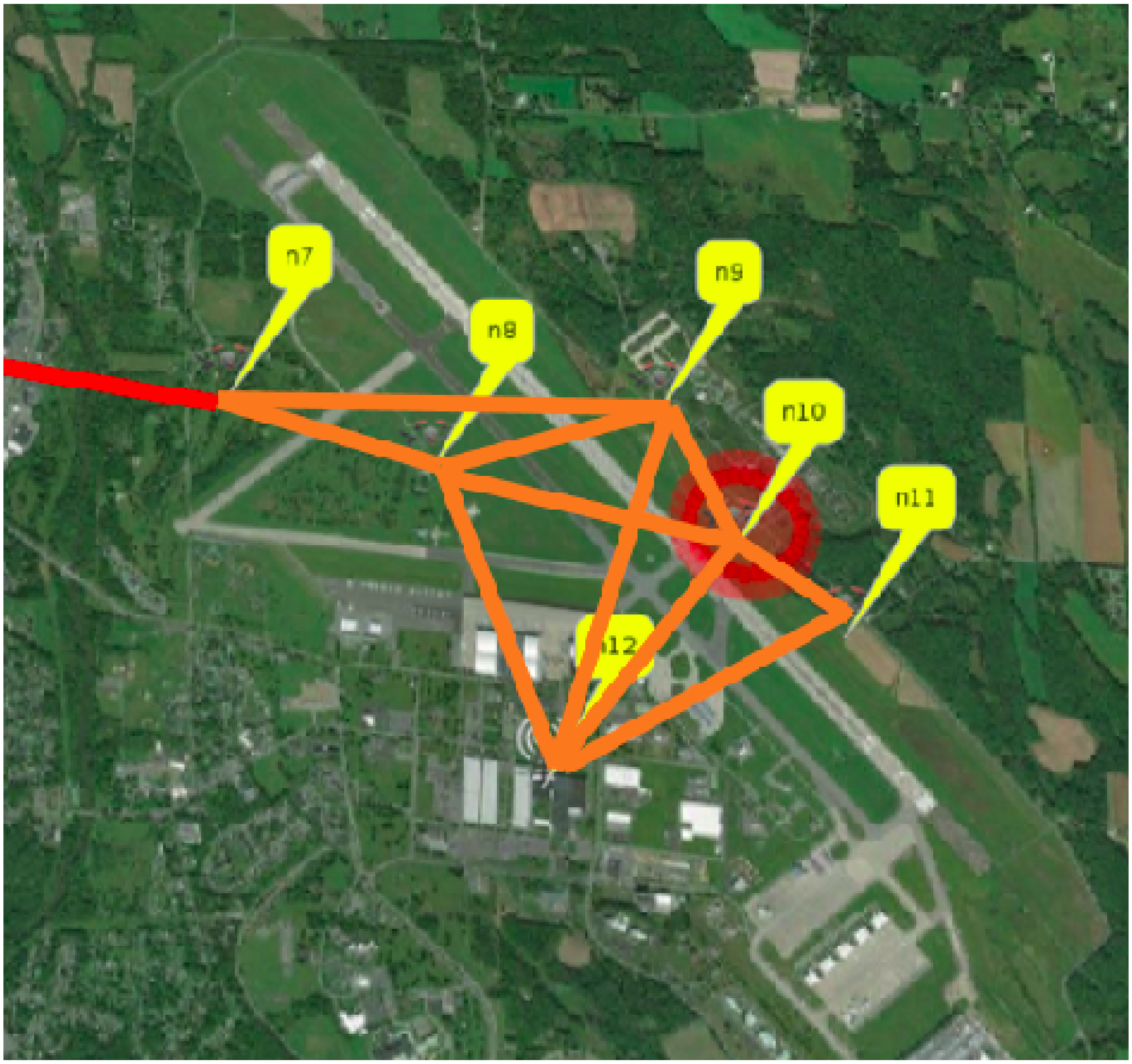

Figure 1.1: Air Force Environment with a Swarm Network that has Ad-Hoc Nodes which Transmit Data over Links

A multi-node cross-layer network application consists of a communications payload and data compression for transmission. Additionally, the application must take into account the concurrent mission demands involved in interactive network optimization. These kinds of applications do not have a centralized controller, which means the communication transmission comes with specific tradeoffs, also termed as a prices, for some variable (*i.e.*; quality, latency, throughput, etc.) [4]. Therefore, the network must be optimized at each node according to the constraints and requirements specified for the application. In other words, a dynamic network should have a domain-agnostic utility [5] that can be modified according to any factors and should be considered within the network utility.

Most recent developments, which will be discussed in Section 1.2, focus on optimizing routing and resource allocation or assignment for a dynamic network in order to improve overall performance, increase efficiency, and/or reduce power. Any multi-node network application, such as a swarm application, should take into account not only what the cross-layer network can accommodate, but also the overall mission-demands on the application. For instance, a swarm of UAVs may communicate with one another using different media in a dynamic, multi-hop network in order to complete a task. The task can vary from delivering a simple text message, audio message, or entire video clip from one end of the network to the other through the UAV nodes within this scenario. Data is transmitted from node to node through the links. To complete the specified task, not only should the dynamic network capabilities of the system be considered, but the requirements for the message being transmitted by the user should be considered as well. This is normally defined by a user who is sending the message through the application. This thesis will focus on optimizing a cross-layer network based on user-defined and network system-defined requirements for multimedia applications (*i.e.*; text, audio and video).

1.2 State of the Art

Recent developments in dynamic or cross-layer network optimization have been focused on mobile ad-hoc networks (MANETs) [6] [7] as well as the emerging 5G networks [8]. Optimization of these types of networks are usually geared towards specific applications or implementations. For instance, the overall network path for aeronautical communications for intra-flight ad-hoc networks has been optimized to improve concurrent flows while satisfying bandwidth and delay requirements [7]. A Wireless Sensor Networks (WSN) has been optimized for energy-efficiency and power reduction [9], while wireless ad-hoc networks have been optimized for actual deployments and routing paths [10]. These type of optimizations can be solved using protocol models and network simulators or through algorithmic models. One can optimize a cross-layer network is through the building of a network architecture possessing design requirements. A next-generation mobile tactical network possesses many requirements that can be optimized by creating an architecture with design requirements for autonomy, scalability, and adaptability [11]. However, a network architecture will still need to be optimized for performance and efficiency after it is designed. This is achieved using techniques such as utility function maximization, which is used for cross-layer network optimization in this thesis.

Cross-layer optimization can be achieved by maximizing a network utility function relating different conditions. The utility function has been derived from an resources in economics [12]. The utility function uses the price of variables in order to weigh in on the overall utility of the model. A utility function attempts to distribute resources equally and fairly. Additionally, it considers alternatives for the overall system. For example, a utility function can be used to optimize down-link scheduling in a Orthogonal Frequency-Division Multiplexing (OFDM) network via dynamic subcarrier assignment (DSA) algorithm and adaptive power allocation (APA) algorithm to solve the function [13].

Currently, most of the the work and accomplishments for cross-layer network optimization are tailored to maximize a utility function of a network over one or more constraints. Based on the type of network, the cross-layer optimization approach will choose requirements and parameters to optimize the network utility function. One type of dynamic network studied within the context of cross-layer optimization are ad-hoc wireless networks‘ [4]. In these types of networks, the network utility function has been optimized for real-time applications by controlling congestion and contention as well as meeting delay requirements in [14]. Furthermore, simple network utility functions based solely on the data rate for an ad-hoc network can be maximized for flow and link capacity constraints based on Quality-of-Service (QoS) classes [15].

All of the aforementioned research into cross-layer network optimization focused on maximizing a dynamic network for a specific performance and efficiency constraints. They have focused on creating a cross-layer network that can conform to a real-time application or to a specific performance without considering application or performance based requirements. However, for a dynamic network with many nodes, the application might need different requirements based on users and systems needs. The cross-layer network optimization approach should consider these requirements in the network utility function in order to maximize its constraints. This thesis proposes a methodology for a dynamic network to be optimized according to users and systems (networks) constraints for an application to meet mission demands.

1.3 Thesis Contribution

This thesis contributes to the enhancement of multi-node and multimedia dynamic networks that do not have a centralized controller. The thesis provides a methodology to optimize network data transmission based on user and system (network) requirements for media options and media quality in an application. The solution to the cross-layer optimization problem provides an optimal data rate that the media method should use to transmit data. This research makes the following contributions:

- **Used multimedia quality models to determine user and system requirements for a cross-layer network application to meet mission demands.** The thesis introduces multimedia quality models that are existing or modified in order to determine the quality of text, audio, and video transmission. The user and system application requirements for media quality will use values from these quality models to input into the cross-layer network optimization problem. Additionally, data is transferred through text, audio, and video codecs. Therefore, the media quality is based off of media transmission using the codecs.

- **Network Utility Maximization (NUM) problem is used to create the optimization problem.** The user and system application requirements derived from the multimedia quality models are input into the network utility functions that are a part of the NUM problem for this research. The NUM problem is a function of data rate and is composed of utility functions for text, audio, and video with a link capacity constraint. The utility functions are maximized to solve for the optimal data rate to transfer data over a link based on the input requirements.

- **Network optimization problem can be converged to a distributed solution using dual decomposition.** A distributed solution to the cross-layer optimization problem can be solved using the dual decomposition method that converges the optimization problem into a distributed solution for the optimal data rate.

Multi-node dynamic networks are used in the following types of applications in communications and networking [16]:

- bluetooth networks
- universal mobile telecommunication system networks
- wireless networks
- serverless chat

Additionally, the application considers multiple media options such as image and video applications [16]. These types of applications include:

- pedestrian detection and tracking
- character recognition
- object tracking
- traffic incident detection

Finally, sensor networks can utilize this research for wireless sensor networks, ad-hoc sensor networks, and swarm-based sensor networks. This thesis is also important for security and military applications for intrusion and incident detection. A specific Air Force scenario will be introduced for the implementation of the research from this thesis in Chapter 4.

1.4 Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 will present an overview of cross-layer network optimization. It will also discuss research and developments relevant to this thesis in detail. Chapter 3 will present the methodology of cross-layer network optimization based on user and system defined application requirements for data transmission quality and link capacity for text, audio, and video. Chapter 4 will describe an implementation of the cross-layer network optimization method from this thesis. Finally, Chapter 5 will conclude this thesis with a summary of the work and discuss any future work that should be considered and developed from this thesis.

Chapter 2

Overview of Cross-Layer Network Data Optimization

In this chapter we discuss a background overview of cross-layer network data optimization based on application requirements for quality and link capacity. To understand the motivation for this type of dynamic network optimization, fundamentals of cross-layer network optimization will be given in Section 2.1, including the network utility maximization (NUM) problem. In order to optimize a cross-layer network for the quality of the data transmission and capacity at a link for a media option, the NUM problem is defined in Section 2.1.1. Realistically, any one of the three media options, text, audio or video, can be used at any link. Therefore, this thesis will focus on each media method individually in terms of their network utility functions as well as combining them into one utility function in order to create an overall problem. The network utility is based on data rates or bit rates and takes the user and system defined application requirements for quality into account. Each media option uses a quality model in order to define the application requirements, which are detailed in Section 2.2. Finally, the solutions to the NUM problem are converged to a single solution using the dual decomposition method, defined in Section 2.4. Any related work that was used and improved upon for this thesis will be discussed in Section 2.5. Section 2.6 will describe the objectives of the research and Section 2.7 will conclude this chapter with a summary.

2.1 Cross-Layer Network Optimization

A cross-layer network allows the joint consideration of performance and functionality between network layers in the Open Source Information (OSI) model or transmission control protocol/internet protocol (TCP/IP) model, increasing efficiency and adaptability [17]. Networks vary with time and are prone to errors. Cross-layer optimization allows performance improvement of applications that use error-prone and time-varying networks across the network layers [18]. It inputs requirements and optimizes the network across all of the layers in the OSI model or TCP/IP model to maximize the overall dynamic network for some variable. Figure 2.1 gives a diagram of how a cross-layer optimization problem is implemented on the OSI network layers. A dynamic network application for this thesis would apply the optimization requirements across each layer to optimize all the layers as a whole and maximize it for some variable or performance. The OSI model includes the following layers from low to high: physical, data link, network, transport, session, presentation, and application [19]. Data is received from the lower layers to the higher layers and data is transmitted from the higher layers to the lower layers. The physical layer is the lowest layer of the OSI model that manages reception and transmission of raw data over a physical medium to the higher layers (*i.e.*; electrical/optical, mechanical, and functional interfaces). The data link layer has two sublkayers: the logical link control (LLC) and the media access control (MAC). The LLC sublayer manages error-free transfer of data frames from node to node. It establishes and terminates logical links, controls frame flow, sequences frames, acknowledges frames, and retransmits these frames. The MAC sublayer manages the connection to the physical layer, checks for errors, and manages the recognition of addresses for received frames. The network layer controls the operation of the subnet and determines the physical path that data should take based on network conditions, priority of service, routing, traffic control, frame fragmentation, reassembly, address mapping, usage accounting, and other factors. The transport layer makes sure that messages are delivered error-free without loss or duplication and in sequence. The session layer creates a virtual connection, or session, between two nodes in a network to establish, manage, and terminate dialog or data transmission [20]. The presentation layer manages how the network application formats the data that is sent. It allows the application to read the message through encryption and decryption, formatting, and translation. Finally, the highest layer in the OSI model is the application layer which provides an interface for the end user operating the device that is connected to the network and it supports data file transfers and messaging.

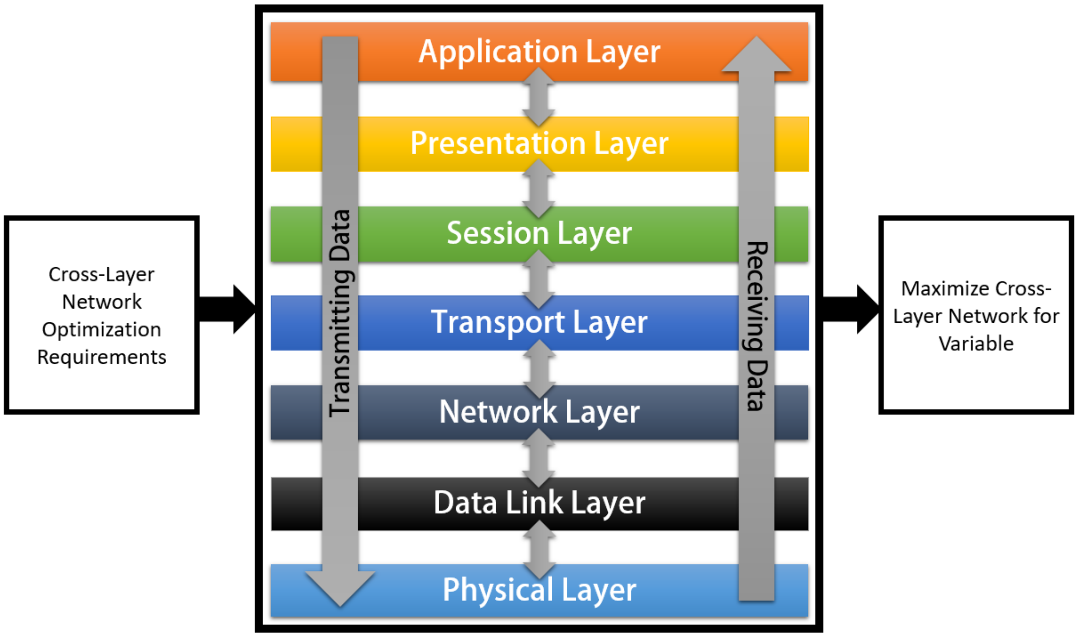

Figure 2.1: Cross-Layer Network Optimization on the OSI Layers

Another open standard networking model that can network requirements can be optimized over is the TCP/IP model that consists of the following layers from low to high [21]: link, internet, transport, and application. Figure 2.2 gives a diagram of how a cross-layer optimization problem is implemented on the TCP/IP network layers. Data is received from the lower layers to the higher layers and data is transmitted from the higher layers to the lower layers. The link layer, or network interface layer, is the lowest layer of the TCP/IP model that combines the physical and data link layer functions from the OSI model such as modulation, line coding and bit synchronization, frame synchronization and error detection, and LLC and MAC layer functions. The internet layer is related to the network layer of the OSI model and manages traffic routing, traffic control, fragmentation, and logical addressing. The transport layer is directly related to the transport layer of the OSI model and manages message segmentation, acknowledgement, traffic control, session multiplexing, error detections and retransmission, and message ordering. The application layer is the highest layer of the TCP/IP model which is relates to the session, presentation, and application layers of the OSI model which manages session establishment, maintenance and termination, character code translations, data conversion, compression and encryption, and other functions.

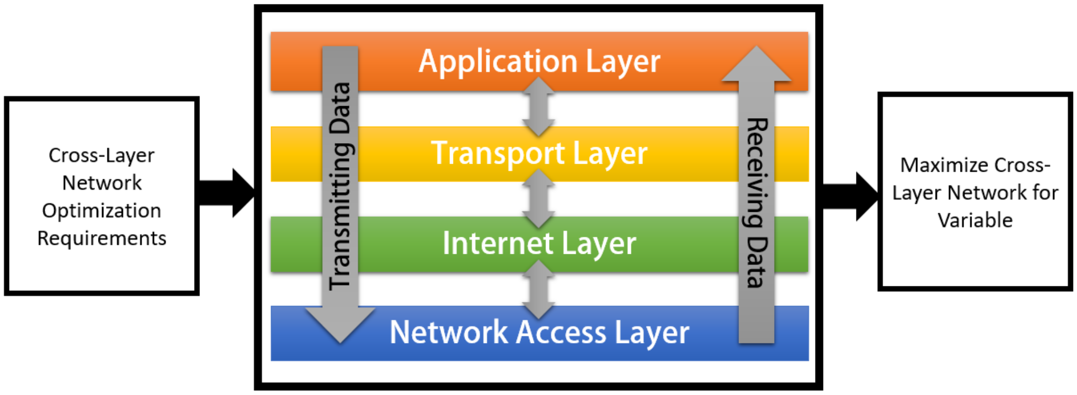

Figure 2.2: Cross-Layer Network Optimization on the TCP/IP Layers

Generally, a cross-layer network is optimized for networks that contain independent nodes with no centralized controller [14]. Figure 2.3 shows how a network architecture is generally designed with and without a centralized controller.

These type of networks are wireless networks, such as ad-hoc networks, MANET networks, and swarm networks. Cross-layer optimization is important for wireless networks since the nodes may be mobile and its functionality then applies to multiple network layers. Additionally, these modern communication networks accommodate multiple users, services, and applications as well as different system (network) requirements [22]. Optimizing networks for these requirements to increase performance, efficiency, quality, or any other variable can be achieved through different algorithms. For this study, a NUM problem was investigated to optimize the quality and link capacity in a cross-layer network based on user and system requirements. A convergent solution to this NUM can be obtained through the

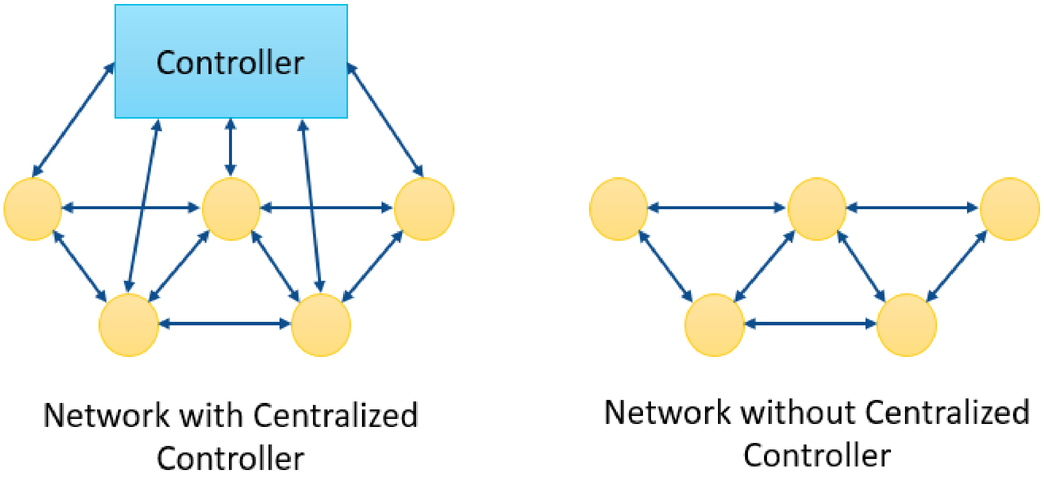
Controller
Network with Centralized Controller
Network without Centralized Controller

Figure 2.3: Basic Network Architecture Design With and Without a Centralized Controller

dual decomposition method.

2.1.1 Network Utility Maximization Problem

The idea of proportional fairness in a communication network is one of the keys to understanding the NUM problem. The utility proportional fairness policy was introduced in [23] according to which resources must be allocated in a fair manner among different applications to lead to the calculation of closed form solutions for the NUM problem. Figure 2.4 shows a basic ideas of how contrained resources are allocated fairly among the nodes in a multi-node network.

An optimal solution thus is achieved when the users requirements for some choice of price or rate and the network (system) requirements for some choice of price or rate are at an equilibrium for a NUM problem through proportional fairness [24]. Essentially, a NUM problem is a control algorithm used to analyse and achieve stability and fairness in a network. In this thesis, it is used for cross-layer optimization to achieve stability and fairness for quality and link capacity using user and system defined requirements for quality prices of transmission data (*i.e.* text, audio and video). The prices are determined from quality models that are introduced later in this thesis.

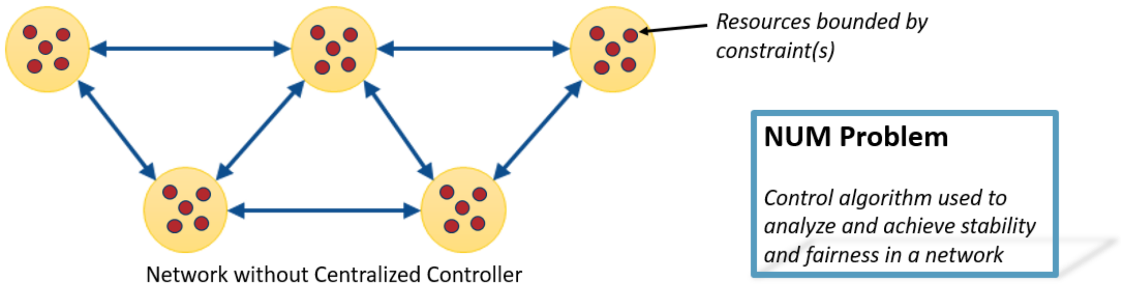
Resources bounded by constraint(s)
NUM Problem
Control algorithm used to analyze and achieve stability and fairness in a network
Network without Centralized Controller

Figure 2.4: Basic Network Multi-Node Architecture with NUM Optimization for Resource Allocation

When formulating a cross-layer optimization problem, a network utility function is derived from design requirements. The utility function is then maximized subject to the variable or variables in order to improve the network performance. A generalized network utility maximization problem (GNUM) is defined as [25]:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{s} U_s(x_s, P_{e,s}) + \sum_{j} V_j(w_j), \\
\text{subject to} \quad & \mathbf{Rx} \leq \mathbf{c}(\mathbf{w}, \mathbf{R_e}) \\
& \mathbf{x} \in \mathcal{C}_{\mathbf{1}}(\mathbf{P_e}), \mathbf{x} \in \mathcal{C}_{\mathbf{2}}(\mathbf{F}) \text{ or } \mathbf{\Pi}(\mathbf{w}), \\
& \mathbf{R} \in \mathcal{R}, (\mathbf{F}) \in \mathcal{F}, (\mathbf{w}) \in (\mathbf{W}).
\end{aligned}
\tag{2.1}
$$

This GNUM problem optimizes the variables $\mathbf{x}, \mathbf{w}, \mathbf{P_e}, \mathbf{R}$, and $\mathbf{F}$. $U_s(x_s, P_{e,s})$ is a utility function for node or source s that is dependent on the source rate, or data rate, x_s and decoding error probabilities $P_{e,s}$. $V_j(w_j)$ is another utility function for network element j that is dependent on the physical layer resources w_j. R is the routing matrix and $\mathbf{c}$ are the logical link capacities as functions of physical layer resources $\mathbf{w}$ and the desired decoding error probabilities P_e. $\mathbf{x}$ is the source rate vector that is greater than 0. The source rate vector is constrained by the set $\mathcal{C}_1(\mathbf{P_e})$, which is the interplay between channel decoding reliability that takes account of the rate-reliability tradeoff and coding issue. The source rate vector is also constrained by $\mathcal{C}_2(\mathbf{F})$, which is the medium access success probability where $\mathbf{F}$ is the contention matrix. $\mathcal{C}_2(\mathbf{F})$ is also generally known as the schedulability constraint set, $\mathbf{\Pi}$. This constraint captures the issue of MAC random access or scheduling. For this thesis, the utility functions are concave and can be solved using convex optimization for a practical solution. Finally, $\mathcal{R}, \mathcal{F}$, and $\mathcal{W}$ are respectively the sets of single-path or multipath routing schemes, possible scheduling or contention-based medium access schemes, and the sents of possible physical later resource allocation schemes that constrain $\mathbf{R}, \mathbf{F}$, and $\mathbf{w}$. The constraints of the GNUM problem are based on both system-defined constraints as well as the physical constraints of the network. The first constraint states that the data sent through each link which is defined by link matrix $\mathbf{R}$ multiplied by the rate vector $\mathbf{x}$, is less than the capacity $\mathbf{c}$ of the links. In this formulation, $\mathbf{c}(\mathbf{w}, \mathbf{R_e})$ is dependent on the physical layer resources and the error probability.

The GNUM problem can be modified by holding some variables constant and specifying some of the dependencies to create a required NUM problem for optimizing a particular network. For this thesis, there will be user experience and system-defined requirements or constraints in the NUM problem. If the NUM problem is convex then it can be solved efficiently [26] since the local minumum is the global minimum in these types of problems [25].

The network has l links and each link has a set of $S(l)$ sources or nodes. They each transmit at a data rate of x_s bits per second. x_s is in the set of positive data rates, $\mathcal{X}$, between 0 to 50,000 kilobits per second. Each link has a link capacity of c that corresponds to the media option being used, whether it is audio, video, or text, and its maximum data rate, or bit rate, that can be transmitted. A node has a utility function of $U_s(x_s)$ [27]. Each node transmits a flow possessing an assigned price that results from both a system-defined and user-defined parameters for its utility function. The Network Utility Maximization (NUM) problem that will be optimized for the cross-layer network at a link is defined as follows:

$$\begin{aligned} \text{maximize} \quad & \sum_s U_s(x_s) \\ \text{subject to} \quad & \mathbf{x_s} \leq \mathbf{c}, \\ & \mathbf{x_s} \in \mathcal{X} \text{ where } \mathcal{X} \in [\mathbf{0}, \mathbf{50000}] \text{ kilobits per second} \end{aligned} \tag{2.2}$$

The NUM problem will depend on the media method being used for the network application. The network utility function varies for each media option, but they each follow a sigmoidal function model [28]. A sigmoidal function is either convex, concave or both [29]. Note that a concave function problem can still be solved as a convex function problem since a concave function is a reflection of a convex function. In other words, a concave function is the negative of a convex function.

The network utility functions for text, audio, and video in this thesis are concave at the bounds of low and high data rates and convex in between. The reason for this change from concavity to convexity to concavity within a function is so that the utility can model a real-time model of the network [30]. In a real-time model, the media application adapts to the data rate and should adjust accordingly to maintain its quality. This type of model requires a minimum and maximum level of performance guarantee for the data rates. Below the minimum level and above the maximum level, the quality of the media application will become unnacceptable. Additionally, the network utility follows the Friedman-Savage utility function which defines the utility function as being a sum of two convex sigmoidal functions [31] [32]. Since the network utility is concave at low and high bounds and convex in between, it forms an ”s-shape” function [29] that can be modeled by the logisitic function defined as [33]:

$$f(x) = \mathcal{L}/(1 + e^{k(x-x_0)}) \tag{2.3}$$

where $\mathcal{L}$ is the curve's maximum value, $\|$ is the logistic growth rate or steepness of the curve, x is the input to the function and x_0 is the x-value of the sigmoid's midpoint. In the case of a utility function, the curve's maximum value is 1 and the input to the function is the data rate x_s. The sigmoid's midpoint and logistic growth, however, will vary for the text, audio, and video utility functions. The sigmoid's midpoint is the difference between the upper data rate threshold and the lower data rate threshold defined for each media. The logistic growth can be solved through a system of equations for the text, audio, and video utility functions. At the lower data rate bound, the utility function will be approximately equal to 0 or 0.001, x_s is equal to the lower data rate threshold, and the logistic growth is unknown. At the upper data rate bound, the utility function will be approximately equal to 1 or 0.999 , x_s is equal to the upper data rate threshold, and the logistic growth is unknown. Setting the utility function at the lower data rate bound equal to the upper data rate bound and solving for the logistic growth gives the steepness of the curve for the specific media's utility function.

Text is an elastic application and is not delay-sensitive [30]. This means that the utility function increases with an increase in data rate. However, the marginal improvement decreases as the data rate increases. Additionally, an elastic application generally has no maximum data rate that affects its performance since it can accomodate any delay. However, the utility function for text in this thesis has a maximum data rate threshold of 32 kilobits per second since that is the minimum data rate threshold for audio [34]. Theefore, the utility function is valid between 0 and 32 kilobits per second, which correspond to the minimum and maximum data rate thresholds. In the case of this cross-layer network, the elastic utility function is modeled by Equation (2.4). Figure 2.5 gives the model of the utility function for text. $U_T(x_s)$ is the utility function for text as a function of the data rate x_s. The logistic growth or steepness of the curve is 2.3748×10^{-4}.

$$\begin{aligned} &U_T(x_s) = 1/(1 + e^{-(2.3748\times10^{-4})\times x_s}) \\ &\text{where } x_s \in [0, 32] \text{ kilobits per second} \end{aligned} \tag{2.4}$$

On the other hand, audio and video are inelastic applications and are delay-sensitive applications and vary depending on the network load and data rates [30]. In the case of a cross-layer network, the utility function model has data rate requirements depend on audio and video data rate thresholds which are independent of network congestion. For this thesis, the network uses MP3 [34] for the audio codec and H.265 [35] for the video codec. The maximum data rate threshold for audio is 320 kilobits per second and the maximum data rate threshold for video is 50,000 kilobits per second.

For audio, the utility function would be approximately zero for a lower data rate threshold of 32 kilobits per second. At the upper data rate threshold of 320 kilobits per second, the utility function approaches unity [34] . $U_A(x_s)$ is the utility for audio as a function of the data rate x_s. The logisitic growth or steepness of the curve is 2.6387×10^{-5} and the sigmoid's midpoint is 288 kilobits per second. The utility function for audio is given as:

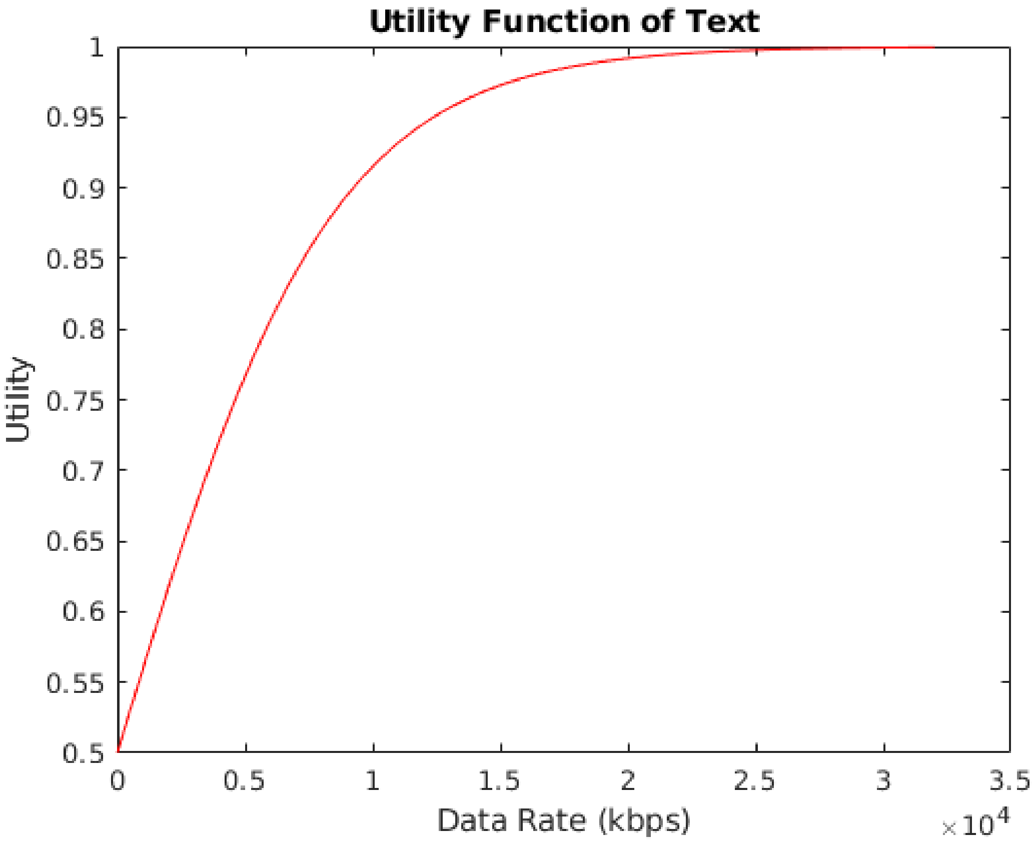

Figure 2.5: Model of Text Utility Function

$$U_A(x_s) = 1/(1 + (e^{-(2.6387\times10^{-5})(x_s-288000)})) \quad \text{where } x_s \in [32, 320] \text{ kilobits per second} \tag{2.5}$$

And Figure 2.6 gives the model of the utility function for audio.

For video, the utility function is approximately 0 at the lower data rate threshold of 12,000 kilobits per second. For the upper data rate threshold of 50,000 kilobits per second, the utility function approaches unity [35]. $U_V(x_s)$ is the utility for video as a function of the data rate x_s. The logistic growth or steepness of the curve is 1.9998×10^{-7} and the sigmoid's midpoint is 38,000 kilobits per second. The utility function for video is given as:

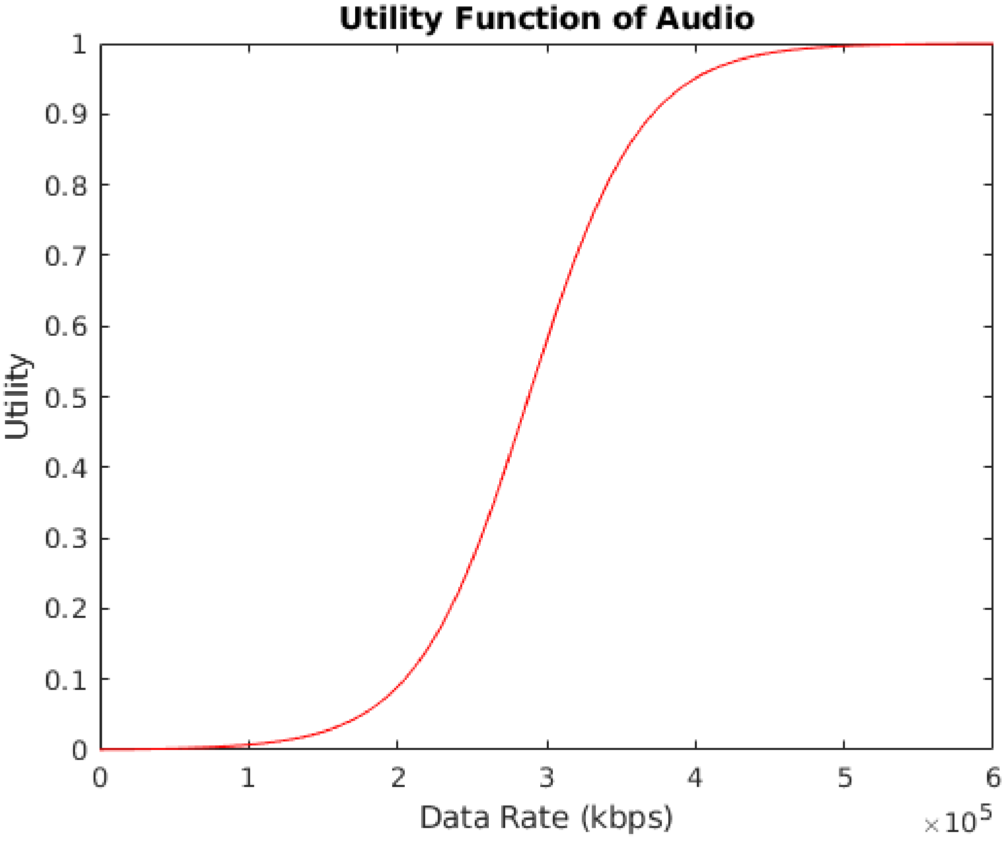

Figure 2.6: Model of Audio Utility Function

$$U_V(x_s) = 1/(1 + (e^{-(1.9998\times 10^{-7})(x_s - 38000000)})) \qquad (2.6)$$
$$\text{where } x_s \in [12000, 50000] \text{ kilobits per second}$$

And Figure 2.7 gives the model of the utility function for video.

The sigmoidal utility functions have to be combined into one function that can be used to determine the utility of this cross-layer network design for a flow from a link. A way to model this is with the following utility function:

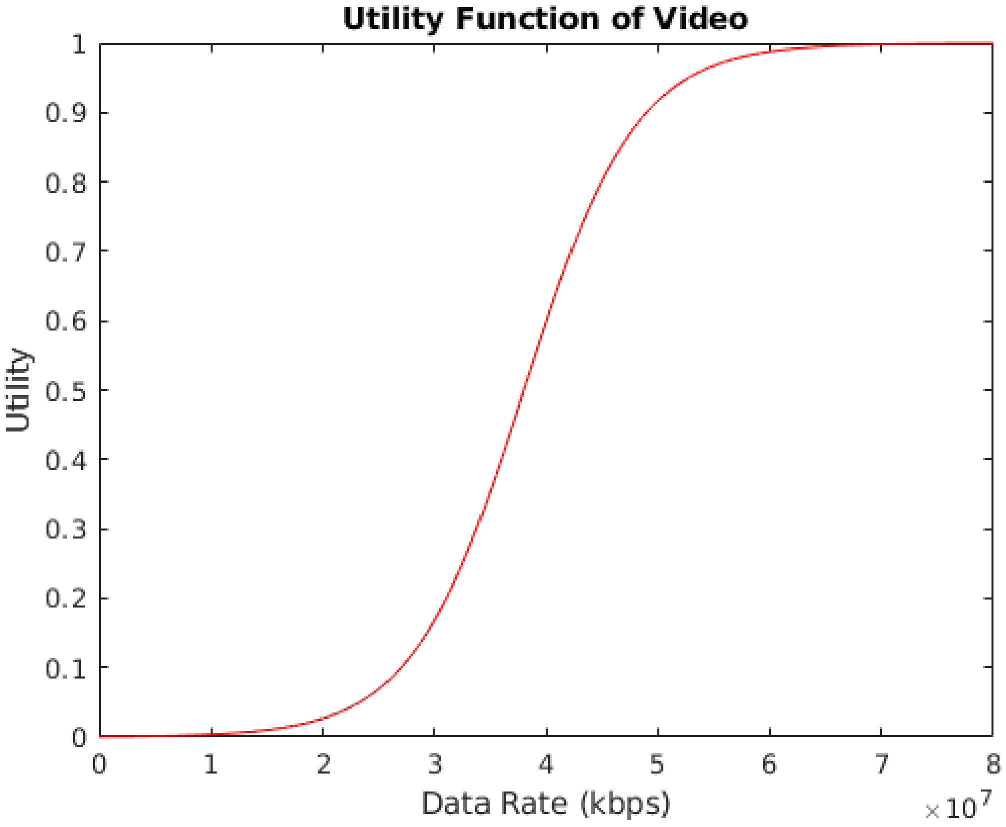

Figure 2.7: Model of Video Utility Function

$$\begin{aligned} &U_S(x_S) = (I_T \times P_T \times U_T) + (I_A \times P_A \times U_A) + (I_V \times P_V \times U_V) \\ &\text{ where } P_i \in [0,1],\, I_i \in [0,1], \\ &\sum P_i \leq 1 \text{ and } \sum I_i \leq 1 \end{aligned} \tag{2.7}$$

The parameters I_i and P_i are the user and system defined parameters respectively that correlate to the user and system requirements for quality of the media. The user defined quality parameter is what the user of the application requires for the cross-layer network based on mission-demand while the system defined quality parameter is what the cross-layer network requires for the application based on what the network is capable of for the mission-demand. The parameters I_T and P_T specify the for text quality, I_A and P_A are parameters for audio quality and I_V and P_V are parameters for video quality. Section 2.2 will describe how these parameters are determined from the user's point of view and the system's point of view.

After the user and system defined parameters are known, the universal utility function can be determined. The next step is to optimize or maximize the universal utility function for the cross-layer network using Equation (2.2). Another way to optimize this cross-layer network optimization based on application requirements is to converge the problem to a distributed solution through the dual decomposition method described in Section 2.4. It is also possible that the NUM problem has only one optimal solution for the data rate and does not need to be converged. In that case, this final step can be skipped.

2.2 Application Quality for Text, Audio and Video

The quality of text, audio and video are used for the user and system defined parameters in the universal utility function for the cross-layer network. They each derive their value from a normalized quality model.

Text is an elastic application and it is either transmitted or not. Therefore, there is no varying quality. Text quality for the user parameter is I_T while for the system parameter it is P_T. These parameters can either be a 0 for text that is not transmitted or 1 for text that is transmitted. This means that if the user and system indicates the text as being transmitted, the text quality parameter, I_T and P_T will equal to 1. If the user and system indicate the text as not being transmitted, the text quality parameters I_T and P_T, will equal to 0.

For the purposes of this thesis, it is important that the audio quality scale used is a normalized scale between 0 and 1 in order to comply with the universal utility function. Conventional audio quality scales are not normalized to this range, but one can be derived from the Perceived Audio Quality (PEAQ) Algorithm [36]. The PEAQ Algorithm gives an audio sample an objective difference grade (ODG) value between -4.0 to 0.0 to define the sound quality. The audio sample is graded according to its distorted noise: -4.0 means the audio sample is very annoying, -3.0 is annoying, -2.0 is slightly annoying, -1.0 is perceptible, but not annoying and 0.0 is imperceptible [37] [38]. A very annoying audio sample has significant distortion and is the worst quality. An audio sample with imperceptible noise has no distortion and is the best quality. Table 2.1 shows the ODG value scale for the PEAQ Algorithm.

Table 2.1: Audio Distortion vs. Objective Difference Grade (ODG) [38]

Audio Distortion vs. Objective Difference Grade (ODG)	
Audio Distortion	Objective Difference Grade (ODG)
Imperceptible	0.0
Perceptible, but not annoying	-1.0
Slightly annoying	-2.0
Annoying	-3.0
Very annoying	-4.0

In a Normalized Audio Scale derived from the ODG scale of the PEAQ Algorithm, 0.2 is very annoying, 0.4 is annoying, 0.6 is slightly annoying, 0.8 is perceptible, but not annoying and 1.0 is imperceptible. Note that 0.0 on the Normalized Audio Scale means there is no audio sample at all and that is an option for the user and system defined parameters. The conversion between the ODG value and Normalized Audio value, which is called norm, is below in Equation (2.8).

$$\text{norm} = (\text{ODG} + 5.0)/5.0 \tag{2.8}$$

Table 2.2 shows the Normalized Audio Scale.

The user and system defined parameters that will be determined by the Normalized Audio Scale for the Universal Utility Function are I_A and P_A respectively. To determine the audio quality value, a MATLAB program was used. The program calculates the PEAQ algorithm's ODG value that is then converted to the Universal Audio Scale value using Equation (2.8). Additionally, the program takes an ideal audio sample as reference in order to calculate the quality of a chosen audio sample. For this thesis, an MP3 audio sample with

Table 2.2: Audio Distortion vs. Normalized Audio Scale

Audio Distortion vs. Normalized Audio Scale	
Audio Distortion	Normalized Audio Scale
Imperceptible	1.0
Perceptible, but not annoying	0.8
Slightly annoying	0.6
Annoying	0.4
Very annoying	0.2
No Audio	0.0

a bit rate of 256 kilobits per second was chosen as the reference sample. Section 2.3.2 will go into detail as to why an MP3 audio sample with a bit rate of 256 kilobits per second was chosen for reference. Additionally, the sampling rate accepted by the PEAQ Algorithm's MATLAB program is only 48,000 Hertz for any audio sample. Therefore, the reference audio sample has to have a sampling rate of 48,000 Hertz.

Similar to the audio scale, the quality scale for video also has to be normalized between 0 and 1. There are already existing scales that can be used and can comply with the universal utility function. The Structural Similarity (SSIM) Index can be used to measure video quality between 0 to 1, where 1 is the best quality video [39]. However, the Peak Signal to Noise Ratio (PSNR) is commonly used to determine video quality, therefore for the video quality results in Chapter 2.3, the PSNR values will also be shown in addition to the SSIM Index for video quality.

The user and system defined parameters that will be determined by the SSIM Index for the universal utility function are I_V and P_V, respectively. A MATLAB program was used to determine the SSIM Index and PSNR. The program calculates the SSIM and PSNR values by comparing the tested video sample frame to a reference video sample frame. All the video samples used for this thesis were of the H.265 or High-Efficiency Video Coding (HEVC) codec 2.3.3. The reference video sample had a bit rate of 50,000 kilobits per second at a frame rate of 30 frames per second, a sampling rate between 66,846,720 samples per second and 133,693,440 samples per second, and a resolution of 2048x1080. Section 2.3.3 will discuss why the H.265 codec was used and how the reference video sample was chosen [40]. Additionally, the video sample frames were selected through the FFmpeg platform that can save a selected frame from a video as a raw format that can be used in the MATLAB program to calculate the SSIM and PSNR values. It is important the frames are saved in a raw format in order to ensure the quality from the specified bit rate, frame rate, sampling rate, and resolution of the video sample does not differ from the original quality.

2.3 Codecs for Text, Audio and Video

In order to create text, audio, and video samples, a codec must be implemented. A codec encodes and decodes digital data for text, audio and video. For this thesis, the codecs were chosen according to what is commonly being used for applications today. The codec being used in this thesis for text is Huffman Coding, while the codec being used for audio is MP3, and the codec being used for video is H.265 or HEVC. All of the audio and video samples were configured using the FFmpeg platform in Linux. The following sections will discuss why these codecs were chosen and how they were implemented to create data samples to test for application quality.

2.3.1 Huffman Coding Text Codec

The codec being used for text is Huffman Coding, created by David A. Huffman [41]. It is often used for lossless data compression [42]. Huffman coding compresses information, such as a text message, in order to transmit over a network. It compresses or encodes the text into a binary message for transmission and then decompresses it or decodes it when it gets to the destination. The Huffman Table is used to encode each character in the text message according to their probability into a binary code. The probability for each character is determined through the Huffman Algorithm, which uses a binary tree [43].

Text does not have a data rate based quality and is rather graded on whether it is transmitted or not. Through Huffman Coding, the text is compressed to ensure transmission through a cross-layer network. Two MATLAB programs were used to perform Huffman

Coding. A text message is saved in a text file, which is then passed through the first MATLAB program with the Huffman Algorithm to encode or compress the text into a binary message which is saved in a text file. This text file with the binary message can then be decoded or decompressed through the second MATLAB program to a text message saved in a text file.

2.3.2 MP3 Audio Codec

The codec used for audio is MPEG-1/2 Layer-3 or MP3 [44]. MP3 is popular for audio data transmission because it is an open standard, has accessible encoders and decoders, and is supported by many technologies and applications. It also has efficient compression and is lossless. The sampling rate required for use due to the PEAQ Algorithm's MATLAB program is 48000 Hertz. Therefore, the MPEG-1 Layer-3 codec is being used. For this sampling rate, the bit rate can vary between 32 kilobits per second to 320 kilobits per second. The utility function for audio introduced earlier in this chapter uses this bit rate range for its bounds. Also, as mentioned before, 256 kilobits per second was used for the reference audio sample in order to determine quality of audio samples through the PEAQ Algorithm. A higher bit rate offers better quality sound since more bits are being transmitted per second. 256 kilobits per second provides CD quality audio for an MP3 audio sample. Therefore, it is used for the reference audio sample and all other audio samples will be tested against it to determine audio quality. Finally, all of the audio samples were created using the FFmpeg platform on Linux. A raw WAV audio sample named as 'track.wav' was downloaded from [45] in wav format. It was then converted to MP3 audio samples using FFmpeg for reference and testing.

2.3.3 H.265 Video Codec

The codec used for video is the H.265 codec or HEVC codec. This codec is an improvement of the compression performance of the H.264 codec with approximately a 50 percent bit rate reduction, but still maintains the same video quality [46]. The highest resolution that produced video samples without any interruptions on the Linux computer platform for this thesis was 2048x1080. The range of sampling rates supported by this resolution are 66,846,720 samples per second to 133,693,440 samples per second. Additionally, the bit rate range for this resolution is 12,000 kilobits per second to 50,000 kilobits per second. The frame rate used for the video samples were 30 frames per second. The reference video sample that the reference video sample frame was selected from to determine the SSIM and PSNR values through the MATLAB program had a bit rate of 50,000 kilobits per second. All of the other video sample frames produced were tested against this reference for their video quality. Finally, the FFmpeg platform was utilized to create the video samples and to select the frames to be tested from them. A raw MKV video sample named as 'video-h265.mkv' was downloaded from [47] in mkv format. It was then converted to HEVC video samples using FFmpeg for reference and testing.

2.4 Convergence through Dual Decomposition

Decomposition reduces a larger problem into subproblems that are coordinated by a master problem [25]. Figure 2.8 shows how the larger NUM problem can be decomposed into separate subproblems.

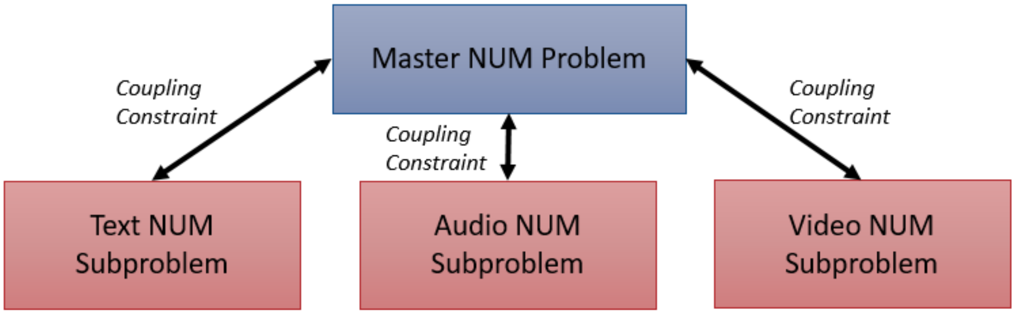

Figure 2.8: Decomposition of NUM Problem into Subproblems

The techniques used to perform decomposition are categorized as primal decomposition or dual decomposition. In primal decomposition methods, the master problem gives each subproblem an amount of resources it can use and properly allocates the resources. In dual decomposition methods, the master problem sets the price for resources to each subproblem and obtains the best pricing strategies. The subproblems then decide the amount of resources to use depending on the price. Dual decomposition is the best technique to use to solve and converge the convex network utility maximization (NUM) problem to a single solution. This is because the master NUM problem, shown in Equation (2.2), is composed of multiple utility functions for text, audio, and video that can each decompose the NUM problem into a subproblem. Additionally, the master NUM problem identifies a price for resources. In this research, the price is the link capacity (*i.e.*; maximum data rate or throughput) and the resource is the data rate. In the master NUM problem, the constraint defines that the data rate for the application must be less than the link capacity. This link capacity varies for each subproblem depending on which media option is being used at the link for data transmission. Therefore, the NUM problem therefore has a coupling constraint, which is that the data rate must be less than the link capacity. The constraint can be modified or relaxed to decouple the NUM problem into several subproblems [25]. The dual decomposition method is also applicable to the NUM problem due to the zero duality gap [26]. A zero duality gap is also known as strong duality. Strong duality is an optimization concept according to which the primal optimal solution is equal to the dual optimal solution. This is normally the case for solutions to a convex NUM problem when they are converged to a single solution through dual decomposition.

The dual decomposition method presented in Equation (2.10) produces a distributed solution. λ is the Langrange multiplier that presents a link cost which is a tradeoff defined for the network:

$$\begin{aligned} \text{maximize} \quad & \sum_i f_i(x_i) \\ \text{subject to} \quad & \mathbf{x_i} \in \mathbf{X_i}, \\ & \sum_i h_i(x_i) \leq \mathbf{c} \end{aligned} \tag{2.9}$$

$$\begin{aligned} \text{maximize} \quad & f_i(x_i) - \lambda \times h_i(x_i) \\ \text{subject to} \quad & \mathbf{x_i} \in \mathbf{X_i} \end{aligned} \tag{2.10}$$

For the purposes of this thesis, the cross-layer network may use one of the three media options. Depending on which of the media methods are used, the problem can be customized to accommodate the method or methods. For instance, if audio is being used for data transmission at a link, then the universal utility function will be based on only the audio utility function defined in Equation (2.5), which will then be maximized in the NUM problem. Thus, the constraint that was changed or defined were the user and system parameters for the media quality, which decoupled the problem into a subproblem. In addition to the media, the utility function also varies according to the link capacity for the problem.

Through dual decomposition, the optimization problem would then become a simple optimization problem shown in Equation (2.11), where λ is the Langrange multiplier (greater than 0), which is the media distortion for the network media. Additionally, the link capacity would become a penalty, instead of a constraint, that is subtracted from the data rate when there is any positive Langrange multiplier. In this case, the Langrange multiplier would be the distortion of the data being transmitted:

$$\begin{aligned} \text{maximize} \quad & \sum_s U_s(x_s) - \lambda \times (x_s - c) \\ \text{subject to} \quad & \mathbf{x_s} \in \mathcal{X}, \\ & \lambda > \mathbf{0} \end{aligned} \tag{2.11}$$

As mentioned previously, the variable λ is the distortion of the data being transmitted. Since this thesis only considers the quality of one media option at every link, the distortion will be for either text, audio, or video data. Furthermore, text is either sent or not sent through text compression and decompression. Therefore, it cannot be distorted and λ would approach unity if text is transmitted. However, for audio and video data, there will be some distortion that needs to be accounted for in the optimization of the cross-layer NUM problem. The distortion of audio and video can be determined from a rate-distortion curve of audio and video through best fitting of the actual quality data [48]. The distortion can be determined for audio and video data based on the difference between the actual quality and best-fitting quality at any given data rate or bit rate.

2.5 Relevant Research in Cross-Layer Network Optimization

This thesis focuses on optimizing applications that require the use of text, audio, and video codecs for transferring data at a link (between nodes). The user and system (network) then defines the quality required for each media which is included in the cross-layer network optimization. Previous research has designed a cross-layer network system that controls the video encoding rate, the transmission rate, and the channel coding rate to maximize video quality [49]. The research of compressed-sensing-enabled video streaming for wireless multimedia sensor networks developed a video encoder based on compressed sending. It also introduced the Compressive Distortion Minimizing Rate Control (C-DMRC), that is a new distributed cross-layer control algorithm that maximizes the received video quality over a multi-hop wireless network by jointly regulating the compressed sensing (CS) sampling rate, the data rate input to the network, and the rate of a simple parity-based channel encoder. Additionally, C-DMRC uses the estimated received video quality as the bases of the rate-control decision where the transmitting node controls quality of the transmitted video. This is because the data rate of the video is linearly dependent on the video quality. The nodes also adapt to the rate of change of the transmitted video quality based on an estimate of the impact that a change in the transmission rate will have on the received video quality. Finally, the C-DMRC algorithm is an iterative solution to the optimal rate allocation problem which finds the data rates that maximize the sum of video qualities. The research uses the structural similarity (SSIM) index to evaluate video quality.

This thesis is similar to the compressed-sensing-enabled video streaming for wireless multimedia sensor networks research because it extends the maximization of video quality to text, audio, and video. It creates an algorithm to determine the optimal data rate for multimedia transmission at a link between two nodes of a cross-layer network based on user and system requirements for media quality. The media quality is determined through quality models, including SSIM for video. Each media option is transmitted through codecs that compress text, audio, and video data, similar to CS Video Encoder in [49].

2.6 Cross-Layer Network Optimization Objectives

The objective of the cross-layer network optimization based on application requirements for multimedia quality is to:

1. Model video, audio and text source quality for the user and the system. The Perceptual Evaluation of Audio Quality (PEAQ) algorithm [38] is used to determine audio quality and the Structural Similarity (SSIM) index [39] applied to determine video quality.
2. Derive a utility function and network utility maximization (NUM) [25] problem for link capacity and quality of video, audio, and text based on the data rate or bit rate.
3. Converge the cross-layer optimization problem into a distributed solution for the optimal data rate based on the link capacity and data transmission quality using the dual decomposition method.

The mathematical approach for cross-layer network optimization can ultimately be used as an algorithm for optimization of all of links in a multi-node network.

2.7 Chapter Summary

This chapter gave an overview of cross-layer network optimization. It introduced the network utility maximization (NUM) problem that is used in this thesis to maximize media quality over a network link using user and system media quality requirements as inputs. These media quality requirement inputs are derived from text, audio, and video quality models. The quality models are based on text, audio, and video codecs used for data transmission over a network link. The output to the NUM problem is the optimal data rate for transmission over a network link. This chapter also describes how to solve the NUM problem. An alternative way to solve for the optimization problem is using the dual decomposition method which converges it to a distributed solution for optimal data rate across the cross-layer network. Finally, this chapter introduces relevant research for cross-layer network optimization that were used to conduct the research for this thesis and the objectives of the thesis.

Chapter 3

Cross-Layer Network Optimization for Text, Audio, and Video

In this chapter, the cross-layer network optimization based on application requirements will be through specific constraints and results for each of the media options: text, audio, and video. It will give an in-depth view into how the user and system defined parameters for quality are taken into account for the cross-layer network utility maximization problem and its solution. The thesis focuses on optimizing a network for one media method based on the defined quality and for link capacity. Figure 3.1 gives a diagram of the components or steps to performing the cross-layer network optimization in this study. The user and system defined quality requirements for the application are inputs to formulate the convex utility function. If a distributed solution is not needed, the NUM problem is formulated and solved to find the optimal data rate. If a distributed solution is needed, the dual decomposition problem is formulated and converged to solve for the optimal data rate.

Section 3.1 will focus on optimization of the dynamic network based on application requirements for text. Section 3.2 will focus on optimization of the dynamic network based on application requirements for audio. Section 3.3 will focus on optimization of the dynamic network based on application requirements for video. Finally, Section 3.4 will provide a summary to conclude this chapter.

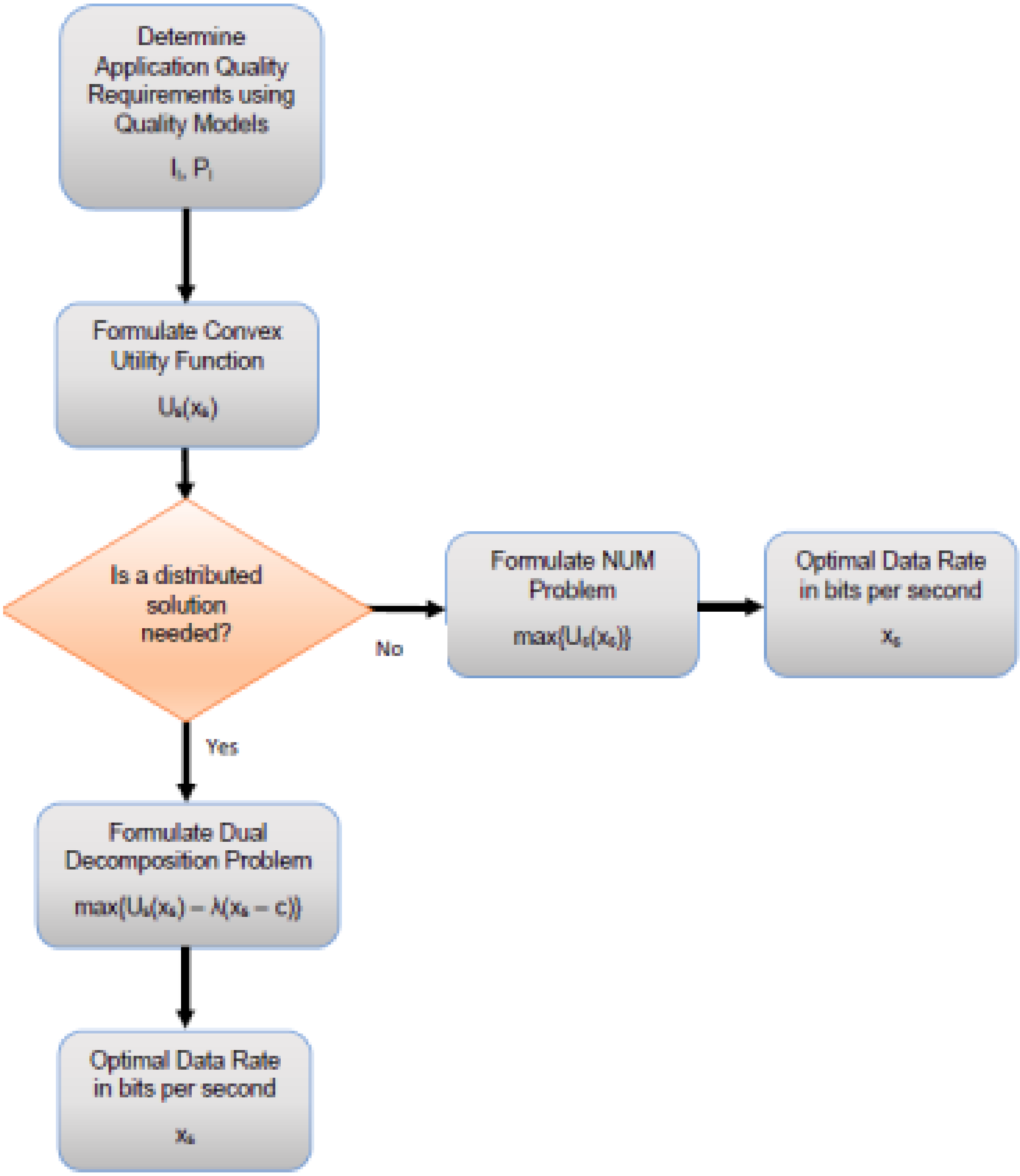
Determine Application Quality Requirements using Quality Models
I_i, P_i
Formulate Convex Utility Function
$U_s(x_s)$
Is a distributed solution needed?
No
Formulate NUM Problem
$\max\{U_s(x_s)\}$
Optimal Data Rate in bits per second
x_s
Yes
Formulate Dual Decomposition Problem
$\max\{U_s(x_s) - \lambda(x_s - c)\}$
Optimal Data Rate in bits per second
x_s

Figure 3.1: Cross-Layer Network Optimization Based on Application Requirements Thesis Component Diagram

3.1 Cross-Layer Network Optimization for Text

The cross-layer network optimization for text is relatively simple compared to audio and video. This is due to the fact that text is either transmitted or not. A text message can be compressed or encoded to send and then decompressed or decoded once received using Huffman Coding. Referring back to the universal utility function in Equation (2.7), the user and system defined parameters for text are I_T and P_T. For the text transmission to be optimized for the network, both of the parameters would have to equal to one (See Section 2.3.1) and the bit rate would have to be below 32 kilobits per second. With these constraints, the universal utility function shown in Equation (2.7) would then be simplified into Equation (2.4). Note that c, which is the link capacity, would be the maximum data rate or bit rate allowed for text. In this case, that is 32 kilobits per second. However, the link capacity is subject to change according to the maximum throughput allowed at the each network link and therefore will be left as c. The NUM problem would then become the following:

$$\begin{aligned} &\text{maximize} && \sum_s U_s(x_s) \text{ where } U_s(x_s) = 1/(1+e^{-(2.3748\times 10^{-4})\times x_s}) \\ &\text{subject to} && \mathbf{x_s} \leq \mathbf{c}, \\ & && \mathbf{x_s} \in [\mathbf{0}, \mathbf{32}] \text{ kilobits per second} \end{aligned} \tag{3.1}$$

To solve this NUM problem through the dual decomposition method described in Section 2.4, note that λ, which is the Langrangian multiplier that defines the distortion of the data being transmitted, would just be 0 since text is an inelastic transmission method that can either be sent or not without any distortion. Thus, the dual decomposition of the NUM problem would then become the following:

$$\begin{aligned} &\text{maximize} && \sum_s U_s(x_s) \text{ where } U_s(x_s) = 1/(1+e^{-(2.3748\times 10^{-4})\times x_s}) \\ &\text{subject to} && \mathbf{x_s} \leq \mathbf{c}, \\ & && \mathbf{x_s} \in [\mathbf{0}, \mathbf{32}] \text{ kilobits per second} \end{aligned} \tag{3.2}$$

Notice that the dual decomposition of the original NUM problem for the text transmission method are the same as there is no distortion to be accounted for in the solution.

3.2 Cross-Layer Network Optimization for Audio

The cross-layer optimization for audio has to specify the exact user and system defined parameters or requirements for audio quality. The user and system defined parameters for audio quality are I_A and P_A in Equation (2.7), which is the universal utility function, respectively. These parameters are determined by the Normalized Audio Scaleand will be a value greater than 0 and less than or equal to 1. Additionally, the data rate or bit rate would have to be greater than 32 kilobits per second and less than or equal to 320 kilobits per second. Finally, the maximum data rate or link capacity, c, can be defined by the maximum throughput allowed for each link. This would modify the universal utility function from Equation (2.7) to the following:

$$\begin{aligned} &\text{maximize} \\ &\qquad U_s(x_s) \text{ where } U_s(x_s) = I_A \times P_A \times 1/(1 + (-(2.6387 * 10^{-5}) \times (e^{(x_s - 288000)}))) \\ &\text{subject to} \qquad \mathbf{x_s} \leq \mathbf{c}, \\ &\qquad\qquad\qquad \mathbf{x_s} \in [\mathbf{32}, \mathbf{320}] \text{ kilobits per second} \end{aligned} \tag{3.3}$$

Notice the NUM problem with the universal utility function simplifies to only consider the utility function for audio from Equation (2.5) since only the audio transmission method is being utilized for the cross-layer network. The audio utility function is then multiplied by the user and system defined parameters for audio quality to form the overall NUM problem.

The user and system defined audio quality parameters are determined by the Normalized Audio Scale derived from the PEAQ Algorithm.. This thesis uses a MATLAB program to determine the audio quality. The reference audio sample possessing the ideal audio quality has a bit rate of 256 kilobits per second and a sampling rate of 48000 Hertz, which is a constant sampling rate used for all of the audio samples. Other audio samples will be tested against this reference audio sample in order to determine their quality on the Normalized Audio Scale. Figure 3.2 displays the behavior of the Normal Audio Scale for MP3 based on the bit rate. The audio quality was measured for bit rate values from 0 kilobits per second to 320 kilobits per second. Table 3.1 shows the bit rates in kilobits per second versus the normalized audio quality values. The audio quality value increases, which means audio quality increases, as bit rate increases. The audio quality value decreases slightly for 320 kilobits per second is because the reference audio sample has a bit rate of 256 kilobits per second, which is still ideal.

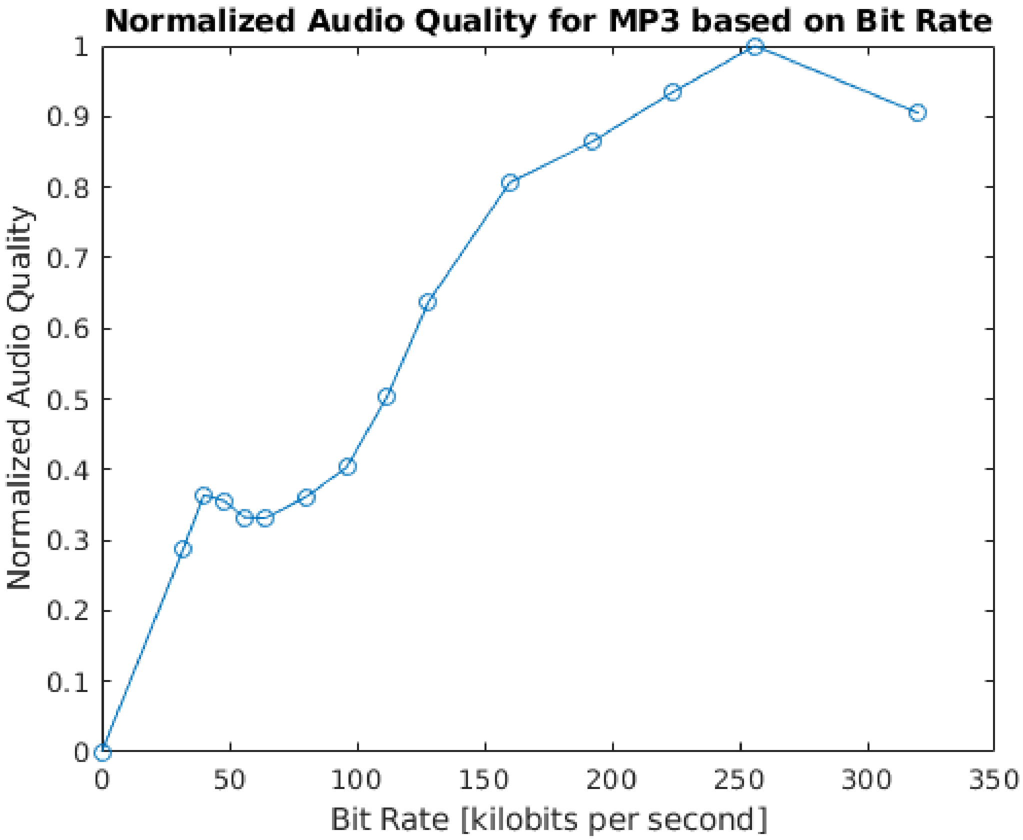

Figure 3.2: Normalized Audio Quality Scale for MP3 based on Bit Rate. This curve is an analysis of the MP3 Audio quality. It is generally increasing and concave.

Once the user and system defined parameters are determined for audio quality, the NUM problem can then go through dual decomposition to find a solution and optimize the cross-layer network. Equation (3.4) shows the dual decomposition of the NUM for the audio transmission method in order to achieve the cross-layer network optimization. The variable λ, which is the Langrange multiplier and distortion of audio, specifically for

Table 3.1: Bit Rate vs. Normalized Audio Quality Value. This is an analysis of the audio quality for the MP3 files created at the bit rate, or data rates, shown in the table.

Bit Rate vs. Normalized Audio Quality Value	
Bit Rate [kilobits per second]	Normalized Audio Quality Value
32	0.2874
40	0.3637
48	0.3554
56	0.3317
64	0.3308
80	0.3609
96	0.4029
112	0.5042
128	0.6370
160	0.8067
192	0.8638
224	0.9348
256	1.0000
320	0.9046

MP3 audio in this thesis. The Langrange multiplier can be determined from best fitting of the Normalized Audio Quality Scale for MP3 based on bit rate shown in the graph in Figure 3.2. The absolute value of the difference between the cubic-fitting or the rate-distortion curve and the actual plot gives the distortion in terms of the Normalized Audio Quality Scale, which is what should be used for λ when optimizing the cross-layer network for audio transmission across the network links. The cubic-fitting of the actual data was found in MATLAB. Figure 3.3 shows the cubic-fitting of the Normalized Audio Quality data. Additionally, Table 3.2 gives the distortion for λ at each bit rate that the Normalized Audio Quality value was measured for the audio sample in this thesis. Note for the best-fitting, the Normalized Audio Scale values are only shown up to 256 kilobits per second since that is the ideal audio sample for this thesis and it gives the best curve-fitting.

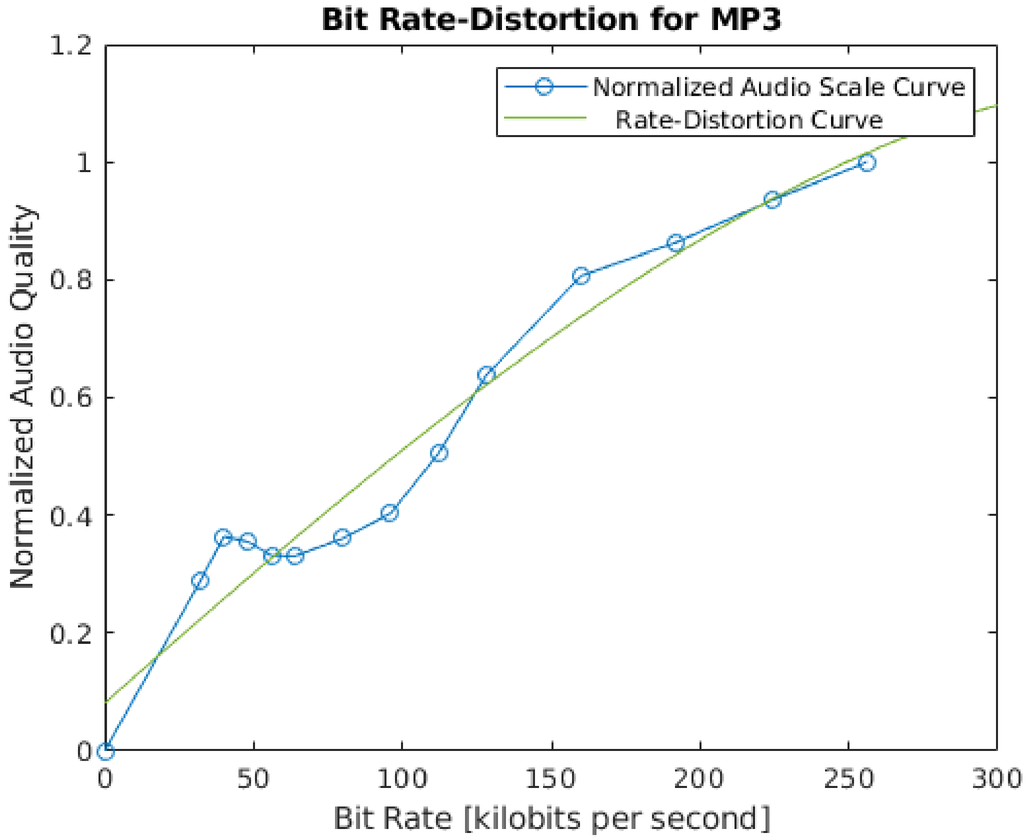

Figure 3.3: Cubic-Fitting of Normalized Audio Quality Values for MP3. This fitting is the rate-distortion curve for MP3 Audio. It can be used to determine the Langrange multiplier for the dual decomposition.

After determining λ by determining the distortion value that best matches the user's chosen quality value for I_A, the dual decomposition of the NUM problem for audio transmission is derived from Equation (2.11) to the following:

$$\begin{aligned} &\text{maximize} && \sum_s U_s(x_s) - \lambda \times (x_s - c) \\ &\text{where} && U_s(x_s) = I_A \times P_A \times 1/(1 + (-(2.6387 \times 10^{-5}) \times (e^{(x_s - 288000)}))) \\ &\text{subject to} && \mathbf{x_s} \in [\mathbf{32}, \mathbf{320}] \text{ kilobits per second} \end{aligned} \tag{3.4}$$

Table 3.2: Distortion for Langrange Multiplier of MP3 Audio. This is an analysis of the audio distortion, or noise, for the MP3 files created at the bit rate, or data rate, shown in the table.

Distortion for Langrange Multiplier of MP3 Audio	
Bit Rate [kilobits per second]	Distortion in terms of Normalized Audio Quality Value
32	0.0640
40	0.1052
48	0.0620
56	0.0037
64	0.0315
80	0.0691
96	0.0830
112	0.0098
128	0.0249
160	0.0687
192	0.0208
224	0.0017
256	0.0000

3.3 Cross-Layer Network Optimization for Video

For the cross-layer network optimization of video, the user and system defined parameters for video quality are I_V and P_V respectively. The universal utility function from Equation (2.7) would then be modified to only accomodate the utility function for video. Additionally, the maximum data rate or link capacity, c, can be defined by the maximum throughput allowed for each link. The NUM problem would then become the following:

$$
\begin{aligned}
&\text{maximize} \\
&U_s(x_s) \text{ where } U_s(x_s) = && I_V \times P_V \times 1/(1 + (-(1.9998 \times 10^{-7}) \times (e^{(x_s - 38000000)}))) \\
&\text{subject to} && \mathbf{x_s} \leq \mathbf{c}, \\
& && \mathbf{x_s} \in [\mathbf{12000}, \mathbf{50000}] \text{ kilobits per second}
\end{aligned}
\tag{3.5}
$$

The video codec used was H.265 or HEVC for this thesis. HEVC is known for its high quality compression performance at lower data rates compared to existing video codec standards [46]. Therefore, the user and system defined parameters will hold SSIM values that are high quality. Additionally, the constraints for the HEVC video samples in this thesis are very specific due to the video performance accepted by the Linux platform used, which means only a few variable data rates or bit rates could be tested for. The resolution for all of the video samples is 2048x1080, the sampling rate is between 66,846,720 samples per second and 133,693,440 samples per second and the frame rate is 30 frames per second. For these constraints, the data rates or bit rates HEVC can perform on are Level 4.0 and 4.1 bit rates. These rates are 12,000 kilobits per second, 30,000 kilobits per second, 20,000 kilobits per second and 50,000 kilobits per second. The SSIM values, which are used for the user and system defined video quality parameters, and the PSNR values were determined for these bit rates. All of the tested video samples were tested against the reference video sample or ideal video sample which is the one with the highest bit rate. In this case, that is 50,000 kilobits per second. Figures 3.4, 3.5, 3.6 and 3.7 show the frames taken from the video samples tested for this thesis:

The graph in Figure 3.8 shows the curve for the SSIM quality values based on the bit rates and Table 3.3 gives the data points. The graph in Figure 3.9 shows the curve for the PSNR values based on the bit rates and Table 3.4 gives the data points. Note for PSNR values, when comparing two of the same videos for quality, the value comes out to 0 so the video sample with 50,000 kilobits per second was omitted as a test point.

After, the video quality parameters are determined by the user and the system, the NUM problem can be dual decomposed to find a solution to optimize the cross-layer network for video transmission across the network links. In order to find a solution, the Langrangian

HEVC / h265
www.h265files.com
Location: Dachstein / Austria
Footage: Frederick Tschernutter, Helmut Strasser
Music: LoveShadow - You Made It Up (cdk Mix)

Figure 3.4: Sample Video Frame with 12000 kilobits per second Bit Rate. This is from a video sample that is being tested in the PEAQ algorithm.

HEVC / h265
www.h265files.com
Location: Dachstein / Austria
Footage: Frederick Tschernutter, Helmut Strasser
Music: LoveShadow - You Made It Up (cdk Mix)

Figure 3.5: Sample Video Frame with 20000 kilobits per second Bit Rate. This is from a video sample that is being tested in the PEAQ algorithm.

Multiplier, λ, or distortion for dual decomposition must be determined. The distortion will be determined by creating a best-fitting linear regression or rate-distortion line on the graph

HEVC / h265
www.h265files.com
Location: Dachstein / Austria
Footage: Frederick Tschernutter, Helmut Strasser
Music: LoveShadow - You Made It Up (cdk Mix)

Figure 3.6: Sample Video Frame with 30000 kilobits per second Bit Rate. This is from a video sample that is being tested in the PEAQ algorithm.

HEVC / h265
www.h265files.com
Location: Dachstein / Austria
Footage: Frederick Tschernutter, Helmut Strasser
Music: LoveShadow - You Made It Up (cdk Mix)

Figure 3.7: Sample Video Frame with 50000 kilobits per second Bit Rate. This is from the video sample that is being references in the PEAQ algorithm.

for SSIM values versus bit rate found in Figure 3.8. The absolute value of the difference between the linear regression and the actual SSIM curve at any given data rate will give

Figure 3.8: SSIM Values based on Bit Rate for HEVC. The SSIM quality increases as bit rate, or data rate, increases.

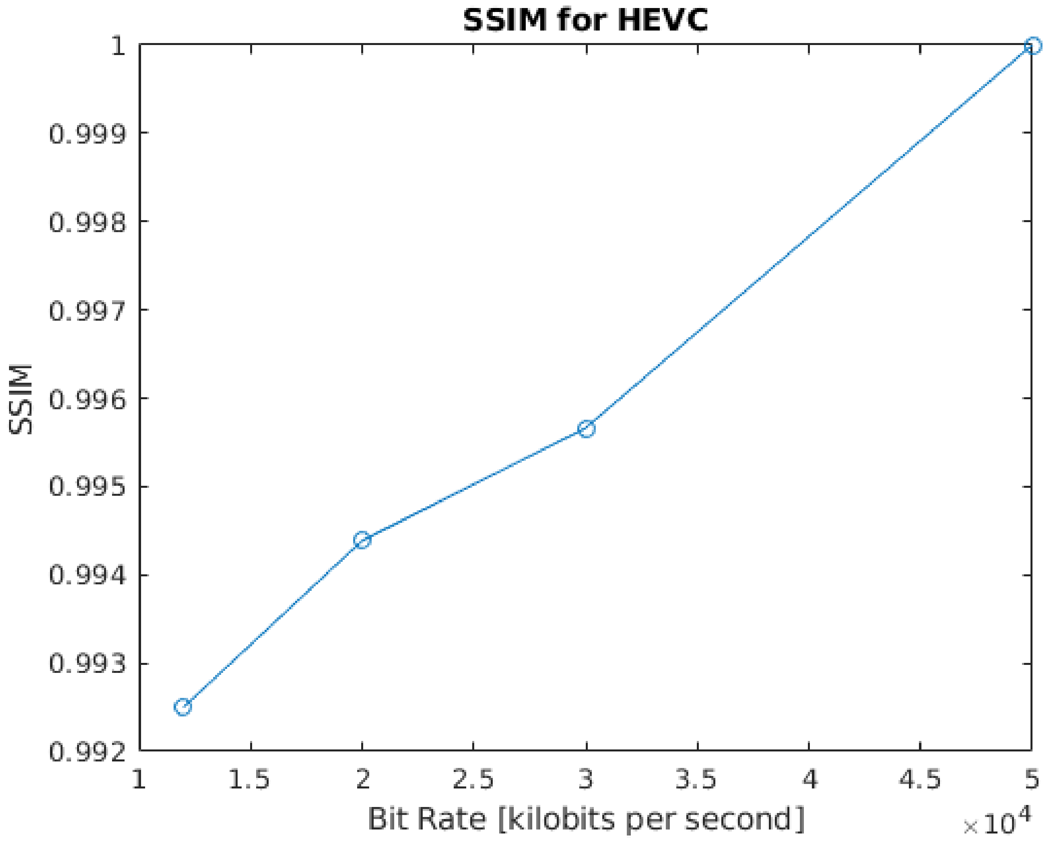

Table 3.3: Bit Rate vs. SSIM Values

Bit Rate vs. SSIM Values	
Bit Rate [kilobits per second]	SSIM
12000	0.992498
20000	0.994385
30000	0.995661
50000	1.000000

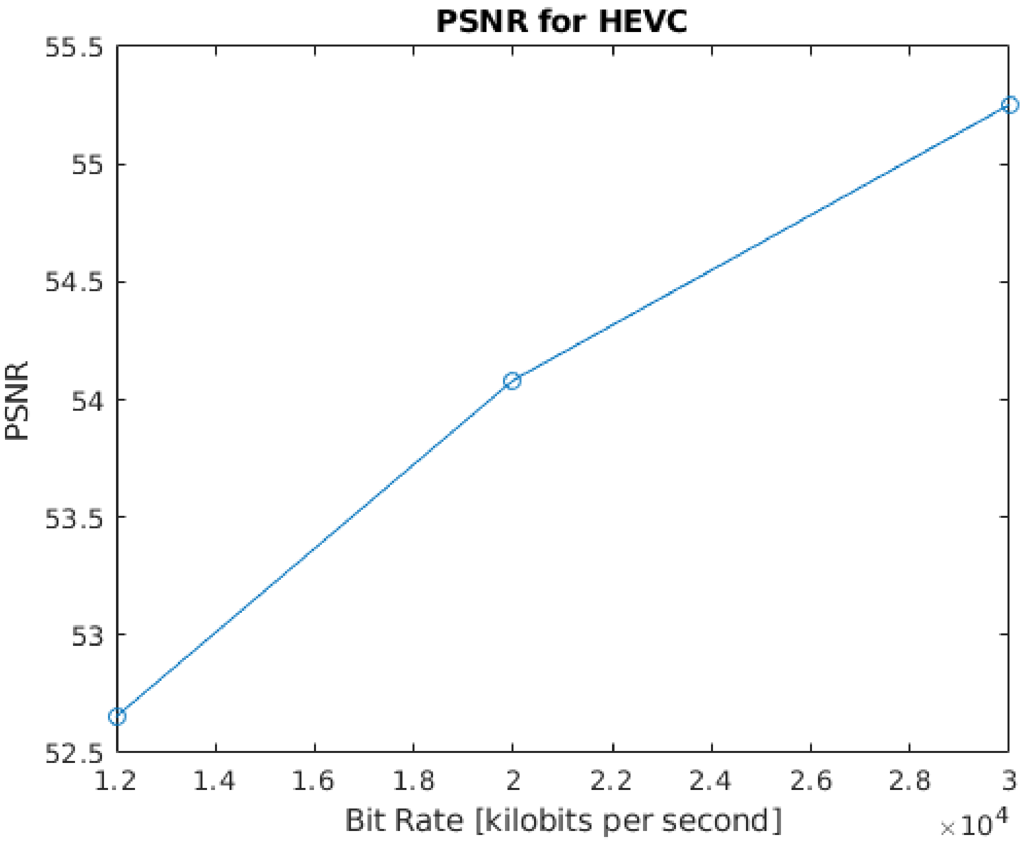

Figure 3.9: PSNR Values based on Bit Rate for HEVC. The PSNR quality increases as bit rate, or data rate, increases.

Table 3.4: Bit Rate vs. PSNR Values

Bit Rate vs. PSNR Values	
Bit Rate [kilobits per second]	PSNR
12000	52.653395
20000	54.082003
30000	55.241703
50000	0.0000000

the distortion for λ. Figure 3.10 gives the linear regression of the actual SSIM curve.

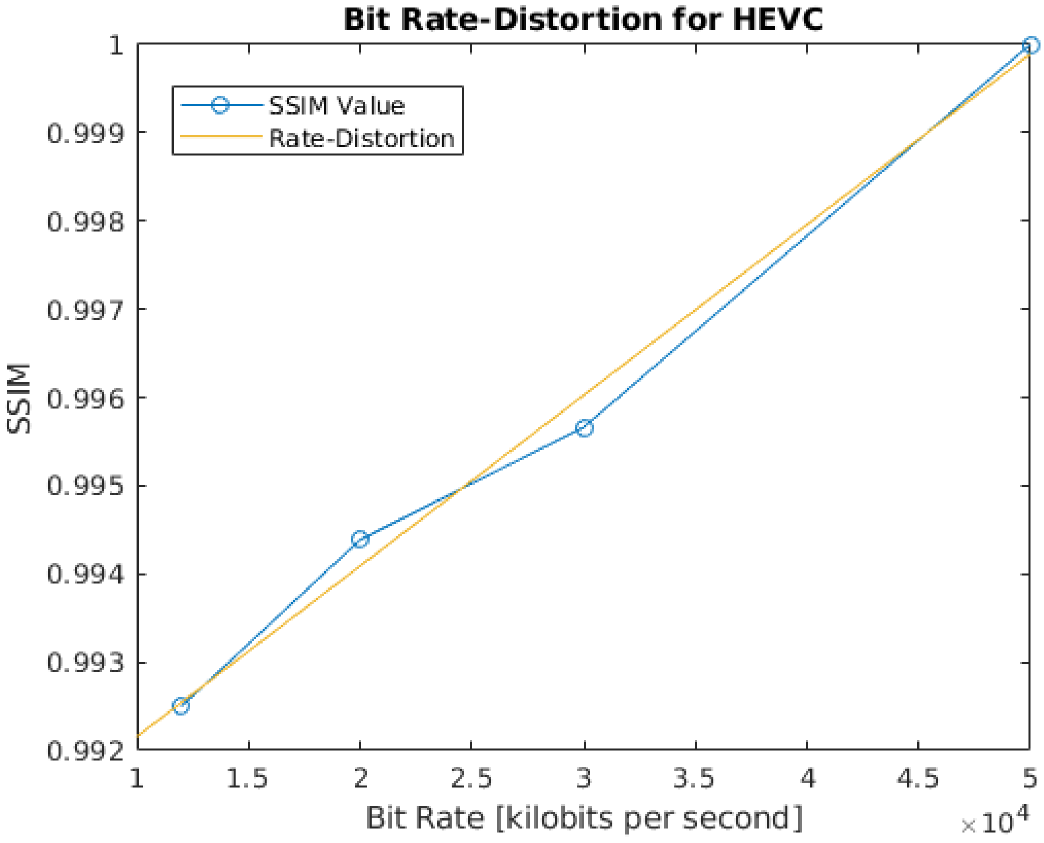

Figure 3.10: Linear Regression of SSIM Values for HEVC. This linear regression is the rate-distortion curve for HEVC and can be used to determine the Langrange multiplier for dual decomposition.

Based on this plot, the distortion values that can be used for the Langrangian Multiplier at each bit rate that the SSIM was measured for is found in Table 3.5.

After determining λ by determining the distortion value that best matches the user's chosen quality value for I_V, the dual decomposition of the NUM problem for video transmission can be derived from Equation (2.11) in order to produce:

Table 3.5: Distortion for Langrangian Multiplier for HEVC

Bit Rate vs. Distortion	
Bit Rate [kilobits per second]	Distortion in terms of SSIM Index
12000	0.0000000
20000	0.0002850
30000	0.0003390
50000	0.0001000

$$\begin{aligned} &\text{maximize} \quad \sum_s U_s(x_s) - \lambda \times (x_s - c) \\ &\text{where} \\ &U_s(x_s) = \quad I_V \times P_V \times 1/(1 + (-(1.9998 \times 10^{-7}) \times (e^{(x_s - 38000000)}))) \\ &\text{subject to} \quad \mathbf{x_s} \in [\mathbf{12000}, \mathbf{50000}] \text{ kilobits per second} \end{aligned} \tag{3.6}$$

3.4 Chapter Summary

The network for the multimedia application in this research only focuses on one media option for data transmission. This means that the system may be able to accommodate more than one media method, for which the quality is input as a system requirement, but the user defines either text, audio, or video quality as a requirement for transmission across the network links. This chapter reviewed the cross-layer network optimization for each of the media options, which are text, audio, and video, and the related NUM problem and dual decomposition problem that are solved to determine the optimal data rate.

Chapter 4

Implementation of Cross-Layer Network Optimization

This chapter introduces an Air Force network scenario that is used to implement the cross-layer network optimization methodology from this thesis. Section 4.1 introduces the scenario and describes the nodes and links for the network being optimized. The optimization will determine the best media option and optimal data rate to send an incident report between two nodes. Section 4.2 details the first steps to the network optimization, which are to determine the system and user defined requirements for media quality, the media option being optimized for, and the network utility function. Section 4.3 describes the network optimization by solving the NUM problem for the scenario's network using the MATLAB Software for Disciplined Convex Programming (CVX). Finally 4.4 concludes the chapter with a chapter summary.

4.1 Cross-Layer Network

The Air Force provides a multi-node, swarm network scenario which serves as a common environment that can be used to implement the cross-layer network optimization from this thesis. The environment consists of a network with ten aerial nodes that monitor the Rome, New York, USA area around Griffiss Air Force Base through a multimedia application. In the scenario, node n10 experiences an incident of interest which is detected by nodes n7, n8, n9, and n11. However, node n7 must report the incident to node n6 over a link. Node n6 must then transmit this data to node n3 which will then report it to the gateway node wlan1 at the Air Force Base. Figure 4.1 displays the scenario with the nodes and Figure 4.2 displays the nodes with links that connect them to transmit data.

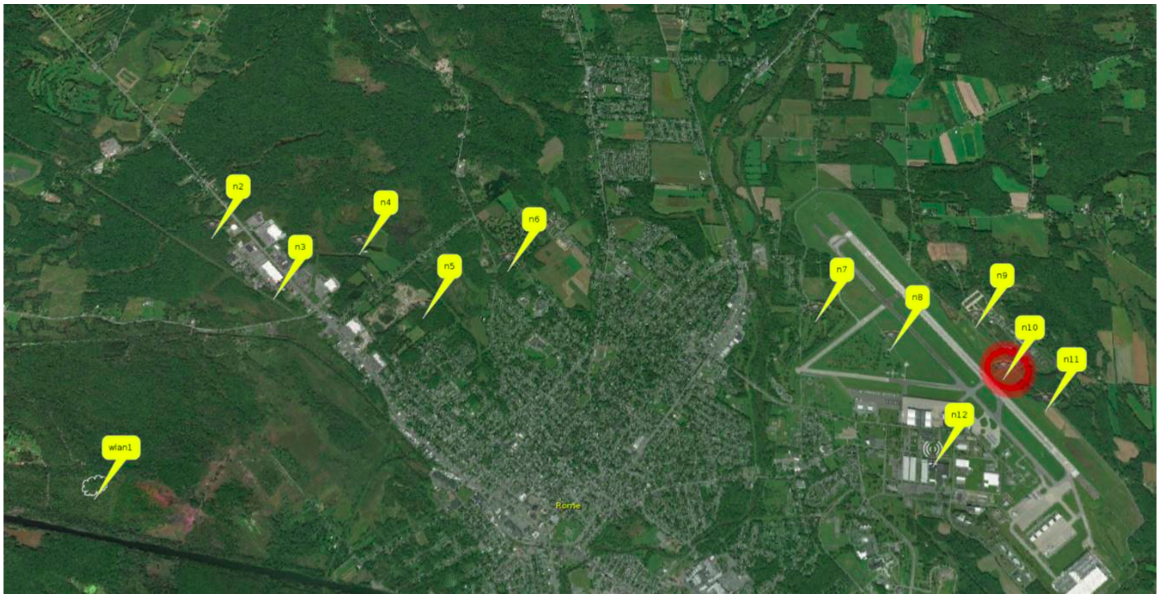

Figure 4.1: Air Force Network Scenario with Nodes. It is located in Rome, NY near Griffiss Air Force Base.

The multimedia application that monitors the network nodes must implement the cross-layer network optimization method from this thesis. The nodes that are detecting the incident of interest at node n10 must determine the optimal media option that can be used to transmit the report from node n7 to node n6 over a link, which is indicated as a red link in Figure 4.2. These are the nodes that the cross-layer network optimization will be performed on.

4.2 Network Utility based on Media Quality Requirements

To determine the optimal media option, which can be text, audio, or video, the system's or network's capability must be considered for nodes n7 and n6. These nodes can

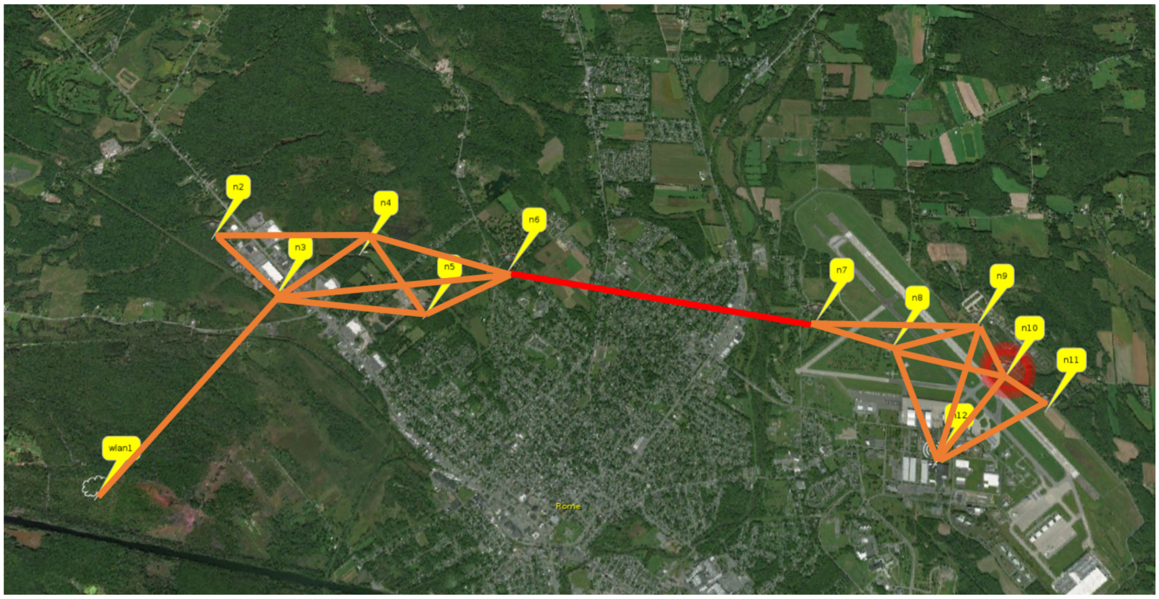

Figure 4.2: Air Force Network Scenario with Nodes and Links. The red link indicated the link that the incident report data is being transmitted over and optimized for between nodes n7 and n6.

accomodate a HEVC video message with an SSIM index of 0.991, 10 MP3 audio messages with a Normalized Audio Scale value of 0.9, or 100 Huffman Compression text message files. This means that the system parameter for video quality P_V is equal to 0.991, the system parameter for audio quality P_A is equal to 0.90, the system parameter for text quality P_T is equal to 1. These parameters are constants in the universal utility function from Equation (2.7), which then becomes the following:

$$\begin{aligned} &U_s(x_s) = (I_T \times 1 \times U_T) + (I_A \times 0.90 \times U_A) + (I_V \times 0.991 \times U_V) \\ &\text{ where } I_i \in [0,1], \\ &\sum I_i \leq 1. \end{aligned} \tag{4.1}$$

Next, the user's decision for the application's media option must be determined for nodes n7 and n6. In this scenario and thesis, only one media option can be used. The user wants the incident report to be transmitted at the red link as a audio messages with a normalized audio scale quality of 0.80. Therefore, the user parameters for video quality I_V is equal to

0, the user parameter for audio quality I_A is equal to 0.80, and the user parameter for text quality I_T is equal to 0. With the known user and system quality parameters, the universal utility function then becomes:

$$U_s(x_s) = (U_A \times 0.90 \times 0.80) = (U_A \times 0.72) \tag{4.2}$$

where U_A is derived from Equation (2.5). Thus, the utility function then becomes the following:

$$U_s(x_s) = (0.72/(1 + (e^{-(2.6387 \times 10^{-5}) \times (x_s - 288000)}))) \tag{4.3}$$

4.3 Optimizing the Cross-Layer Network

The derived network utility function from the scenario can be inputed into the NUM problem from Equation (4.4). The NUM problem for this implementation is then derived as:

$$\begin{aligned} \text{maximize} \quad & \sum_s U_s(x_s) \text{ where } U_s(x_s) = (0.72/(1 + (e^{-(2.6387 \times 10^{-5})(x_s - 288000)}))) \\ \text{subject to} \quad & \mathbf{x_s} \leq \mathbf{320} \text{ kilobits per second }, \\ & \mathbf{x_s} \in [\mathbf{32}, \mathbf{320}] \text{ kilobits per second} \end{aligned} \tag{4.4}$$

where the data rate is less than 32 kilobits per second which is the link capacity and the highest data rate threshold for the MP3 media. Additionally, the data rate is in between 32 and 320 kilobits per second, which are the lower and upper bound thresholds for the data rate of the MP3 media. Using the MATLAB Software for Disciplined Convex Programming (CVX) [50], the optimal solutions for the NUM problem in Equation (4.4) can be derived. CVX turns MATLAB into an optimization modeling language that constructs and solves convex problems. Fortunately, there was only one solution to the NUM problem for the application in the scenario where the optimal data rate for the cross-layer network is 50,355.90 bits per second. This determines that the optimal network transmission method for the incident report at node n7 and n6, shown in Figure 4.2, is up to 20 audio messages transferred at an optimal data rate of 50,355.90 bits per second. This optimal solution meets the system and user defined requirements for the cross-layer network of the application in this Air Force scenario.

Another way to solve for the optimal data rate is by converging the problem into a distributed solution using the dual decomposition method defined in Equation (3.4). The dual decomposition method would be modified for this application that uses audio for transmission to:

$$\begin{aligned} &\text{maximize} && \sum_s U_s(x_s) - \lambda \times (x_s - 320000) \\ &\text{where} && U_s(x_s) = 0.72/(1 + (-(2.6387 \times 10^{-5}) \times (e^{(x_s - 288000)}))) \\ &\text{subject to} && \mathbf{x_s} \in [\mathbf{32}, \mathbf{320}] \text{ kilobits per second} \end{aligned} \tag{4.5}$$

where λ is the Langrange Multiplier, which is the distortion for audio of the NUM problem.

4.4 Chapter Summary

This chapter implemented the cross-layer network optimization for a multimedia application on an Air Force environment that consists of a scenario for a multi-node, swarm network. It detailed the derivation of the NUM problem using the media quality requirements for the user and the system. It is assumed that the user and system have already used the quality models to determine the input requirements. The chapter then solved the NUM problem using the CVX Software to determine the optimal data rate for MP3 audio message transmission at two nodes to send an incident report.

Chapter 5

Conclusion

In this thesis, a mathematical approach was successfully developed and implemented on a scenario for cross-layer network optimization based on user and system (network) application requirements for data transmission quality and capacity. This mathematical approach can be extended to the entire application network. The following sections will describe the accomplishments of this thesis and potential future work.

5.1 Research Outcomes

The cross-layer network optimization was completed by converging a network utility maximization (NUM) problem into a solution through dual decomposition. The user and system (network) defined requirements for quality of data transmission methods or communication methods, which include text, audio and video, were determined using different algorithms through MATLAB software that produced normalized scales. The quality value based on the normalized scale was included in the utility function(s). Additionally, the distortion of the communication method was determined when performing the dual decomposition for the NUM problem using MATLAB software. These distortion values were found from the normalized scale curves for text, audio and video based on best-fitting curve methods. Finally, the link capacity is the maximum data rate (bit rate) or maximum allowed throughput at each link of the network being optimized. The network in this thesis only handles one communication method for optimization. In conclusion, this thesis derived a mathematical approach to optimizing a cross-layer network for data transmission quality and capacity for the text, audio and video communication methods. The optimal solution is the data rate and media option that is the best to use for transferring data withing a dynamic network for an application. The derived network optimization was implemented on an Air Force environment with a multi-node, swarm network for which a multimedia application must transmit an incident report. This is just an example of what this research can be utilized for.

5.2 Future Work

This thesis should be developed even further by updating this mathematical approach to optimize a cross-layer network based on user and system defined application requirements for quality and link capacity into an algorithm that can be implemented on a dynamic network. The algorithm can then be built in a software that can be implemented on multimedia applications for multi-node networks. The software should be developed to implement the algorithm iteratively across a cross-layer network in order to optimize all of the nodes in a multi-node application. This software would take the quality models for audio, video and text and NUM problem and converge it into a solution using dual decomposition. It would simply take the user and system application requirements for quality which can be either a specified data rate (bit rate), specified communication method, and/or specified quality index or value for a communication method. Based on the user and system defined requirements, the software can then optimize the nodes for quality and capacity. If this software is built and implemented on a real-time swarm application, many achievements would be made in order to efficiently meet mission-demands for quality and throughput in a swarm network. Finally, this research can be developed even further by including other factors that may need to be optimized based on user and system needs besides quality and capacity. Such network factors can include latency, traffic congestion, error rates, and jitter. All future work can be simulated or emulated through software for real-time testing before being implemented in an application. Example emulators that were created by the United States Navy Research Laboratory (NRL) is the Common Open Research Emulator (CORE) and Extendable Mobile Ad-Hoc Network Emulator (EMANE) that model a mobile network in real-time through links and physical layers of a network (i.e., cross-layer network).

Bibliography

[1] Z. Akhtar and T. H. Falk, “Audio-visual multimedia quality assessment: A comprehensive survey,” *IEEE Access*, vol. 5, pp. 21 090–21 117, Sept 2017.

[2] A. Stoica, “Swarm autonomy.” [Online]. Available: https://www-robotics.jpl.nasa.gov/tasks/showTask.cfm?FuseAction=ShowTask&TaskID=349&tdaID=700142

[3] W. Truszkowski, H. L. Hallock, C. Rouff, J. Karlin, J. Rash, M. Hinchey, and R. Sterritt, “Autonomic systems,” *NASA Monographs in Systems and Software Engineering Autonomous and Autonomic Systems: With Applications to NASA Intelligent Spacecraft Operations and Exploration Systems*, p. 173186, 2009.

[4] K. Osborn, “The u.s. air force of the future: Hypersonic weapons, swarm strikes and stealth fighters,” Nov 2018. [Online]. Available: https://nationalinterest.org/blog/buzz/us-air-force-future-hypersonic-weapons-swarm-strikes-and-stealth-fighters-37482

[5] E. Karami and S. Glisic, “Optimization of routing, network coding and scheduling in wireless multicast ad-hoc networks with topology compression,” in *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*. Tokyo, Japan: IEEE, Apr 2010, pp. 410–414.

[6] M. Fiedler, S. Chevul, O. Radtke, K. Tutschku, and A. Binzenhofer, “The network utility function: A practicable concept for assessing network impact on distributed services,” Jan 2005.

[7] W. Jiang, S. Huang, and J. Zhao, “Research on qos optimization for intra-flight ad hoc

network," in *2nd IEEE International Conference on Computer and Communications*, 2016.

[8] ——, "Research on qos optimization for intra-flight ad hoc networks," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC).* Chengdu, China: IEEE, May 2017.

[9] X. Zhang and J. Wang, "Joint heterogeneous statistical-qos/qoe provisionings for edge-computing based wifi offloading over 5g mobile wireless networks," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS).* Princeton, NJ, USA: IEEE, May 2018, pp. 1–6.

[10] H. Kim, H. Lee, and S. Lee, "A cross-layer optimization for energy-efficient mac protocol with delay and rate constraints," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Prague, Czech Republic: IEEE, Jul 2011, pp. 2336–2339.

[11] Z. Karakehayov, "Deployment-specific lifetime/real-time optimization for wireless ad hoc networks," in *2009 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications.* Rende, Italy: IEEE, Dec 2009, pp. 661–663.

[12] S. Ci and J. Sonnenberg, "A cognitive cross-layer architecture for next-generation tactical networks," in *MILCOM 2007 - IEEE Military Communications Conference.* Orlando, FL, USA: IEEE, Oct 2007.

[13] Y. Wang, W. Wang, W. Cui, K. G. Shin, and Z. Zhang, "Distributed packet forwarding and caching based on stochastic network utility maximization," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1264–1277, Jun 2018.

[14] I. A. Buturlin, Y. V. Gaidamaka, and A. K. Samuylov, "Utility function maximization problems for two cross-layer optimization algorithms in ofdm wireless networks," in *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems.* St. Petersburg, Russia: IEEE, Feb 2013, pp. 63–65.

[15] S. Liao, Z. Yang, W. Chen, and W. Liu, “Cross-layer optimization based on utility maximization with delay constraints in ad hoc wireless networks,” in *2009 ISECS International Colloquium on Computing, Communication, Control, and Management.* Sanya, China: IEEE, Sept 2009, pp. 107–110.

[16] X. Han, X. Zhang, and J. Lv, “Joint rate control and power control in wireless ad hoc networks with qos requirements,” in *2008 Fourth International Conference on Networked Computing and Advanced Information Management.* Gyeongju, South Korea: IEEE, Sept 2008, pp. 480–484.

[17] R. Poli, “An analysis of publications on particle swarm optimisation applications,” *Technical Report CSM-469*, May 2007.

[18] A. M. Wyglinski, M. Nekovee, and Y. T. Hou, *Cognitive Radio Communications and Networks.* Academic Press, 2010.

[19] F. Fu and M. Van Der Schaar, “A new theoretic framework for cross-layer optimization,” in *2008 15th IEEE International Conference on Image Processing.* San Diego, CA, USA: IEEE, Dec 2008, pp. 3080–3083.

[20] Microsoft, “Windows network architecture and the osi model,” Apr 2017.

[21] P. Simoneau, “The osi model: Understanding the seven layers of computer networks,” 2006.

[22] “Pearson it certification.” [Online]. Available: http://www.pearsonitcertification.com/articles/article.aspx?p=1804869

[23] M. Ashour, J. Wang, C. Lagao, N. Aybat, and H. Che, “Non-concave network utility maximization: A distributed optimization approach,” in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications.* Atlanta, GA, USA: IEEE, Oct 2017.

[24] G. Tychogiorgos, A. Gkelias, and K. K. Leaun, “Utility-proportional fairness in wireless networks,” in *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications.* Sydney, NSW, Australia: IEEE, Sep 2012.

[25] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, “Rate control for communication networks: Shadow prices, proportional fairness and stability,” *The Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, Mar 1998.

[26] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, “Layering as optimization decomposition:a mathematical theory of network architectures,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, Jan 2007.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[28] M. Fazel and M. Chiang, “Network utility maximization with nonconcave utilities using sum-of-squares method,” in *Proceedings of the 44th IEEE Conference on Decision and Control.* Seville, Spain, Spain: IEEE, Jan 2006.

[29] B. R. Chandavarkar and G. R. M. Reddy, “Analysis of sigmoidal utility function based handover in heterogeneous networks,” in *2015 International Conference on Microwave, Optical and Communication Engineering (ICMOCE).* Bhubaneswar, India: IEEE, Jun 2015.

[30] M. Udell and S. Boyd, “Maximizing a sum of sigmoids,” May 2015.

[31] H. A. Nguyen, T. V. Nguyen, and D. Choi, “How to maximize user satisfaction degree in multi-service ip networks,” in *2009 First Asian Conference on Intelligent Information and Database Systems.* Dong Hoi, Vietnam: IEEE, Apr 2009.

[32] F. L. Pryor, “The friedman-savage utility function in cross-cultural perspective,” *Journal of Political Economy*, vol. 84, no. 4, pp. 821–834, Aug 1976.

[33] M. Friedman and L. J. Savage, “The utility analysis of choices involcing risk,” *The Journal of Political Economy*, vol. 56, no. 4, pp. 279–304, Aug 1948.

[34] H.-A. Park, “An introduction to logistic regression: From basic concepts to interpretation with particular attention to nursing domain,” *J Korean Academy of Nursing*, vol. 43, no. 2, pp. 154–164, Apr 2013.

[35] M. M. P. E. Group), "Cd 11172-3 - coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s part 3 audio." ISO-IEC/JTCI, Nov 1991.

[36] I. T. Union, "Series h: Audiovisual and multimedia systems - infrastructure of audiovidual services - coding of moving video - high efficiency video coding." Telecommunication Standardization Sector of ITU, Feb 2018.

[37] D. Campeanu and A. Campeanu, "Peaq - an objective method to assess the perceptual quality of audio compressed files," 2014.

[38] M. Salovarda, I. Boljovac, and H. Domitrovic, "Estimating perceptual audio system quality using peaq algorithm," in *2005 18th International Conference on Applied Electromagnetics and Communications*. Dubrovnik, Croatia: IEEE, Apr 2006.

[39] Z. Wang, L. Lu, and A. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, Feb 2004.

[40] D. Brunet, "A study of the structural similarity image quality measure with applications to image processing," Ph.D. dissertation, University of Waterloo, 2012.

[41] D. Huffman, "A method for the construction of minimum-redundancy codes," in *Proceedings of I.R.E.* I.R.E, Sept 1952.

[42] J. Rajan, "Text compression by static huffman coding," Apr 2016.

[43] J. Leeuwen, "On the construction of huffman trees," *ICALP*, pp. 382–410, 1976.

[44] K. Brandenburg, "Mp3 and aac explained," in *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, Sept 1999.

[45] "Computer sounds." [Online]. Available: http://soundbible.com/tags-computer.html

[46] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[47] “Hevc/h265.” [Online]. Available: www.h265files.com

[48] S. Pudlewski, A. Prasanna, and T. Melodia, “Compressed-sensing-enabled video streaming for wireless multimedia sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 1060–1072, April 2012.

[49] S. Pudlewski, N. Cen, Z. Guan, and T. Melodia, “Video transmission over lossy wireless networks: A cross-layer perspective,” *IEEE Journal of Selected Topics in Signal Processing*, pp. 6–21, Jul 2014.

[50] “Cvx users’ guide.” [Online]. Available: http://web.cvxr.com/cvx/doc/intro.html#what-is-cvx

Appendix A

Derivation and Plotting of Media Utility Functions

A.1 textfunc.m

```
clear; clc;
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Text Utility Function
% Filename: textfunc.m
% Purpose: This program derives and plots the text utility function.
% (created by Tasnim Rahman).
syms l;
eqn = (998*exp(-l*32000))+(999*exp(-l*32000)) == 1;
lambda = solve(eqn,l) % lambda = 2.3748e-04
x = 00000:100:32000;
f = 1./(1 + exp(-1*lambda*(x)));
plot(x, f, 'r');
title('Utility Function of Text');
```

```
xlabel('Data Rate (kbps)');
ylabel('Utility');
```

A.2 audiofunc.m

```
clear; clc;
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Audio Utility Function
% Filename: audiofunc.m
% Purpose: This program derives and plots the audio utility function.
% (created by Tasnim Rahman).
syms l;
eqn = (998*exp(-l*320000))+(999*exp(-l*320000)) == exp(-l*32000);
lambda = solve(eqn,l) % lambda = 2.6387e-05
x = 00000:1000:600000;
f = 1./(1 + exp(-1*lambda*(x-288000)));
plot(x, f, 'r');
title('Utility Function of Audio');
xlabel('Data Rate (kbps)');
ylabel('Utility');
```

A.3 vidfunc.m

```
clear; clc;
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Video Utility Function
% Filename: vidfunc.m
% Purpose: This program derives and plots the video utility function.
% (created by Tasnim Rahman).
syms l;
eqn = (998*exp(-l*500000000))+(999*exp(-l*500000000)) == exp(-l*120000000);
lambda = solve(eqn,l) % lambda = 1.9998e-07
x = 00000:1000:800000000;
f = 1./(1 + exp(-1*lambda*(x)));
plot(x, f, 'r');
title('Utility Function of Video');
xlabel('Data Rate (kbps)');
ylabel('Utility');
```

Appendix B

Huffman Coding

B.1 fhstart.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Huffman Coding Algorithm
% This program and algorithm is open-sourced and has been modified by
% Tasnim Rahman. The original creator of the this Huffman Coding MATLAB
% Software is Jeny Rajan
% Filename: fhstart.m
% Purpose: This program compresses a text file for data transmission through
% histogram building and probability calculation
%Huffman coding algorithm
% Total program is divided into three modules
% (1) Histogram building & Probability calculation (fhstart.m)
% (2) Huffman Tree drawing phase(fhtree1.m)
% (3) Huffman Code generating module(fhcode.m)

%%%% USAGE %%%%%%%
% Run the program fhstart.m
% The filename entered should be a Text file

clc;
clear all;
k=input('Enter the file name :','s');
fid = fopen(k,'r');
F = fread(fid);
img = char(F');
mx=255;
[x y z]=size(img);
h(1:mx)=0;
disp('Histogram building phase started....');
for i=1:y
        iy=img(i);
        val=double(iy);
        h(val)=h(val)+1;
end
disp('Probability calculating phase started...');
i=1:mx
p(i)=h(i)/(x*y);
j=1;
for i=1:mx
        if(p(i)~=0)
         lst(j)=i;
         lst(j+1)=p(i);
         j=j+2;
        end
 end
[tt,mx]=size(lst);
disp('sorting phase started....');
for i=2:2:mx
    for j=i:2:mx
        if (lst(i)>lst(j))
            temp1=lst(i-1);
            temp2=lst(i);
            lst(i-1)=lst(j-1);
            lst(i)=lst(j);
            lst(j-1)=temp1;
            lst(j)=temp2;
        end
    end
end
disp('Building Huffman Tree.....');
fhtree1(lst,img);
```

B.2 fhtree1.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Huffman Tree
% This program and algorithm is open-sourced and has been modified by
% Tasnim Rahman. The original creator of the this Huffman Coding MATLAB
% Software is Jeny Rajan
% Filename: fhstart.m
% Purpose: This program draws the Huffman tree for compression

%Hufman tree drawing phase
function HT=fhtree1(lst,img)
[p,q]=size(lst);
[tt,mx]=size(lst);
sz1=q;
xx=1;
k1=0;
prt=0;
while (k1<1)
    k1=lst(2)+lst(4);
    prt=prt-1;
    lstn(xx)=lst(1);
    lstn(xx+1)=0;
    lstn(xx+2)=prt;
    xx=xx+3;
    lstn(xx)=lst(3);
    lstn(xx+1)=1;
    lstn(xx+2)=prt;
    xx=xx+3;
    lst(1)=prt;
    lst(2)=k1;
    lst(3)=99;
    lst(4)=99;

for i=2:2:mx
    for j=i:2:mx
        if (lst(i)>lst(j))
            temp1=lst(i-1);
            temp2=lst(i);
            lst(i-1)=lst(j-1);
            lst(i)=lst(j);
            lst(j-1)=temp1;
            lst(j)=temp2;
        end
    end
end
end
lstn(xx)=lst(1);
lstn(xx+1)=lst(2);
lstn(xx+2)=lst(3);
disp('Calling fhcode...');
fhcode(lstn,img)
```

B.3 fhcode.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Huffman Code Generator
% This program and algorithm is open-sourced and has been modified by
% Tasnim Rahman. The original creator of the this Huffman Coding MATLAB
% Software is Jeny Rajan
% Filename: fhstart.m
% Purpose: This program generates the Huffman Code

function HC=fhcode(lstn,img)
disp('Code generating phase entered...');
[lm,ln]=size(lstn);
ntt=ln-1;
[im,in]=size(img);
t=0;
idd=input('Enter destination huffman code file name : ','s');
tab=input('Enter the Huff Table name (for decoding purpose): ','s');
tb = fopen(tab,'w+');
idd=fopen(idd,'w+');
fst1='';
fst2='';
ed=0;
din=0;
% Traversing the Tree resembling list resulting in reverse Huffman code for a
    particular character%
disp('Building Huffman Table.........');
for i=1:in
    k=img(i);
    ftemp=img(i);
    a=0;
    for j=1:3:ln
        if(lstn(j+2)==99)
            break;
        end
        if(lstn(j)==k)
            a=a+1;
            ary(a)=lstn(j+1);
            k=lstn(j+2);
        end
    end
    % Reversing the reverse Huffman Code%
    for b=a:-1:1
        t=t+1;
        hc(t)=ary(b);
        fprintf(idd,'%d',ary(b));
        fst1=int2str(ary(b));
        fst2=strcat(fst2,fst1);
    end
    %Building Huffman Table for Decoding%
    din=0;
    for z=1:ed
        if dict(z)==ftemp
            din=1;
        end
    end
    if din==0
        ed=ed+1;
        dict(ed)=ftemp;
        fprintf(tb,'%c',' ');
        fprintf(tb,'%c',ftemp);
        fprintf(tb,'%s',fst2);
    end
    fst1='';
    fst2='';
end
fclose(tb);
%Converting 8 bit Binary to ASCII character and storing the result in specified file%
disp('Converting binary huffman codes to ASCII characters......');
nme=input('Enter the destination file name :','s');
disp('Generating the compressed file..........');
id = fopen(nme,'w+');
for i=1:8:t
    ck=t-i+1;
    if(ck>8)
        tp=(hc(i:i+7));
        num=8;
    else
        tp=(hc(i:t));
        num=ck;
    end
    temp1=b2d(tp,num);
    temp2=char(temp1);
    fprintf(id,'%c',temp2);
end
fclose(id);
fclose(idd);
disp('Generated Compressed file');
return
```

B.4 b2d.m

```
function PW=b2d(tp,num)
PW=0;
j=0;
for i=num:-1:1
PW=PW+pow2(tp(i),j);
j=j+1;
end
return
```

B.5 fhdecode.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Huffman Code Generator
% This program and algorithm is open-sourced and has been modified by
% Tasnim Rahman. The original creator of the this Huffman Coding MATLAB
% Software is Jeny Rajan
% Filename: fhstart.m
% Purpose: This program decompresses the binary message into a text file after
    transmission
%Huffman decoder for text compression
%When the program asks for file name enter file name given to the destination
    file
clc;
clear all;
%Accepting the decoded file and storing it in character array a%
nme=input('Enter the file name (the ASCII file enterd at compression time) : ','
    s');
id = fopen(nme,'r');
a = fscanf(id,'%c',inf);
fclose(id);
[m,n]=size(a);
k=1;
for i=1:n
    b(i)=double(a(i));
end
%Constructing the original decoded file from the 8 bit block file
%and storing the result in the character array d%
for i=1:n
    c = dec2bin(b(i),8);
    for j=1:8
        d(k)=c(j);
        k=k+1;
    end
end
%Accepting the Huffman Table%
%Enter the table name given at the time of compression
nme2=input('Enter the table name (Table name entered at compression time) : ','s
    ');
id2 = fopen(nme2,'r');
```

```
a2=fscanf(id2,'%c',inf);
fclose(id2);
[m1,n1]=size(a2);
chk=0;
cnt=1;
str='';
temp=0;
%Partiotning the Huffman table into character array and corresponding
%code array-cd1(character),cd2(code)%
for j=1:n1
    if chk==1 & a2(j)~=' '
        str=strcat(str,a2(j));
        cd2{cnt-1}=str;
    end
        if temp==1
        cd1(cnt)=a2(j);
        cnt=cnt+1;
        chk=1;
        temp=0;
        str='';
        end

        if a2(j)==' '
            temp=1;
            chk=0;
            if j>1
                if a2(j-1)==' '
                    chk=1;
                    temp=0;
                    str='';
                end
            end

        end
end
%reconstructing original text from huffman code%
[m2,n2]=size(d);
```

```
[m3,n3]=size(cd2);
% Enter a file name to deliver the decoded output
nme=input('Enter the file name (to produce output) :','s');
id = fopen(nme,'w+');
comp='';
tap=0;
disp('Decompression starts.........');
for i=1:n2
      cnt=0;
      z1=d(i);
      m=num2str(z1);
      for j=1:n3
          k=strcmp(m,cd2(j));
          if(k==1 & tap==0)
              fprintf(id,'%c',cd1(j));
              comp='';
              cnt=1;
          end
       end

       if(cnt==0)
            comp=strcat(comp,num2str(z1));
            tap=1;
            for j=1:n3
                m=cd2(j);
                k=strcmp(comp,cd2(j));
                if(k==1)
                    cd1(j);
                    fprintf(id,'%c',cd1(j));
                    comp='';
                    tap=0;
                end
            end
        end
    end
disp('Decompression Over');
fclose(id);
```

Appendix C

Audio Quality Programs on MATLAB

C.1 PEAQTest.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: PEAQ Algorithm
% This program and algorithm is open-sourced and has been modified by
% Tasnim Rahman. The original creator of the PEAQ Algorithm is Peter Kabal
% of the Multimedia Signal Processing Lab at McGill University.
% Filename: PEAQTest.m
% Purpose: This program uses the PEAQ Algorithm for audio and noise
% perception in order to determine the quality of an audio sample by
% comparing a test audio sample to a reference audio sample (ideal audio
% sample). For this thesis, the reference audio sample has a bit rate of
% 256 kilobits per second. This file inputs the test and reference audio
% samples for the PEAQ Algorithm in order to calculate the Objective
% Difference Grade (ODG) Value,  Model Output Variable (MOVB) vector and
% the Normalized Audio Scale Value (norm)
% (created by Tasnim Rahman).

clear; clc;

% Input Audio Samples
ref = 'track_256k.mp3' % Reference Audio Sample; 16 times oversampling
test = 'track_320k.mp3' % Tested Audio Sample; 4 times oversampling

% ODG Value and MOVB Vector
% The ODG value is a quality measure acquired from listening tests from
% human participants. The MOVB vector provides potention insight into
% different psychoacoustic dimensions. The 11 parameters of the MOVB vector
% define: bandwidth of the reference signal, bandwidth of the test signal,
% logarithm of the averaged total noise-to-mask ratio, windowed averaged
% difference in modulation between reference and test signal, average
% distorted block, harmonic structure of the error over time, averaged
% modulation difference, averaged modulated difference with empahsis on new
% modulations and modulation changes, RMS value of the averaged noise
% loudness, maximum of the probability of detection, and relative fraction
% of frames.
[odg, movb] = PQevalAudio_fn(ref, test)

% Preliminary calculation for Normalized Audio Scale Value
uni = (odg + 5.0) / 5.0;

% Calculate the Normalized Audio Scale Value for the quality parameters
% used for cross-layer network optimization
if uni > 1
    norm = 1.000
elseif uni <= 1
    norm = (odg + 5.0 / 5.0)
end
```

C.2 PQevalAudio.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: PEAQ Algorithm
% This program and algorithm is open-sourced and has been modified by
% Tasnim Rahman. The original creator of the PEAQ Algorithm is Peter Kabal
% of the Multimedia Signal Processing Lab at McGill University.
% Filename: PQevalAudio.m
function PQevalAudio (Fref, Ftest, StartS, EndS)
% Perceptual evaluation of audio quality.

% - StartS shifts the frames, so that the first frame starts at that sample.
%   This is a two element array, one element for each input file. If StartS is
%   a scalar, it applies to both files.
% - EndS marks the end of data. The processing stops with the last frame that
%   contains that sample. This is a two element array, one element for each
%   input file.  If EndS is as scalar, it applies to both files.

% P. Kabal $Revision: 1.2 $  $Date: 2004/02/05 04:25:24 $

% Globals (to save on copying in/out of functions)
global MOVC PQopt

% Analysis parameters
NF = 2048;
Nadv = NF / 2;
Version = 'Basic';

% Options
PQopt.ClipMOV = 0;
PQopt.PCinit = 0;
PQopt.PDfactor = 1;
PQopt.Ni = 1;
PQopt.DelayOverlap = 1;
PQopt.DataBounds = 1;
PQopt.EndMin = NF / 2;

addpath ('CB', 'MOV', 'Misc', 'Patt');

if (nargin < 3)
    StartS = [0, 0];
end
if (nargin < 4)
    EndS = [];
end

% Get the number of samples and channels for each file
WAV(1) = PQwavFilePar (Fref);
WAV(2) = PQwavFilePar (Ftest);

% Reconcile file differences
PQ_CheckWAV (WAV);
if (WAV(1).Nframe ~= WAV(2).Nframe)
    disp ('>>> Number of samples differ: using the minimum');
end

% Data boundaries
Nchan = WAV(1).Nchan;
[StartS, Fstart, Fend] = PQ_Bounds (WAV, Nchan, StartS, EndS, PQopt);

% Number of PEAQ frames
Np = Fend - Fstart + 1;
if (PQopt.Ni < 0)
    PQopt.Ni = ceil (Np / abs(PQopt.Ni));
end

% Initialize the MOV structure
MOVC = PQ_InitMOVC (Nchan, Np);

% Initialize the filter memory
Nc = PQCB (Version);
for (j = 0:Nchan-1)
```

```
    Fmem(j+1) = PQinitFMem (Nc, PQopt.PCinit);
end

is = 0;
for (i = -Fstart:Np-1)

    % Read a frame of data
    xR = PQgetData (WAV(1), StartS(1) + is, NF);    % Reference file
    xT = PQgetData (WAV(2), StartS(2) + is, NF);    % Test file
    is = is + Nadv;

    % Process a frame
    for (j = 0:Nchan-1)
        [MOVI(j+1), Fmem(j+1)] = PQeval (xR(j+1,:), xT(j+1,:), Fmem(j+1));
    end

    if (i >= 0)
        % Move the MOV precursors into a new structure
        PQframeMOV (i, MOVI);   % Output is in global MOVC

        % Print the MOV precursors
        if (PQopt.Ni ~= 0 & mod (i, PQopt.Ni) == 0)
            PQprtMOVCi (Nchan, i, MOVC);
        end
    end
end

% Time average of the MOV values
if (PQopt.DelayOverlap)
    Nwup = Fstart;
else
    Nwup = 0;
end
MOVB = PQavgMOVB (MOVC, Nchan, Nwup);

% Neural net
ODG = PQnNet (MOVB);
```

```
% Summary printout
PQprtMOV (MOVB, ODG);

%----------
function PQ_CheckWAV (WAV)
% Check the file parameters

Fs = 48000;

if (WAV(1).Nchan ~= WAV(2).Nchan)
    error ('>>> Number of channels differ');
end
if (WAV(1).Nchan > 2)
    error ('>>> Too many input channels');
end
if (WAV(1).Nframe ~= WAV(2).Nframe)
    disp ('>>> Number of samples differ');
end
if (WAV(1).Fs ~= WAV(2).Fs)
    error ('>>> Sampling frequencies differ');
end
if (WAV(1).Fs ~= Fs)
    error ('>>> Invalid Sampling frequency: only 48 kHz supported');
end

%----------
function [StartS, Fstart, Fend] = PQ_Bounds (WAV, Nchan, StartS, EndS, PQopt)

PQ_NF = 2048;
PQ_NADV = (PQ_NF / 2);

if (isempty (StartS))
    StartS(1) = 0;
    StartS(2) = 0;
elseif (length (StartS) == 1)
    StartS(2) = StartS(1);
```

```
end
Ns = WAV(1).Nframe;

% Data boundaries (determined from the reference file)
if (PQopt.DataBounds)
    Lim = PQdataBoundary (WAV(1), Nchan, StartS(1), Ns);
    fprintf ('PEAQ Data Boundaries: %ld (%.3f s) - %ld (%.3f s)\n', ...
             Lim(1), Lim(1)/WAV(1).Fs, Lim(2), Lim(2)/WAV(1).Fs);
else
    Lim = [Starts(1), StartS(1) + Ns - 1];
end

% Start frame number
Fstart = floor ((Lim(1) - StartS(1)) / PQ_NADV);

% End frame number
Fend = floor ((Lim(2) - StartS(1) + 1 - PQopt.EndMin) / PQ_NADV);

%----------
function MOVC = PQ_InitMOVC (Nchan, Np)
MOVC.MDiff.Mt1B = zeros (Nchan, Np);
MOVC.MDiff.Mt2B = zeros (Nchan, Np);
MOVC.MDiff.Wt   = zeros (Nchan, Np);

MOVC.NLoud.NL   = zeros (Nchan, Np);

MOVC.Loud.NRef  = zeros (Nchan, Np);
MOVC.Loud.NTest = zeros (Nchan, Np);

MOVC.BW.BWRef  = zeros (Nchan, Np);
MOVC.BW.BWTest = zeros (Nchan, Np);

MOVC.NMR.NMRavg = zeros (Nchan, Np);
MOVC.NMR.NMRmax = zeros (Nchan, Np);

MOVC.PD.Pc = zeros (1, Np);
MOVC.PD.Qc = zeros (1, Np);
MOVC.EHS.EHS = zeros (Nchan, Np);
```

Appendix D

Video Quality Programs on MATLAB

D.1 demo.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: SSIM and PSNR Values
% This is an implementation of the algorithm for calculating the
% Structural SIMilarity (SSIM) index between two images
% Please refer to the following paper and the website with suggested usage
% Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image
% quality assessment: From error visibility to structural similarity,"
% IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612,
% Apr. 2004. University. This program is open-sourced and written by
% Mahmoud Afifi from Assiut University, Egyptand. It has been modified by
% Tasnim Rahman
% Filename: demo.m
% Purpose: This program take two raw frames from video samples in order to
% determine the quality of the video. One of the frames is a reference
% frame with ideal video quality and the other frame is from the video that
% is being tested in order to determine its quality through the Structural
% Similarity Index (SSIM) and the Peak Signal-to-Noise Ratio (PSNR)

image1=imread('SampleFrame_50000k.tif'); % Tested Video Frame
image2=imread('SampleFrame_50000k.tif'); % Reference Video Frame
psnr_=getPSNR(image1,image2); % Calculate the PSNR Value
ssim_=getMSSIM(image1,image2); % Calculate the SSIM Value
fprintf('PSNR= %f - SSIM= %f\n',psnr_,ssim_); % Print out the PSNR and SSIM
    Values
```

D.2 ssim.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: SSIM and PSNR Values
% This is an implementation of the algorithm for calculating the
% Structural SIMilarity (SSIM) index between two images
% Please refer to the following paper and the website with suggested usage
% Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image
% quality assessment: From error visibility to structural similarity,"
% IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612,
% Apr. 2004. University. This program is open-sourced and modified by
% Tasnim Rahman
% Filename: ssim.m
function [mssim, ssim_map] = ssim(img1, img2, K, window, L)

% ========================================================================
% SSIM Index with automatic downsampling, Version 1.0
% Copyright(c) 2009 Zhou Wang
% All Rights Reserved.
```

```
%
% ----------------------------------------------------------------------
% Permission to use, copy, or modify this software and its documentation
% for educational and research purposes only and without fee is hereby
% granted, provided that this copyright notice and the original authors'
% names appear on all copies and supporting documentation. This program
% shall not be used, rewritten, or adapted as the basis of a commercial
% software or hardware product without first obtaining permission of the
% authors. The authors make no representations about the suitability of
% this software for any purpose. It is provided "as is" without express
% or implied warranty.
%----------------------------------------------------------------------
%
% This is an implementation of the algorithm for calculating the
% Structural SIMilarity (SSIM) index between two images
%
% Please refer to the following paper and the website with suggested usage
%
% Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image
% quality assessment: From error visibility to structural similarity,"
% IEEE Transactios on Image Processing, vol. 13, no. 4, pp. 600-612,
% Apr. 2004.
%
% http://www.ece.uwaterloo.ca/~z70wang/research/ssim/
%
% Note: This program is different from ssim_index.m, where no automatic
% downsampling is performed. (downsampling was done in the above paper
% and was described as suggested usage in the above website.)
%
% Kindly report any suggestions or corrections to zhouwang@ieee.org
%
%----------------------------------------------------------------------
%
%Input : (1) img1: the first image being compared
%        (2) img2: the second image being compared
%        (3) K: constants in the SSIM index formula (see the above
%            reference). defualt value: K = [0.01 0.03]
%        (4) window: local window for statistics (see the above
%            reference). default widnow is Gaussian given by
%            window = fspecial('gaussian', 11, 1.5);
%        (5) L: dynamic range of the images. default: L = 255
%
%Output: (1) mssim: the mean SSIM index value between 2 images.
%            If one of the images being compared is regarded as
%            perfect quality, then mssim can be considered as the
%            quality measure of the other image.
%            If img1 = img2, then mssim = 1.
%        (2) ssim_map: the SSIM index map of the test image. The map
%            has a smaller size than the input images. The actual size
%            depends on the window size and the downsampling factor.
%
%Basic Usage:
%   Given 2 test images img1 and img2, whose dynamic range is 0-255
%
%   [mssim, ssim_map] = ssim(img1, img2);
%
%Advanced Usage:
%   User defined parameters. For example
%
%   K = [0.05 0.05];
%   window = ones(8);
%   L = 100;
%   [mssim, ssim_map] = ssim(img1, img2, K, window, L);
%
%Visualize the results:
%
%   mssim                        %Gives the mssim value
%   imshow(max(0, ssim_map).^4)  %Shows the SSIM index map
%========================================================================


if (nargin < 2 || nargin > 5)
   mssim = -Inf;
   ssim_map = -Inf;
   return;
end

if (size(img1) ~= size(img2))
   mssim = -Inf;
   ssim_map = -Inf;
   return;
end

[M N] = size(img1);

if (nargin == 2)
   if ((M < 11) || (N < 11))
       mssim = -Inf;
       ssim_map = -Inf;
      return
   end
   window = fspecial('gaussian', 11, 1.5);  %
   K(1) = 0.01;                  % default settings
   K(2) = 0.03;                  %
   L = 255;                                      %
end

if (nargin == 3)
   if ((M < 11) || (N < 11))
       mssim = -Inf;
       ssim_map = -Inf;
      return
   end
   window = fspecial('gaussian', 11, 1.5);
   L = 255;
   if (length(K) == 2)
      if (K(1) < 0 || K(2) < 0)
           mssim = -Inf;
        ssim_map = -Inf;
        return;
      end
   else
       mssim = -Inf;
    ssim_map = -Inf;
       return;
   end
end

if (nargin == 4)
   [H W] = size(window);
   if ((H*W) < 4 || (H > M) || (W > N))
       mssim = -Inf;
       ssim_map = -Inf;
      return
   end
   L = 255;
   if (length(K) == 2)
      if (K(1) < 0 || K(2) < 0)
           mssim = -Inf;
        ssim_map = -Inf;
        return;
      end
   else
       mssim = -Inf;
    ssim_map = -Inf;
       return;
   end
end

if (nargin == 5)
   [H W] = size(window);
   if ((H*W) < 4 || (H > M) || (W > N))
       mssim = -Inf;
       ssim_map = -Inf;
      return
   end
   if (length(K) == 2)
      if (K(1) < 0 || K(2) < 0)
```

```
           mssim = -Inf;
        ssim_map = -Inf;
        return;
      end
   else
       mssim = -Inf;
    ssim_map = -Inf;
       return;
   end
end


img1 = double(img1);
img2 = double(img2);

% automatic downsampling
f = max(1,round(min(M,N)/256));
%downsampling by f
%use a simple low-pass filter
if(f>1)
    lpf = ones(f,f);
    lpf = lpf/sum(lpf(:));
    img1 = imfilter(img1,lpf,'symmetric','same');
    img2 = imfilter(img2,lpf,'symmetric','same');

    img1 = img1(1:f:end,1:f:end);
    img2 = img2(1:f:end,1:f:end);
end

C1 = (K(1)*L)^2;
C2 = (K(2)*L)^2;
window = window/sum(sum(window));

mu1   = filter2(window, img1, 'valid');
mu2   = filter2(window, img2, 'valid');
mu1_sq = mu1.*mu1;
mu2_sq = mu2.*mu2;
```

```
mu1_mu2 = mu1.*mu2;
sigma1_sq = filter2(window, img1.*img1, 'valid') - mu1_sq;
sigma2_sq = filter2(window, img2.*img2, 'valid') - mu2_sq;
sigma12 = filter2(window, img1.*img2, 'valid') - mu1_mu2;

if (C1 > 0 && C2 > 0)
   ssim_map = ((2*mu1_mu2 + C1).*(2*sigma12 + C2))./((mu1_sq + mu2_sq + C1).*(
       sigma1_sq + sigma2_sq + C2));
else
   numerator1 = 2*mu1_mu2 + C1;
   numerator2 = 2*sigma12 + C2;
    denominator1 = mu1_sq + mu2_sq + C1;
   denominator2 = sigma1_sq + sigma2_sq + C2;
   ssim_map = ones(size(mu1));
   index = (denominator1.*denominator2 > 0);
   ssim_map(index) = (numerator1(index).*numerator2(index))./(denominator1(index
       ).*denominator2(index));
   index = (denominator1 ~= 0) & (denominator2 == 0);
   ssim_map(index) = numerator1(index)./denominator1(index);
end

mssim = mean2(ssim_map);

return
```

D.3 getMSSIM.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: SSIM and PSNR Values
% This is an implementation of the algorithm for calculating the
% Structural SIMilarity (SSIM) index between two images
% Please refer to the following paper and the website with suggested usage
% Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image
% quality assessment: From error visibility to structural similarity,"
% IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612,
% Apr. 2004. University. This program is open-sourced and written by
% Mahmoud Afifi from Assiut University, Egyptand. It has been modified by
% Tasnim Rahman
% Filename: getMSSIM.m
function mssim=getMSSIM(frameReference,frameUnderTest)
%Written by: Mahmoud Afifi ~ Assiut University, Egypt
%Reference: Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli,
%Image quality assessment: From error visibility to structural similarity,
%IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004

%///////////////////////////////// INITS  ////////////////////////////////
C1 = 6.5025;
C2 = 58.5225;
frameReference=double(frameReference);
frameUnderTest=double(frameUnderTest);
frameReference_2=frameReference.^2;
frameUnderTest_2=frameUnderTest.^2;
frameReference_frameUnderTest=frameReference.*frameUnderTest;
%///////////////////////////////// PRELIMINARY COMPUTING
    ////////////////////////////////
mu1=imgaussfilt(frameReference,1.5);
mu2=imgaussfilt(frameUnderTest,1.5);
mu1_2=mu1.^2;
mu2_2=mu2.^2;
mu1_mu2=mu1.*mu2;

sigma1_2=imgaussfilt(frameReference_2,1.5);
sigma1_2=sigma1_2-mu1_2;
sigma2_2=imgaussfilt(frameUnderTest_2,1.5);
sigma2_2=sigma2_2-mu2_2;
sigma12=imgaussfilt(frameReference_frameUnderTest,1.5);
sigma12=sigma12-mu1_mu2;
```

```
%///////////////////////////////// FORMULA ////////////////////////////////

t3 = ((2*mu1_mu2 + C1).*(2*sigma12 + C2));
t1 =((mu1_2 + mu2_2 + C1).*(sigma1_2 + sigma2_2 + C2));
ssim_map =  t3./t1;
mssim = mean2(ssim_map); mssim=mean(mssim(:));
```

D.4 getPSNR.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: SSIM and PSNR Values
% This is an implementation of the algorithm for calculating the
% Structural SIMilarity (SSIM) index between two images
% Please refer to the following paper and the website with suggested usage
% Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image
% quality assessment: From error visibility to structural similarity,"
% IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612,
% Apr. 2004. University. This program is open-sourced and written by
% Mahmoud Afifi from Assiut University, Egyptand. It has been modified by
% Tasnim Rahman
% Filename: getPSNR.m
function psnr=getPSNR(frameReference,frameUnderTest)
s1=double(frameReference-frameUnderTest).^2;

    s = sum(sum(s1));
    sse = s(:,:,1)+s(:,:,2)+s(:,:,3);
    if( sse <= 1e-10)
        psnr=0;
    else
        mse  = sse / double(size(frameReference,1)*size(frameReference,2)*size(
            frameReference,3));
        psnr = 10.0 * log10((255 * 255) / mse);
    end
end
```

Appendix E

Plotting Quality Versus Bit Rate in MATLAB

E.1 audioqualitygraph.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Graph Audio Quality
% Filename: audioqualitygraph.m
% Purpose: This MATLAB program graphs the Normalized Audio Scale values
% from PEAQTest.m for the bit rates.

BitRates = [0, 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256];
Norm = [0, 0.2874, 0.3637, 0.3554, 0.3317, 0.3308, 0.3609, 0.4029, 0.5042, 0
    .6370, 0.8067, 0.8638, 0.9348, 1.000];

figure

plot(BitRates,Norm, '-o')
```

```
title('Normalized Audio Scale Value based on Bit Rates for MP3');
xlabel('Bit Rate [kilobits per second]');
ylabel('Normalized Audio Quality')
```

E.2 videoqualitygraph.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Graph Video Quality
% Filename: videoqualitygraph.m
% Purpose: This MATLAB program graphs the Structural Similarity Index
% Values (SSIM) and Peak Signal-to-Noise Ratio (PSNR) Values
% from demo.m for the bit rates.

BitRates = [12000, 20000, 30000, 50000];
BitRate2= [12000, 20000, 30000];
SSIM = [0.992498, 0.994385, 0.995661, 1.0000000];
PSNR = [52.653395, 54.082003, 55.251703];

figure

plot(BitRates,SSIM, '-o')

title('SSIM for HEVC');
xlabel('Bit Rate [kilobits per second]');
ylabel('SSIM')

figure

plot(BitRate2,PSNR, '-o')
```

```
title('PSNR for HEVC');
xlabel('Bit Rate [kilobits per second]');
ylabel('PSNR')
```

Appendix F

Rate-Distortion Curves

F.1 audiodistortion.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Graph Audio Rate-Distortion
% Filename: audiodistortion.m
% Purpose: This MATLAB program graphs the Rate-Distortion for Audio
% specifically MP3 based on the thesis.

function createfigure(X1, Y1)
%CREATEFIGURE(X1, Y1)
%  X1:  vector of x data
%  Y1:  vector of y data

%  Auto-generated by MATLAB on 19-Jul-2019 15:57:47

% Create figure
figure1 = figure;
```

```
% Create axes
axes1 = axes('Parent',figure1);
hold(axes1,'on');

% Create plot
plot1 = plot(X1,Y1,'DisplayName','Normalized Audio Scale Curve',...
    'Marker','o');

% Get xdata from plot
xdata1 = get(plot1, 'xdata');
% Get ydata from plot
ydata1 = get(plot1, 'ydata');
% Make sure data are column vectors
xdata1 = xdata1(:);
ydata1 = ydata1(:);


% Remove NaN values and warn
nanMask1 = isnan(xdata1(:)) | isnan(ydata1(:));
if any(nanMask1)
    warning('GeneratedCode:IgnoringNaNs', ...
        'Data points with NaN coordinates will be ignored.');
    xdata1(nanMask1) = [];
    ydata1(nanMask1) = [];
end

% Find x values for plotting the fit based on xlim
axesLimits1 = xlim(axes1);
xplot1 = linspace(axesLimits1(1), axesLimits1(2));

% Find coefficients for polynomial (order = 3)
fitResults1 = polyfit(xdata1,ydata1,3);
% Evaluate polynomial
yplot1 = polyval(fitResults1,xplot1);
% Plot the fit
fitLine1 = plot(xplot1,yplot1,'DisplayName','   cubic','Tag','cubic',...
    'Parent',axes1,...
    'Color',[0.466 0.674 0.188]);

% Set new line in proper position
setLineOrder(axes1,fitLine1,plot1);

% Create ylabel
ylabel('Normalized Audio Quality');

% Create xlabel
xlabel('Bit Rate [kilobits per second]');

% Create title
title('Bit Rate-Distortion for MP3');

box(axes1,'on');
% Create legend
legend(axes1,'show');

%-------------------------------------------------------------------------%
function setLineOrder(axesh1, newLine1, associatedLine1)
%SETLINEORDER(AXESH1,NEWLINE1,ASSOCIATEDLINE1)
%  Set line order
%  AXESH1:  axes
%  NEWLINE1:  new line
%  ASSOCIATEDLINE1:  associated line

% Get the axes children
hChildren = get(axesh1,'Children');
% Remove the new line
hChildren(hChildren==newLine1) = [];
% Get the index to the associatedLine
lineIndex = find(hChildren==associatedLine1);
% Reorder lines so the new line appears with associated data
hNewChildren = [hChildren(1:lineIndex-1);newLine1;hChildren(lineIndex:end)];
% Set the children:
set(axesh1,'Children',hNewChildren);
```

F.2 videodistortion.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 19 July 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Graph Video Rate-Distortion
% Filename: videodistortion.m
% Purpose: This MATLAB program graphs the Rate-Distortion for Video
% specifically HEVC based on the thesis

function createfigure(X1, Y1)
%CREATEFIGURE(X1, Y1)
%  X1:  vector of x data
%  Y1:  vector of y data

%  Auto-generated by MATLAB on 19-Jul-2019 16:02:27

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
hold(axes1,'on');

% Create plot
plot1 = plot(X1,Y1,'DisplayName','SSIM Value','Marker','o');

% Get xdata from plot
xdata1 = get(plot1, 'xdata');
% Get ydata from plot
ydata1 = get(plot1, 'ydata');
% Make sure data are column vectors
xdata1 = xdata1(:);
```

```
ydata1 = ydata1(:);


% Remove NaN values and warn
nanMask1 = isnan(xdata1(:)) | isnan(ydata1(:));
if any(nanMask1)
    warning('GeneratedCode:IgnoringNaNs', ...
        'Data points with NaN coordinates will be ignored.');
    xdata1(nanMask1) = [];
    ydata1(nanMask1) = [];
end

% Find x values for plotting the fit based on xlim
axesLimits1 = xlim(axes1);
xplot1 = linspace(axesLimits1(1), axesLimits1(2));

% Find coefficients for polynomial (order = 1)
fitResults1 = polyfit(xdata1,ydata1,1);
% Evaluate polynomial
yplot1 = polyval(fitResults1,xplot1);
% Plot the fit
fitLine1 = plot(xplot1,yplot1,'DisplayName','   linear','Tag','linear',...
    'Parent',axes1,...
    'Color',[0.929 0.694 0.125]);

% Set new line in proper position
setLineOrder(axes1,fitLine1,plot1);

% Create ylabel
ylabel('SSIM');

% Create xlabel
xlabel('Bit Rate [kilobits per second]');

% Create title
title('Bit Rate-Distortion for HEVC');
```

```
box(axes1,'on');
% Create legend
legend1 = legend(axes1,'show');
set(legend1,...
    'Position',[0.161309528652401 0.79722222435096 0.253571423728551 0
        .0821428550141198]);

%-------------------------------------------------------------------------%
function setLineOrder(axesh1, newLine1, associatedLine1)
%SETLINEORDER(AXESH1,NEWLINE1,ASSOCIATEDLINE1)
%  Set line order
%  AXESH1:  axes
%  NEWLINE1:  new line
%  ASSOCIATEDLINE1:  associated line

% Get the axes children
hChildren = get(axesh1,'Children');
% Remove the new line
hChildren(hChildren==newLine1) = [];
% Get the index to the associatedLine
lineIndex = find(hChildren==associatedLine1);
% Reorder lines so the new line appears with associated data
hNewChildren = [hChildren(1:lineIndex-1);newLine1;hChildren(lineIndex:end)];
% Set the children:
set(axesh1,'Children',hNewChildren);
```

F.3 convexoptimp.m

```
% Name: Tasnim Rahman
% Advisor: Dr. Alexander Wyglinski and Dr. Scott Pudlewski
% Date Last Modified: 10 August 2019
% Thesis: Cross-Layer Network Optimization based on Application
% Requirements
% MATLAB Program: Convex Optimization for Air Force Scenario
% Filename: convexoptimp.m
% Purpose: This MATLAB program utilizes the CVX program to find the
% optimal solution (optimal data rate) for the NUM problem for audio
% of the Air Force scenario.
s = 1;
cvx_begin
    variables x(s)
    maximize((0.72/(1+(exp(-(2.6387*10^(-5))*(x-288000))))))
    subject to
        x <= 320000
        x >= 32000
cvx_end

% Solution to the NUM problem
% >> convexoptimp

% Successive approximation method to be employed.
%  SDPT3 will be called several times to refine the solution.
%  Original size: 11 variables, 5 equality constraints
%   2 exponentials add 16 variables, 10 equality constraints
%-----------------------------------------------------------------
% Cones  |             Errors              |
%Mov/Act | Centering  Exp cone   Poly cone | Status
%--------+---------------------------------+---------
%  2/  2 | 1.135e+00  9.485e-02  1.526e-08 | Solved
%  2/  2 | 6.355e-02  2.833e-04  1.009e-08 | Solved
%  2/  2 | 3.575e-03  9.054e-07  1.122e-08 | Solved
%  0/  1 | 1.959e-04  1.399e-08  1.131e-08 | Solved
%-----------------------------------------------------------------
%Status: Solved
%Optimal value (cvx_optval): +0.503559
% Multiply optimal solution by 10^5 to scale for the optimal data rate solution
%>> cvx_optval*10^5

%ans =

%  5.0356e+04
```