NTIA Report 15-511

# An Algorithmic Approach to Optimizing Network Resources for Public Safety LTE Networks

**William Kozma Jr.**
**Joel Dumke**
**Brent Johnson**


*report series*

**U.S. DEPARTMENT OF COMMERCE • National Telecommunications and Information Administration**

NTIA Report 15-511

# An Algorithmic Approach to Optimizing Network Resources for Public Safety LTE Networks

**William Kozma Jr.**
**Joel Dumke**
**Brent Johnson**


## U.S. DEPARTMENT OF COMMERCE

December 2014

## DISCLAIMER

Certain commercial equipment and materials are identified in this report to specify adequately the technical aspects of the reported results. In no case does such identification imply recommendation or endorsement by the National Telecommunications and Information Administration, nor does it imply that the material or equipment identified is the best available for this purpose.

## CONTENTS

## FIGURES

## TABLES

## ABBREVIATIONS/ACRONYMS

| | |
|---|---|
| 3GPP | 3rd Generation Partnership Project |
| CIF | Common Intermediate Format |
| CP | Cyclic Prefix |
| DL | Downlink |
| E-UTRAN | Evolved UMTS Terrestrial Radio Access Network |
| eNodeB | Evolved Node B |
| FirstNet | First Responder Network Authority |
| ICM | Iterated Conditional Modes |
| IP | Internet Protocol |
| ISI | Inter Symbol Interference |
| LTE | Long Term Evolution |
| MCMC | Markov Chain Monte Carlo |
| OFDM | Orthogonal Frequency-Division Multiplexing |
| QAM | Quadrature Amplitude Modulation |
| QoS | Quality of Service |
| RE | Resource Element |
| RB | Resource Block |
| TTI | Transmission Time Interval |
| UE | User Equipment |
| UL | Uplink |
| UMTS | Universal Mobile Telecommunications System |
| VGA | Video Graphics Array |

## EXECUTIVE SUMMARY

The deployment of Long Term Evolution (LTE) networks and equipment within the public safety field will have dramatic effects on how public safety personnel perform their jobs. The LTE-provided, high-bandwidth data will allow for numerous new services such as video streaming, interactive mobile applications, and remote file access/delivery. Without proper management of these new network resources, they can quickly become saturated, causing congestion through the network. This is particularly true of streaming video services. Users may choose HD quality video streams that can quickly crowd out all other communication, even though a lower quality video stream would suffice for the purpose.

This report develops a method to optimize the scheduling and allocation of network resources for video streaming. We build on our previous efforts that identify a relationship between video bitrate and utility. We have shown that for certain sets of video characteristics (such as lighting, motion, and target size), at higher bitrates there are diminishing returns to increasing the bitrate, while at lower bitrates utility drops quickly with decreases in bitrate. This allowed the creation of a set of relationship curves and bitrate recommendations for certain video characteristics.

We present a mathematical model of an LTE network and introduce the concept of network usefulness, which is a measure of the cumulative value provided by the network to its users for a given network resource allocation. We compute usefulness by summing the utility provided to a user of a video stream and weighting it against the user's priority for all users in the network. The utility allows the model to incorporate the value of a given video stream, which was assessed through past experiments using subjective testing of actual public safety practitioners.

The priority term allows us to incorporate the concept of local control. This serves two important purposes. The first is that having local control over their own network is important to public safety practitioners. The second purpose is that it allows us to build a better algorithm, allowing it to adapt to the specific scenario at hand, instead of designing our algorithm around the dubious assumption that all users and video streams are of equal importance. We can leverage the inputs of the public safety personnel at the scene who have the most knowledge of the incident to generate a resource allocation that can best optimize the network for their use.

With our network model defined, we borrow established optimization algorithms (Iterated Conditional Modes and the Metropolis-Hastings Algorithm) and concepts (Simulated Annealing) from image processing, showing how they can be applied directly to this problem space with minimum alteration. These algorithms allow us to search a high-dimensional solution space in an efficient manner for an optimal resource allocation for the network scheduler.

We simulate our proposed algorithms in an LTE network, varying the network congestion, user priorities, and other system variables. We show that a network resource scheduler based on the Metropolis-Hastings Algorithm allows for local control while also providing scaling and performance benefits.

# AN ALGORITHMIC APPROACH TO OPTIMIZING NETWORK RESOURCES FOR PUBLIC SAFETY LTE NETWORKS

William Kozma Jr., Joel Dumke, Brent Johnson[1]

This report examines the problem of how to allocate limited network resources in a public safety LTE network for the purpose of disseminating video streams to end users. We develop a mathematical model of the network environment, including the definition of a new metric called usefulness. We then apply known optimization techniques and algorithms to our model and analyze the results. We incorporate the concept of local control for public safety practitioners into our model through the use of a priority function and simulate its performance.

Keywords: LTE, network optimization, resource scheduling, video, wireless network

## 1. INTRODUCTION

The roll-out of public safety Long Term Evolution (LTE) wireless networks will have a profound effect on how public safety personnel perform their missions. Current network technologies only support voice, although some networks have been updated to provide low-bandwidth data services to User Equipment (UE). The new LTE network will have the ability to support high-bandwidth applications such as video streaming, fundamentally transforming the way public safety practitioners do their jobs.

Although these LTE networks have the potential to provide large increases in data services to many users, inefficient use of the spectrum can quickly limit their potential. Due to the dynamic and complex nature of the radio spectrum, scheduling who gets which resource, when they get it, and for what duration, is a non-trivial task. The eNodeB in the LTE network architecture is tasked with analyzing the current state of the network and making these scheduling decisions.

Currently, such algorithms are developed by the eNodeB manufactures themselves, and kept as proprietary technology. This limits our understanding of how they evaluate the network and our ability to analyze their usefulness in a public safety network. While we have no reason to doubt that they are optimized for commercial telecommunications, the requirements of public safety could possibly cause inefficiencies within the algorithms. However, the black box nature of the algorithms makes it impossible to truly perform such an analysis.

We look to develop an LTE resource scheduler that is optimized for the requirements of a public safety LTE network. Specifically, we address the issue of delivering multiple video streams to public safety practitioners. Our work is broken into two main efforts: 1) identifying an efficient optimization algorithm for our network model, and 2) extending our model to support local control through use of a priority function.

[1] The authors are with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, CO 80305.

Section 2 provides a brief background on LTE systems. Section 3 defines a mathematical model of the LTE network and presents our utility functions. We present three optimization algorithms in Section 4**,** simulating their performance in Section 5 and presenting the results in Section 6. We present conclusions and recommendations for future work in Section 7.

## 2. BACKGROUND

This section provides a brief overview of LTE networks. The complexity and detail of LTE networks, even the basics, are beyond the scope of this paper, and thus we only address the concepts that are directly relevant to our work.

The LTE standard [2] was released by 3GPP to meet a few key requirements, chief among them being:

- Decreased connection time and transmission latency
- Increased data rates for users
- Improved spectral efficiency, increased bit-rate at the cell boundaries
- Flexible spectrum usage in new and existing bands
- Simple network architecture
- Improved power consumption by mobile terminals

LTE is an IP based, pure packet-switched network, and as such, no central controller is required; instead LTE uses an Evolved Packet Core (EPC) for its core network. The EPC transports data in packets without ever establishing a physical connection via dedicated circuits [1]. The network consists of a base station (eNodeB), User Equipment, and the EPC. With this configuration, connection and handover times are faster than with previous cellular techniques. Communication and decisions between UE and base stations are also faster.

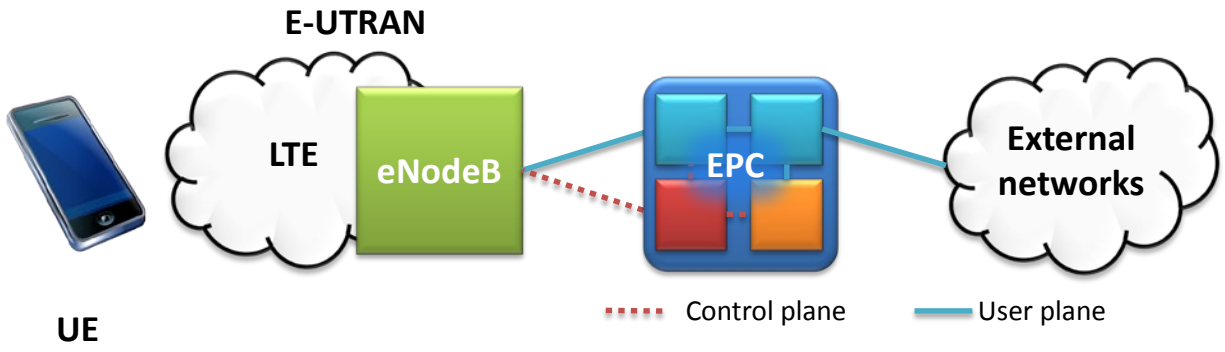



Figure 1. A simplified diagram of the LTE network environment.

LTE incorporates an adaptive scheduler into the eNodeB, which enables the rapid adjustment and efficient utilization of network resources, within its Quality of Service (QoS) [3] requirements, both in the uplink (UL) and downlink phase (DL). One component of this is the Transmission Time Interval (TTI), which for LTE is only 1 millisecond (ms). Decreasing the TTI in LTE further improved system latency, helping to meet one of the design criteria. A principal benefit of the packet switched network is the ability to schedule user(s) in multiple dimensions: time and frequency. Both of these dimensions are described in the following sections. The scheduler allocates network resources in the form of Resource Blocks (RB), defined as 12 consecutive subcarriers for a single frequency slot. RBs are allocated to users by giving the UE a specific number of sub carriers (frequency domain) for a predetermined amount of time (time domain) [3].

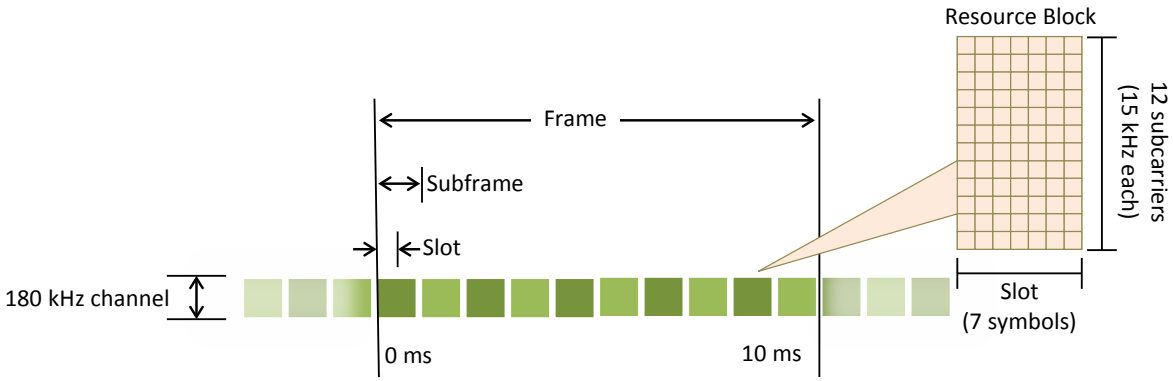



Figure 2. A simple depiction of the relationship between frequency and time highlighting a normal CP resource block and a single 180 kHz channel.

### 2.1 Time Domain

The LTE frame is 10 ms long, divided into 10 sub frames, each 1.0 ms long, which are further divided into 0.5 ms duration slots. Each slot is made up of OFDM symbols, and depending on the mode, uses either a seven symbol normal cyclic prefix (CP) or a six symbol extended CP. The CP is used as a guard period at the beginning of each OFDM symbol, to negate any potential Inter Symbol Interference (ISI) resulting from a high-rate data stream being transmitted serially.

### 2.2 Frequency Domain

In the frequency domain, RBs are groups of 12 subcarriers, each 15 kHz wide, occupying a total of 180 kHz. The data carrier and smallest element in LTE is the resource element (RE), which is one subcarrier × one symbol. This means that one RB is 12 subcarriers × 7 slots = 84 REs with a normal CP, or 72 REs with an extended CP. However, not all REs are used for data service. Some are reserved for signaling information, which is a unique feature of LTE among packet-switched networks that traditionally use a physical layer preamble. Symbol coding is performed on the REs; depending on the channel parameters, symbols can be coded with QPSK (2 bits/symbol), 16QAM (4 bits/symbol), or 64QAM(6 bits/symbol). The number of resource blocks available depends on the available bandwidth, but goes from 6 to 100 PRBs as bandwidth increases. As network conditions allow, multiple bandwidths are available: 1.4 MHz, 3 MHz, 5 MHz, 10 MHz, 15 MHz, or 20 MHz. (Public safety will use a 10 MHz bandwidth for FirstNet.) The result of this architecture is that a 10 MHz LTE channel consists of 50 subchannels × 2000 RB/sec = 100,000 RB/sec that can be scheduled to requesting UEs.

## 3. NETWORK MODEL

We begin presenting our work by defining a mathematical model of our network environment. We define terms and develop a cost function which our scheduling algorithms will optimize. Finally, we present a formal problem statement for this work.

### 3.1 Defining the Network Environment

Let $\mathcal{N}$ be the set of all nodes within a given LTE cell sector $\mathcal{N} = \{n_1, \dots, n_k\}$. A node is defined as the endpoint of a video stream from the base station to a UE. While as presented in this work a node is synonymous with a user, that does not need be the case. A user can request two video streams, in which case one UE is modeled in our network as containing two nodes (one for each video requested). Thus, for consistency with any future work, we use the terminology node. Let $\mathcal{R}$ be the set of all resource blocks representing all 50 subchannels for 1 second duration $\mathcal{R} = \{1, \dots, 50\} \times \{1, \dots, 2000\}$. $\mathcal{F}$ then represents the set of all possible mappings of network resources onto nodes, $\mathcal{F}: \mathcal{R} \to \mathcal{N}$, with $f \in \mathcal{F}$ representing a specific network resource allocation.

### 3.2 Bitrate Function

We define $b(n, f)$ as the downstream bitrate that node $n$ can receive under the network resource allocation $f$. This can be computed by using the resource allocation defined by $f$ along with the channel characteristics between node $n$ and the base station, for each LTE subchannel that is available. Based on these parameters, $b(n, f)$ is computed as follows:

**Algorithm 1** Node Bitrate

1: *Given :* $n, f$
2: $bitrate = 0$
3: **for all** subchannel $s$ in $n.subchannels$ **do**
4: **if** $s$ is available **then**
5: $bitrate \mathrel{+}= f_{RB}^{s}(n) \cdot s.CPMode \cdot s.Modulation \cdot 12$
6: **end if**
7: **end for**
8: **return** $bitrate$

with the following definitions:

- $f_{RB}^{s}(n)$: The number of resource blocks per second assigned to node $n$ on subchannel $s$.
- $s.CPMode$: The number of symbols per slot; either 7 (normal CP) or 6 (extended CP).
- $s.Modulation$: The number of bits per symbol used in encoding data; either 2 (QPSK), 4 (16QAM), or 6 (64QAM).
- 12: The number of sub-carrier frequencies within a single subchannel, as defined in the LTE standard.

### 3.3 Network Usefulness

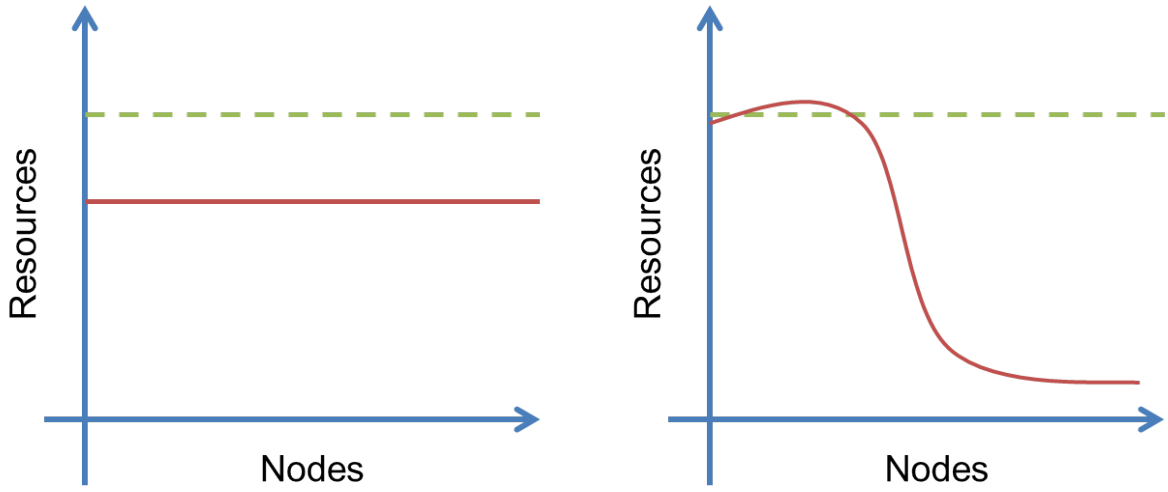
In typical optimization efforts, a cost function is defined for the system and optimization occurs by minimizing the system's cost. In our scenario, we want to maximize the usefulness that the LTE network can provide to its users, i.e., maximize the value provided by the resource blocks based on how they are allocated to nodes. To do this, we define a network usefulness function $U(f)$, as shown in (1), which sums the utility $u(n, f)$ provided by each node in the network under the current resource allocation $f$ and weights it against the node's priority $p(n)$. In this respect, the usefulness function $U(f)$ can be thought of a cost function for the network, and we attempt to optimize the network by identifying an $f \in \mathcal{F}$ that maximizes $U(f)$.

$$U(f) = \sum_{\mathcal{N}}^{n} u(n, f) \cdot p(n) \tag{1}$$

#### 3.3.1 Node Priority

Constructing a rigid algorithm to perform resource scheduling is not in itself desirable for real-life deployment scenarios. An important issue for public safety practitioners in the field is the concept of local control over their network. In an emergency event, local personnel will have the best understanding of what the network needs to be able to support and the goals that need to be accomplished. For example, local public safety personnel will understand which nodes are high priority and must receive data at any cost. No matter how efficiently we design our resource scheduling algorithm, without local input we can only optimize for general assumptions and never anticipate the insights and inputs that are vital to ensure the system is as efficient as possible for the local users in the field.

Node priority allows public safety practitioners to provide inputs into the system that can influence how network resources are allocated by the scheduler. When an emergency event occurs and public safety personnel arrive on the scene, each public safety worker has a different role. While all roles are important to handling the situation, as network resources become constrained it becomes evident that certain personnel require a higher guarantee of network resources compared to others. For example, a firefighter inside a burning building will require a higher guarantee of network resources than a policeman cordoning off the incident area. This leads to the concept of node priority, which is the ability to designate certain nodes within the system as being of higher priority than the rest, so that during times of constrained resources the scheduling algorithm knows which are the most important and can assign resources accordingly.




(a) (b)

Figure 3. Resource allocation alternatives within a resource constrained environment with nodes sorted by priority (higher priority nodes to the left). Dotted line shows minimum utility. (a) All nodes assigned same priority. (b) Nodes assigned different priorities.

To understand the influence that priority assignments have on the overall network, assume we have a network in which all nodes have the same priority (with all other factors being equal, such as bandwidth requests). If we plot the resources allocated within a resource constrained environment, sorting the nodes from highest priority to lowest, we can visualize the impact of priority. In Figure 3(a), we see that since all nodes have the same priority, that every node is allocated the same number of network resources—in essence, every node experiences an equal reduction in the number of resources they receive. However, if we were to assign priority values to nodes and re-plot the results, we get Figure 3(b); higher priority nodes receive their full resource requests at the expense of lower priority nodes.

We define a simple priority function that allows each $n \in \mathcal{N}$ to be assigned its own priority within the model, allowing for fine-grained definition of user priority during network optimization. (For simplification, we limit to integer values.) Local first responders will assign each node a priority in this range, with the same priority allowed to be assigned to multiple nodes. In general, a node with priority 1 will receive ten times the resources received by an identical node with priority 10, although there are many other internal algorithmic parameters and conditions that play into resource assignment. To accomplish this, we use the priority function defined in (2), representing an inverse relationship between the magnitude of the priority value and its effect on the user (i.e., smaller values are of higher priority).

$$p(n) = \frac{1}{priority} \tag{2}$$

### 3.3.2 Utility Functions

The utility function is *the probability that a user can successfully identify the object in the scene*. This was defined and studied in [14] through subjective public safety practitioner experiments. This work, along with other previous research [6] [12] [13], showed that the content and scene of a video stream has a major effect on how a video is perceived at a given bitrate. Thus, in order to successfully optimize a network's resources, we must take into account the properties of the video streams themselves.

In [14], the researchers categorized video clips based on three selected properties:

- **Lighting**: The lighting conditions of the scene: Bright, Dim, or Variable.
- **Motion**: The motion of the object within the scene: High or Low.
- **Target Size**: The relative size of the target of interest in the video: Large or Small

From these properties, they generated generalized use classes and presented public safety practitioners video clips during subjective testing, asking them to identify a target object in the scene. The result was a set of recommendations and curves showing the relationship between bitrate and object identification under fixed conditions, such as scene lighting, motion, and target size.

From this work, we can expand our utility function, $u(n, f)$, to incorporate $v(n)$, which describes the characteristics of the video stream the node $n$ is requesting. We assume that each video stream's properties are known *a priori*, through either real-time analysis at the eNodeB or via some other back-end system. The generalized form of the utility function $u(n, f, v)$, is presented in (3).

$$u(n, f, v) = u(b(n, f), v(n)) \tag{3}$$

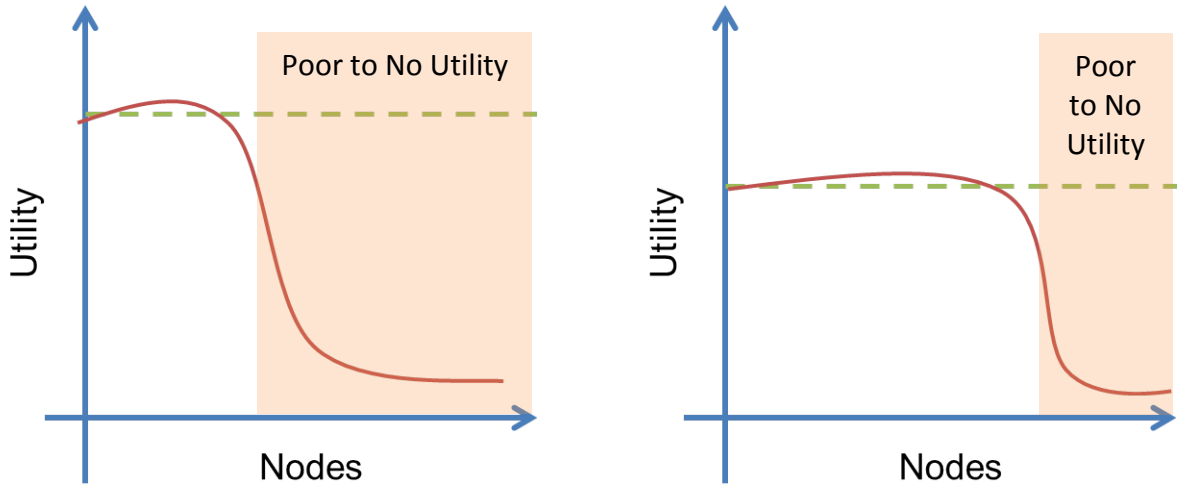
In the rest of this subsection, we introduce the concept of minimum utility and how it relates to our problem of optimal resource scheduling. We then introduce four utility functions which we use in our experiments and which are represented by the curves shown in Figure 5.

### 3.3.3 Minimum Utility

Let us revisit our example of the firefighter inside a burning building, except this time assume there are multiple firefighters. Assigning them all a higher priority than all other public safety personnel will ensure that they are more likely to receive network resources, but assume that the network is so overloaded, including a large number of firefighters, that it still has resource constraints even among high priority users. We introduce the concept of minimum utility to specify a minimum acceptable utility for nodes.

We previously defined utility as the probability of a user successfully identifying an object or event in a video stream for a given bitrate. In essence, utility is a measure of how useable a video stream is to the public safety practitioner. What minimum utility does is define a lowest acceptable limit on utility that the network should provide to users. If this minimum cannot be met with available resources, the algorithm should attempt to assign those resources to another

node such that it can achieve the minimum utility. For example, instead of having two users both with poor utility, both receiving unusable video streams, the network will favor having one user with high utility and one with low/none (of course, details such as node priority and others will determine which user gets high utility and which gets low).




(a) (b)

Figure 4. Resource allocation alternatives with utility calculation; nodes sorted by priority (higher priority nodes to the left). Dotted line shows minimum utility. (a) High minimum utility. (b) Lowering the minimum utility allows more nodes to achieve it.

To understand how minimum utility affects resource allocation, assume, as we did previously, that we plotted Nodes vs. Resources (nodes sorted with higher priority to the left). Figures 4(a) and 4(b) show the effects of altering the minimum utility for network.

As can be observed, a higher minimum utility requires more network resources. In turn, this reduces the number of users that are able to achieve this level of utility in a resource constrained environment. Lowering the minimum utility reduces the number of resources required to achieve this level of utility, thus allowing more users to achieve it. In essence, the minimum utility input affects the peak level of resource allocation. Lowering the peak level allows the algorithm to allocate resources such that more users can achieve it.

### 3.3.4 Experimental Utility Function

The experimental utility function is pulled directly from the data gathered in [14]. It is based on observed and measurable data, gathered through subjective testing of public safety practitioners, to create a set of curves mapping video bitrate to utility for different combinations of video stream properties (Figure 5(a)). This set of utility functions, referred to as $u_{exp}(n, f, v)$, is piecewise-linear, using linear interpolation to derive values between points from the original data set. Technically, this utility function doesn't satisfy our desire for a minimum utility value, since

these curves are fixed and cannot be customized to meet local public safety needs. However, we include them for completeness since all following utility functions will inherit from this dataset.

### 3.3.5 Step Utility Function

Let us define two values, the allocated bitrate $b_{alloc}$ and the minimum bitrate $b_{min}$. The allocated bitrate is self-explanatory in that it is the bitrate that the node is allocated by the scheduler based on the current network resource allocation. The minimum bitrate, however, is derived from the experimental utility function. Given a minimum acceptable utility value for a video stream, what is the corresponding bitrate that is required to support this? In other words, given $u_{min}$, solve for $b_{min}$ in:

$$u_{min} = u_{exp}(n, f, v) = u(b_{min}, v(n))$$

This value is computed using the video classification characteristics and the inverse of the corresponding utility curve from [14].

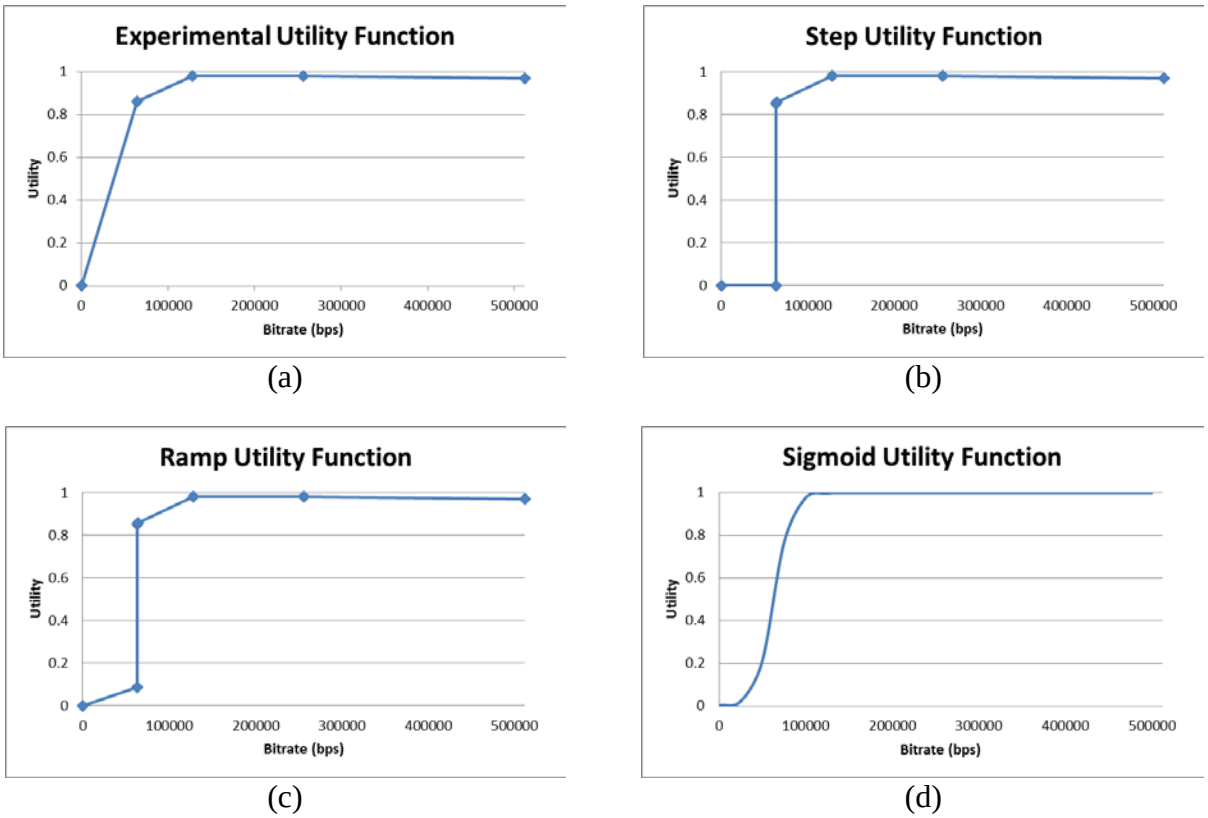



Figure 5. Utility functions. (a) Experimental utility function. (b) Step utility function. (c) Ramp utility function. (d) Sigmoid utility function.

The step utility function incorporates the minimum utility parameter as a union of the experimental utility function with a step function aligned to the critical bitrate. Any allocated bitrate above the critical bitrate would return the value as computed on the experimental utility

function. If the allocated bitrate is less than the critical bitrate, then the utility function will always return zero. Using the previously defined notation, the step utility function is shown in Figure 5(b) and defined in (4).

$$u_{step}(n,f,v) = \begin{cases} 0, b_{alloc} < b_{critical} \\ u_{exp}(n,f,v), b_{alloc} \geq b_{critical} \end{cases} \quad (4)$$

### 3.3.6 Ramp Utility Function

The third utility function that we present is a modification of the step utility function. In this case, we remove the presence of the horizontal function of value 0 for the range $0 \leq b_{alloc} \leq b_{critical}$. We postulate that a requested bitrate within this range has no "incentive" to either move towards greater levels of utility or relinquish its network allocation. On the larger solution space, this would correspond to an area of zero gradient. To attempt to address this, we selected an arbitrary value of 10% and used a ramp function bounded by $0 \leq b_{alloc} \leq b_{critical}$, while still keeping the effects of the step function, as shown in Figure 5(c). The ramp segment is modeled by creating a line through the two points $(0,0)$ and $\left(b_{critical}, 0.1 * u_{exp}(b_{critical})\right)$, resulting in (5).

$$u_{ramp}(n,f,v) = \begin{cases} 0.1\left(\frac{b_{alloc}}{b_{min}}\right) u_{exp}(n,f,v), b_{alloc} < b_{critical} \\ u_{exp}(n,f,v), b_{alloc} \geq b_{critical} \end{cases} \quad (5)$$

### 3.3.7 Sigmoid Utility Function

Our final proposed utility function looks to use the basic shape of the ramp, but to model it with a single, continuous function. Additionally, we look to use a function which does not contain a constant tangent over any interval. To accomplish this, we utilized a function based on a sigmoid curve, which, like the more general logistics curve, has a well-recognized "S" shape (Figure 5(d)). We postulate that a utility function with a constant slope could cause, under certain circumstances, the scheduling algorithm to again become "stuck," similar to the logic motivating the ramp utility function.

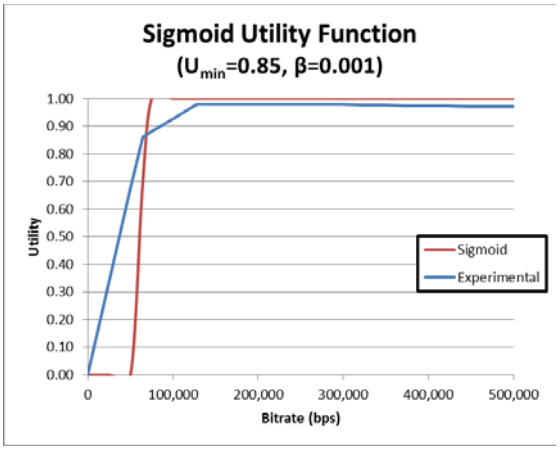
In its general form, a sigmoid function is defined by (6) and centered on the vertical axis, with $\beta$ controlling its rate of increase (saturation).

$$S(x) = \frac{1}{1 + e^{-\beta x}} \quad (6)$$

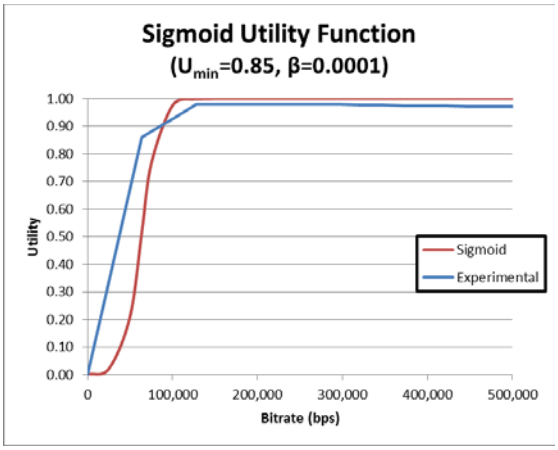
An additional benefit of using a sigmoid curve is that it naturally results in non-negative numbers for all values (in its general form) and has a range of $(0,1)$, which maps directly to the possible ranges of utility. For our model, we perform a mathematical translation to center the sigmoid curve on the critical bitrate. This results in a sigmoid utility function modeled in (7).

$$u_{sigmoid}(n, f, v) = \frac{1}{1 + e^{-\beta(b_{alloc} - b_{critical})}} \tag{7}$$

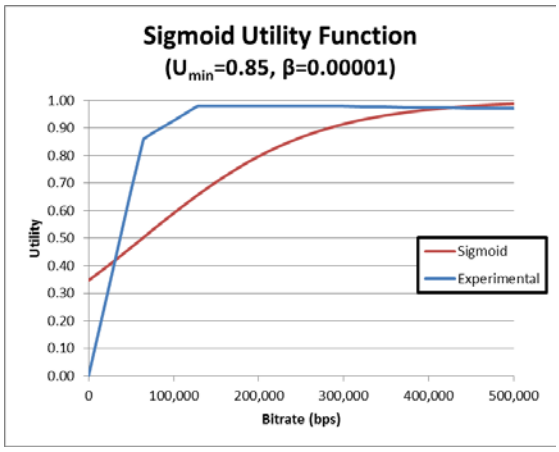
To select a value for $\beta$, we looked for a value that produces an approximately similar shape when plotting $u_{sigmoid}$ vs. $u_{exp}$. In Figure 6, we set the minimum utility to 0.85 and plotted $u_{sigmoid}$ vs. $u_{exp}$ for $\beta = [0.001, 0.0001, 0.00001]$. We see that for $\beta = 0.001$, $u_{sigmoid}$ behaves nearly like a step function, with an additional overshoot and undershoot around $b_{crital}$. On the other hand, for $\beta = 0.00001$, $u_{sigmoid}$ increases too slowly. $\beta = 0.0001$ provides a reasonable approximation of $u_{exp}$ using a continuous function with non-constant slope. Thus, we set $\beta = 0.0001$ for this work.




(a)




(b)




(c)

Figure 6. The effects of β on the sigmoid utility function are shown by the red line, with the blue line showing the experimental function as a reference. (a) β = 0.001, (b) β = 0.0001, (c) β = 0.00001

### 3.4 Maximizing Network Usefulness

With our network model defined, we can compute the total usefulness of the network resource allocation $f$:

$$U(f) = \sum_{n \in \mathcal{N}} u(n, f, v) \cdot p(n) = \sum_{n \in \mathcal{N}} u(b(n, f), v(n)) \cdot p(n) \quad (8)$$

Thus we can formally present our problem statement:

Given a set of all possible network resource allocations $\mathcal{F}: \mathcal{R} \rightarrow \mathcal{N}$, find $f \in F$ such that $U(f) \geq U(f') \forall f' \in \mathcal{F}, f' \neq f$.

# 4. ALGORITHMS

In the previous section, we developed a mathematical model of our network environment by focusing on the concept of usefulness as a function of a given network resource allocation $f \in \mathcal{F}$. In this section, we present three optimization algorithms which were tested to optimize the network by maximizing its usefulness.

## 4.1 Iterated Conditional Modes

Iterated Conditional Modes (ICM) is a deterministic algorithm that guarantees convergence to a local maximum. Proposed by Besag [4] for the application of noise reduction in image processing, ICM sequentially maximizes local conditional probabilities through an iterative approach in which each pixel is maximized conditioned on its neighbors. It uses local conditional probabilities on the assumptions that neighboring pixels tend to have similar values within an image and that noise is injected into pixels independently of their neighbors. Let $y$ be the noisy image of the uncorrupted image $x$, with $x_{i,j}^{k}$ representing the estimation of pixel $(i, j)$ at iteration $k$. ICM computes $x_{i,j}^{k+1} \big| x_{i\pm1,j\pm1}^{k}$ until $x_{i,j}^{k+1} = x_{i,j}^{k}$.

We map ICM onto our problem-space in a straightforward manner. The algorithm iterates over a network of nodes (analogous to a network of image pixels), with the special case that every node in our network is a neighbor of all other nodes. The neighborhood system is an assumption to make image processing feasible. Here, we relax this assumption as there are relatively few users. From this, we derive Algorithm 2, which continually reallocates network resources, selecting a new $f \in \mathcal{F}$, until it finds the $f$ that maximizes $U(f)$.

## 4.2 Metropolis-Hastings Algorithms

The second optimization algorithm we look at is the Metropolis-Hastings Algorithm, first proposed by Metropolis [10] and later extended into the general case by Hastings [8]. This algorithm is a Markov chain Monte Carlo (MCMC) method which can be useful in obtaining a solution where the solution space consists of a large multi-dimensional distribution. By selecting a sequence of samples, the algorithm can generate a distribution that closely approximates the desired distribution, with increasing numbers of samples improving the desired accuracy. Non-deterministic by nature, the algorithm accomplishes this through a random walk along the multi-dimensional distribution, selecting a new point $x_{i+1}$ based on a probability density $Q(x_{i+1}|x_i)$, in which $Q(x_{i+1}|x_i)$ must be symmetric, i.e., $Q(x|y) = Q(y|x)$, and is usually represented as a Gaussian. The standard deviation $\sigma$ of the Gaussian used in the sampler is a variable in our model. We will examine the effects it produces when presenting our results.

**Algorithm 2** Iterated Conditional Modes

*Initialize:* $f \leftarrow InitialResourceAllocation()$
$U_{max} = U(f)$
**do**
  $reallocationOccured = FALSE$
  **for all** $n_i \in \mathcal{N}$ **do**
    **for all** subchannel $s$ in $n_i.subchannels$ **do**
      **for all** $n_j \in \mathcal{N}, n_i \neq n_j$ **do**
        $f^* \leftarrow n_i[s].Deallocate(), n_j[s].Allocate()$
        $U_{temp} = U(f^*)$
        **if** $U_{temp} \leq U_{max}$ **then**
          $f \leftarrow n_i[s].Allocate(), n_j[s].Deallocate()$
          **continue**
        **else**
          $U_{max} = U_{temp}, reallocationOccured = TRUE$
        **end if**
      **end for**
    **end for**
  **end for**
**while (**$reallocationOccured$**)**
**return** $f$

Whereas ICM is a deterministic algorithm and converged to a local maximum (which is not guaranteed to be the global maximum), the Metropolis-Hastings Algorithm has the ability to reject a new point, even if that point appears to improve the desired solution, through the use of an acceptance ratio $\alpha$, as defined in (9). Thus, through properly defined variables, the algorithm can remove itself from a local maximum in search of the global maximum.

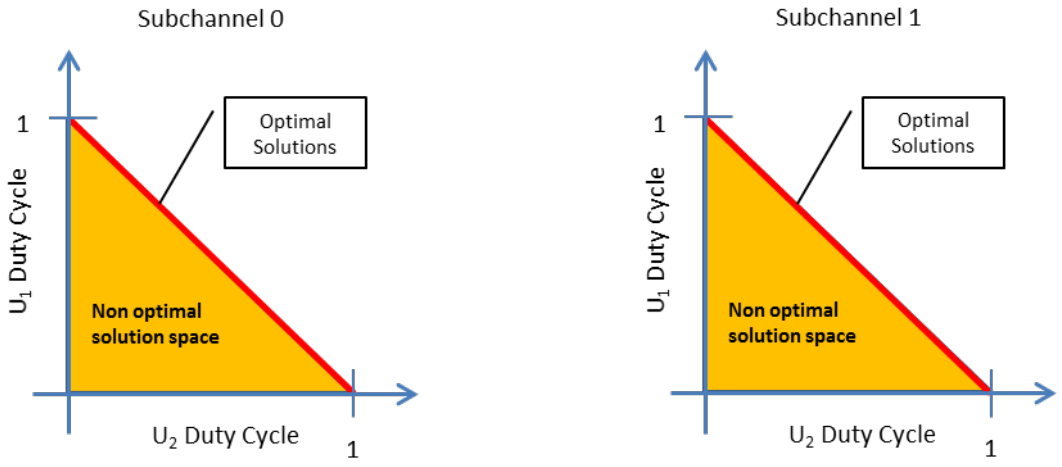
$$\alpha = \frac{U(f_{i+1})}{U(f_i)} \tag{9}$$

Additionally, whereas ICM operates over discrete units (resource blocks), the Metropolis-Hastings Algorithm operates on a continuing spectrum of values. This allows us to work with the more precise duty cycle of the node, providing the added benefit of presenting a more accurate description of the physical world, since in our original definition of $\mathcal{R}$ we had to select an arbitrary value for our time range (1 second). Using the duty cycle removes this arbitrary value from our model, creating a continuous spectrum in the time domain. Algorithm 3 presents the Metropolis-Hastings Algorithm as applied to the problem-space.

**Algorithm 3** Metropolis Hastings Algorithm

*Initialize:* $f_0 \leftarrow InitialResourceAllocation()$
$t = 0, U_t = U(f_t)$
**while** $t < t_{max}$ **do**
  $f_t^* \leftarrow Sample(f_t)$
  $U_t^* = U(f_t^*)$
  **if** $U_t^* > U_t$ **then**
    $f_{t+1} \leftarrow f_t^*$
  **else**
    $\alpha_t = \left(\frac{U_t^*}{U_t}\right)$
    $f_{t+1} \leftarrow f_t^*$ with probability of $(1 - \alpha_t)$
  **end if**
  $t = t + 1$
**end while**
**return** $f_{t_{max}}$
**Procedure** Sample(f)
**for all** $n_i \in f$ **do**
  **for all** subchannel $s$ in $n_i.subchannels$ **do**
    $f^* \leftarrow n_i[s].Allocate(rand(0, \sigma))$
  **end for**
**end for**
**return** $f^*$

### 4.2.1 Solution Space




(a) (b)

Figure 7. Simplified view of non-convex solution space. (a) Subchannel 0, and (b) Subchannel 1.

Before continuing, it is worth considering the solution space the algorithm will operate on. While we know that the solution space is highly multi-dimensional, due to the 50 independent subchannels that make up the wireless communication medium, upon simple examination it can

further be shown that the solution space is non-convex. Take a simple example of an environment with two users of equal priority and only two subchannels. Since the priority is equal, any globally optimum solution can be reproduced by swapping the resources allocated to each user. Hence, there is more than one globally optimum solution, and the space cannot be convex. Further, since the total duty cycle allocated between all users in a given subchannel cannot exceed 1, the solution space for a specific subchannel is bounded by the line $d_1 + d_2 = 1$, with $d_i$ being the duty cycle of node $n_i$. (It is straightforward that duty cycle cannot be negative, and thus the axes serve as the other bounds for the solution space.)

In this example, shown in Figure 7, the shaded region is the complete solution space. Hence, we have an $|\mathcal{N}|$-dimensional solution space for each subchannel. Additionally, since the allocation of an unallocated resource to a node can never cause the overall utility to decrease, i.e., a resource can never have a negative utility, the line marking $d_1 + d_2 = 1$ can be described as the optimal solution space, in that we expect to see the solution that maximizes the usefulness to reside on that line for each subchannel. This gives an optimum solution space of dimension $|\mathcal{N} - 1|$ for each subchannel.

### 4.2.2 Invalid Solutions

The Metropolis-Hastings Algorithm randomly selects a new point based on a normal distribution. It is possible (and highly probable), that an invalid point will be selected when the current point is near the bounds of the solution space. An invalid point is a point that falls outside of the solution space, such as a subchannel that has total duty cycle greater than one, or a duty cycle for a subchannel for a particular node that is negative. While one could have the Metropolis-Hastings Algorithm select another point, this could cause performance issues if the current point sits in a location where a larger majority of the points near it are invalid. To alleviate this, we project the invalid value back onto the normal of the solution surface to arrive at a valid point.

Additionally, we combine the knowledge of the solution space with the ability to project points to gain an additional optimization to our algorithm. Since we know that all maximized solutions must fall on the optimum solution space, we can force the simulator to project all selected points, valid or invalid, to the optimum solution space. We expect that such a behavior may result in performance increases, as the simulator avoids “walking around” within the non-optimal solution area. However, we acknowledge that if local minimums or maximums exist within the surface, forcing the projection may inhibit the algorithm from the possibility of escaping them. We look for such behavior when analyzing the results.

## 4.3 Simulated Annealing

Simulated Annealing [5] [7] [9] is the idea of slowly reducing the probability of accepting a worse solution given your current state. Inspired by the thermodynamic concept of annealing, simulated annealing can be applied to allow one to identify an approximation of a global maximum without requiring an exhaustive search. The Metropolis-Hastings Algorithm lends itself nicely to this approach, as it allows for a non-zero probability of accepting a worse result at any given sample.

Remember that in the Metropolis-Hastings Algorithm, the probability density function $Q(x|y)$ must be symmetric, and is usually a Gaussian distribution centered at $y$. We look to gain efficiencies by applying simulated annealing to the sampler within our network optimization problem. To model this, we map the simulated annealing concept of temperature $T$ to the standard deviation $\sigma$ of the sampler's distribution, with $\sigma_0$ defined as *the initial standard deviation of the sampler*.

During the execution of the Metropolis-Hastings Algorithm, $\sigma \to 0$ as $t \to t_{max}$. Since standard deviation must be a positive, non-zero number, we define $\sigma$ such that it reduces asymptotically to zero through an exponential decay model. Thus we define the standard deviation $\sigma_t$ at a given time iteration $t$ as (10).

$$\sigma_t = \sigma_0 e^{-c\left(\frac{t}{t_{max}}\right)} \tag{10}$$

with $\sigma_0$ being the initial standard deviation and $t_{max}$ being the total time allotted for algorithm execution, in iterations. The constant $c$ defines the reduction in $\sigma_0$ from $t_0$ to $t_{max}$ as (11).

$$\frac{\sigma_0}{\sigma_{t_{max}}} = \frac{1}{e^{-c}} = e^c \tag{11}$$

We adjust the acceptance probability computation in the original Metropolis-Hastings Algorithm so that the above holds true. This is done by updating the computation of $\alpha$ as in (12).

$$\alpha = \left(\frac{u^*}{u}\right)^{\sigma} \tag{12}$$

Applying this to the Metropolis-Hastings Algorithm now yields Algorithm 4. The initial standard deviation $\sigma_0$ and the scaling constant $c$ are both parameters to the algorithm and the effects they have on the final resource allocation are presented in our results.

**Algorithm 4** Metropolis Hastings Algorithm with Simulated Annealing

*Initialize:* $f_0 \leftarrow InitialResourceAllocation()$
$t = 0, U_t = U(f_t)$
**while** $t < t_{max}$ **do**
  $\sigma_t = \sigma_o e^{-c\left(\frac{t}{t_{max}}\right)}$
  $f_t^* \leftarrow Sample(f_t, \sigma_t)$
  $U_t^* = U(f_t^*)$
  **if** $U_t^* > U_t$ **then**
    $f_{t+1} \leftarrow f_t^*$
  **else**
    $\alpha_t = \left(\frac{u_t^*}{u_t}\right)^{\sigma_t}$
    $f_{t+1} \leftarrow f_t^*$ with $p(1 - \alpha_t)$
  **end if**
  $t = t + 1$
**end while**
**return** $f_{t_{max}}$
**Procedure** Sample(f, $\sigma$)
**for all** $n_i \in f$ **do**
  **for all** subchannel $s$ in $n_i.subchannels$ **do**
    $f^* \leftarrow n_i[s].Allocate(rand(0, \sigma))$
  **end for**
**end for**
**return** $f^*$

# 5. SIMULATIONS

In order to compare our three optimization algorithms, we developed our own discrete time simulator to analyze their performance. For a baseline measurement, and since we do not have knowledge of a current eNodeB's scheduling algorithm, we implemented a fourth scheduler based on a greedy algorithm. This greedy scheduler is a simple algorithm in which the highest priority nodes get all the resources they need to maximize their utility at the expense of the lower priority nodes. No attempt is made in this greedy scheduler to ensure all resource blocks are assigned.

## 5.1 Environments

To maintain consistency between algorithm results in order to perform proper comparisons, we always define our network's initial resource allocation in resource blocks. When optimizing with Metro (Metropolis-Hastings Algorithm) or Anneal (Metropolis-Hasting Algorithm with Simulated Annealing), we insert an initial step to convert the number of resource blocks assigned to a node into its corresponding duty cycle.

We analyze our scheduling algorithms in two different network environments. We begin by comparing their performance in an ideal network environment in which all nodes have equal access to every subchannel. This provides a look at how the schedulers behave in ideal conditions.

In the second part of our simulations, we analyze how the scheduling algorithms behave in increasingly degraded network conditions. We run multiple simulations with the percentage of the subchannels available to the UEs varying from 100% to 50%. This is performed by generating network environments in which every UE independently has a fixed percentage of their subchannels defined as unavailable, with the available subchannels randomized among the UEs. We generated 25 such network condition files for each data point, from 100% to 50%, with increments of 10%. For deterministic scheduling algorithms, we ran each of the 25 network conditions once and averaged their results. For non-deterministic algorithms, we used the same method as in the ideal network conditions and ran each network definition file 25 times, averaging the results, for a total of 625 measurements (25 network definitions each simulated 25 times).

## 5.2 Assumptions

We assumed that an LTE subchannel carries 200 RBs per second as opposed to 2000 RBs per second. This reduces the number of nodes required to create a congested network environment, while not affecting the relative performance of any of the scheduling algorithms with respect to each other. This allows us to shorten the duration of each simulation, since in implementing our algorithms we did not focus on computational optimization, such as through implementing paralleling, and thus we shy away from making any statements of this type in our findings. Likewise, we assumed that all nodes modulated their signals using QPSK and with Extended CP Mode—again, to minimize simulation durations.

For the Simulated Annealing sampler, we selected a value of $c = 10$ for the exponential decay model. Similar to how we selected $\beta$ in the sigmoid utility function, we plotted the exponential decay of the sampler with three different values of $c = \{1, 10, 100\}$ in Figure 8. When $c = 100$, the model decays extremely fast over its lifetime. Likewise, when $c = 1$ the model decays slowly and never reaches the extremely small values of $\sigma_t$ that allow the sampler to identify an optimal solution. Thus, we selected $c = 10$, for our simulated annealing sampler. We suspect, based on the Simulated Annealing algorithm, that a relationship exists between $c$ and the run time $t$ (in iterations), but we leave this for future study.

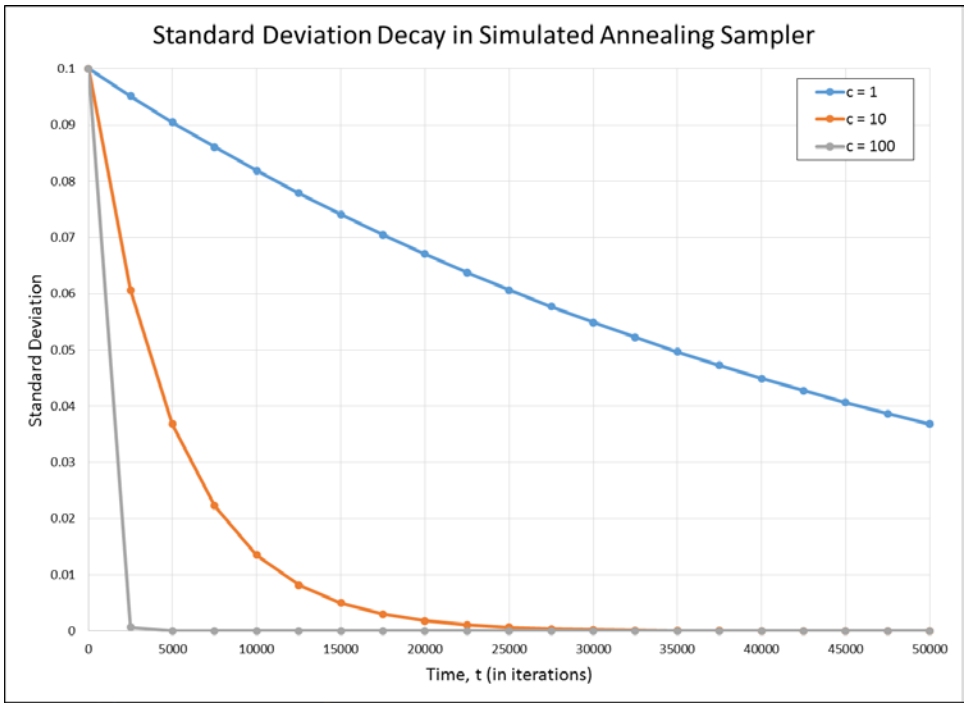



Figure 8. Selection of standard deviation decay parameter.

Lastly, we assumed all video was delivered in CIF format, and that the video content is classified as a Light environment with Moving motion, based on the classifications in [14]. We want to focus on the algorithms themselves and choose not to look at complex relationships between algorithms, channel availability, and video quality and classification. Such investigations are left to future work.

# 6. RESULTS

## 6.1 Baseline Analysis

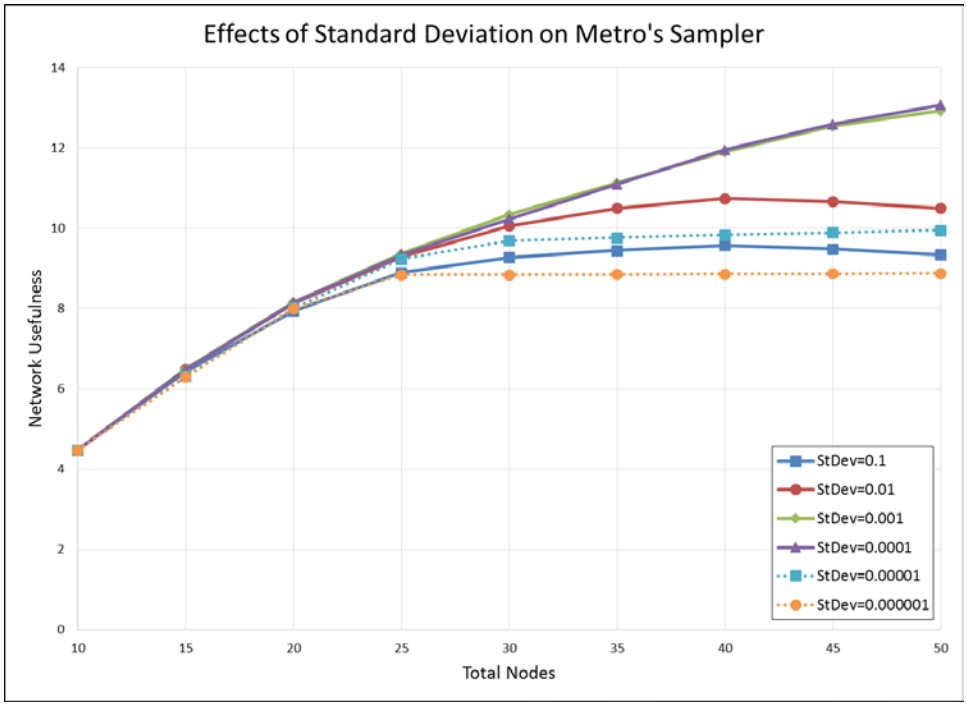
We begin by performing a baseline analysis on our proposed scheduling algorithms. We use an ideal network environment in which every node has all of its subchannels available for transmitting and receiving data. We use the experimental utility function for all our baseline analysis.

The Metro algorithm requires the selection of a standard deviation for its sampler. We tested values from 0.1 to 0.000001 in logarithmic intervals. The results are shown in Figure 9(a). We see that, initially, as we decrease the standard deviation, we get improved overall performance with respect to network usefulness. As the number of nodes increases in the network, a sampler with a large standard deviation is unable to identify an optimum solution, as the large standard deviation reduces the likelihood that the sampler is able to select a solution within the increasingly smaller target area where the optimum solution resides. Decreasing the standard deviation increases the resolution of the sampler.

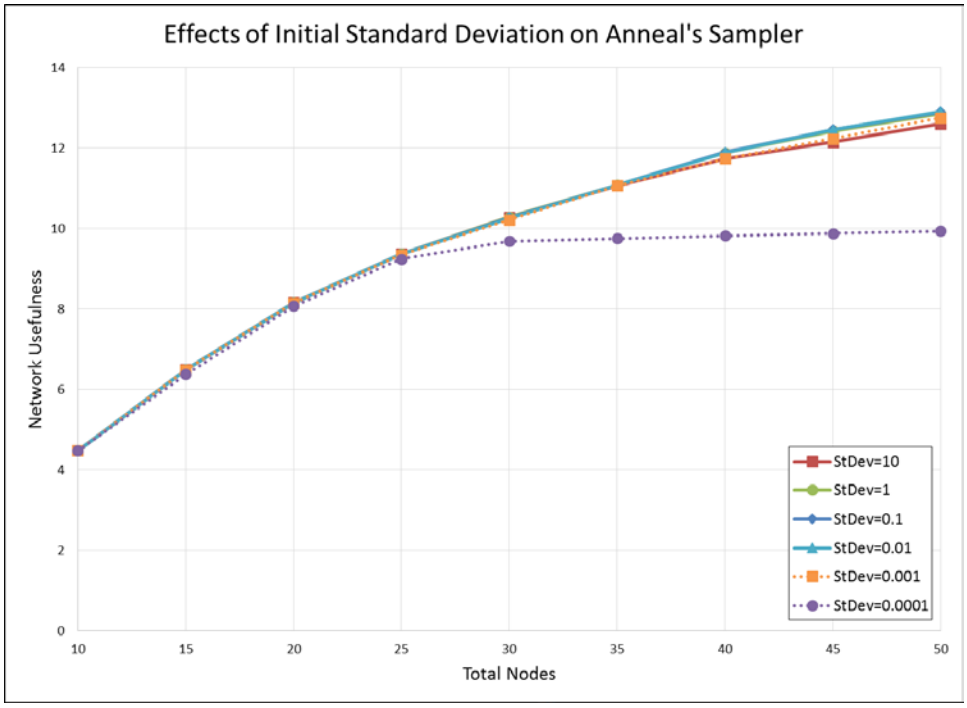
However, if the sampler's selected standard deviation is too small, it suffers from the opposite problem. In this case, the small standard deviation causes the sampler to become stuck in a local area of the solution space since it cannot move across the solution space quickly. This could be overcome by increasing the number of iterations until the algorithm terminates—but at a (possibly large) performance penalty. We selected a standard deviation value of $\sigma = 0.0001$ for use in the Metro scheduling algorithm for the remainder of our results.

The Anneal algorithm also requires the selection of a standard deviation for its sampler. We tested and plotted values from 10 to 0.0001 in logarithmic intervals. The results are shown in Figure 9(b). The Anneal algorithm is more robust against the value of the standard deviation used due to its simulated annealing property. This allows the standard deviation to decrease over time. However, careful inspection of the data shows that it, too, exhibits behavior similar to that of Metro—although in the inverse.

Unlike Metro, in which a smaller standard deviation increases the usefulness, the inverse is true with Anneal. While this may seem counter-intuitive at first, an in-depth look at the trace data produced by the algorithm explains why this is. The Anneal algorithm is based on the Metropolis-Hastings Algorithm, but uses the concept of annealing in its sampler. This means that given a beginning standard deviation of $\sigma$, the sampler will reduce $\sigma \to 0$ as $t \to \infty$ during the execution of the algorithm by means of an exponential decay model, controlled by the scaling constant ($c = 10$ in our simulations). What we are observing is that while decreasing values of standard deviation allows for better results, as shown in the analysis of Metro, not only are there diminishing returns, but once the standard deviation becomes small enough, it negatively impacts the algorithm's performance because it essentially gets stuck in a local area, with the sampler too small to ever move out of it. Thus starting with a larger standard deviation, while decreasing its value during execution, allows Anneal to avoid this issue.




(a)




(b)

Figure 9. Effects of the sampler's standard deviation on network usefulness for (a) Metro, and (b) Anneal.

Similarly, if the initial standard deviation is too large, there might not be enough time for the algorithm to converge before it terminates. This is why we saw less than optimal results when we tested with large standard deviation values, such as 10. Based on these results, we selected a standard deviation value of $\sigma = 0.1$ for use in the Anneal scheduling algorithm for the remainder of our results.

With the standard deviation selected for the Anneal and Metro, we can now perform a baseline comparison of all the scheduling algorithms, shown in Figure 10. We included a Greedy algorithm for comparison. All algorithms, with respect to total network usefulness, have relatively similar performance. Since all nodes have exactly the same network characteristics, the only thing distinguishing them from each other is their priority. Thus algorithms all default to a priority-based allocation similar to what Greedy does by default.

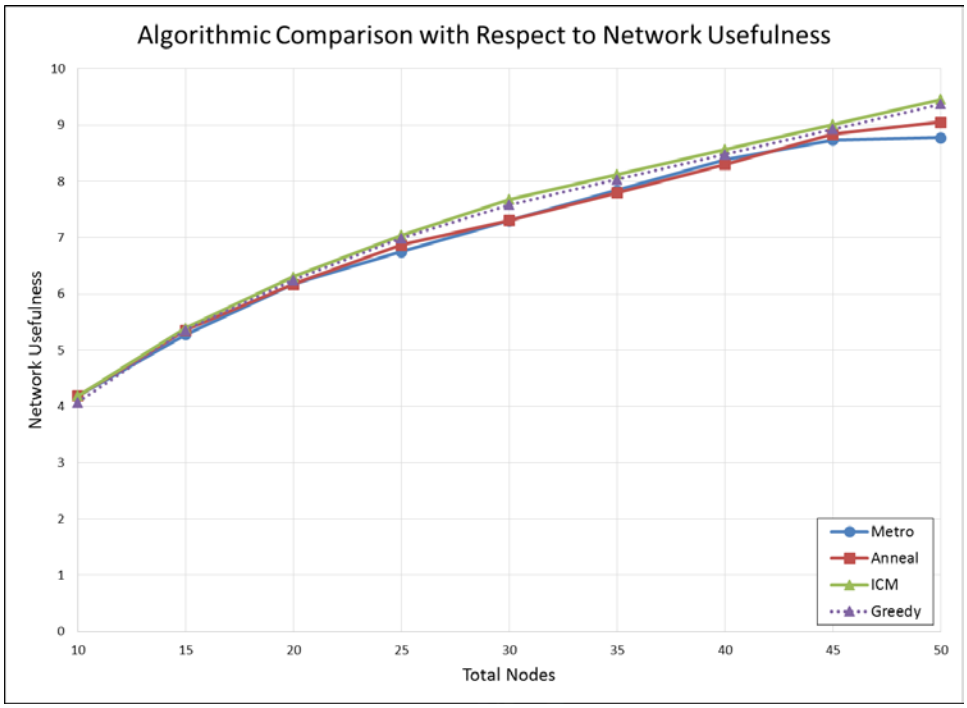



Figure 10. Comparison of scheduling algorithms in an ideal network environment.

### 6.1.1 Computational Performance

We stated that we shy away from giving time-based results to the computation of each algorithm, as we did not optimize our algorithms for computational performance. However, we can analyze the algorithms themselves to glean information. The ICM scheduler iterates through all nodes, restarting its execution for each re-allocation of a resource block. Thus, the algorithm's performance will scale linearly with the number of nodes, i.e., $O(n)$.

Both Metro and Anneal, however, execute for a fixed duration before terminating. Although increasing the number of nodes increases the computation of the sampler (through increasing the dimensions of the solution space to select from), point selection is independent between solution

space dimensions (other than the final step of verifying its validity and mapping back). This means that both Metro and Anneal have constant complexity with respect to $n$, i.e., $O(1)$.

### 6.2 Utility Functions

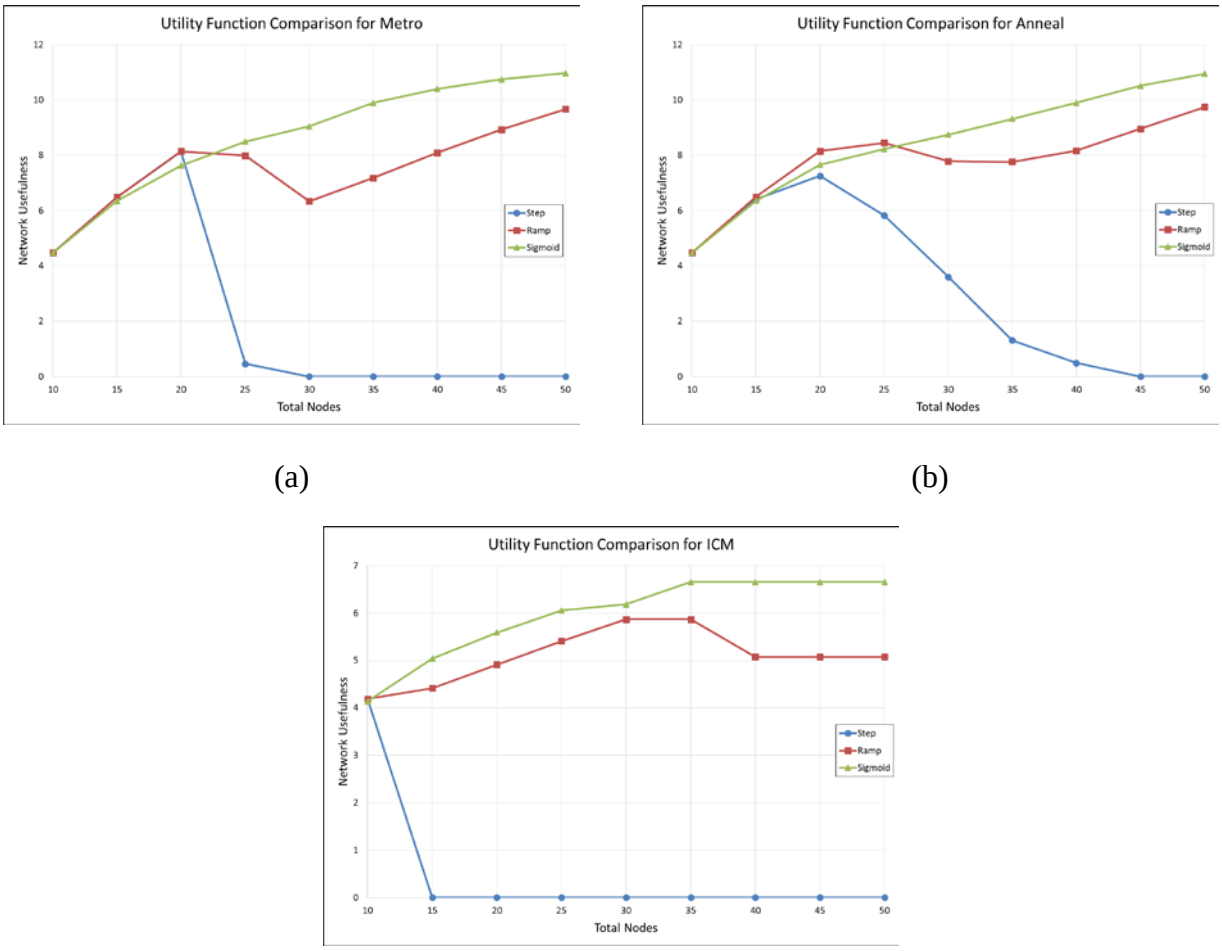



Figure 11. Network usefulness performance of utility functions for (a) Metro, (b) Anneal, and (c) ICM.

Each of our scheduling algorithms relies on a utility function which maps the allocated bitrate to the utility of the video the node is receiving. We presented three utility functions (step, ramp, and sigmoid) which were based upon the results of subjective tests of public safety practitioners [14]. We plot how each utility function maximizes network usefulness in our uniform network environment for Metro, Anneal, and ICM—Figures 11(a), 11(b), and 11(c) respectively.

The sigmoid utility function shows the best overall performance for all scheduling algorithms. For Metro and Anneal, both the ramp and step utility functions appear to perform identically to the sigmoid in environments with lower resource constraints (fewer nodes requesting resources) for Metro and Anneal. However, as resource competition increases, the performance of the step

utility function falls dramatically, and the ramp function incurs a less dramatic decrease. Additionally, the performance of Anneal is more robust to the step and utility functions.

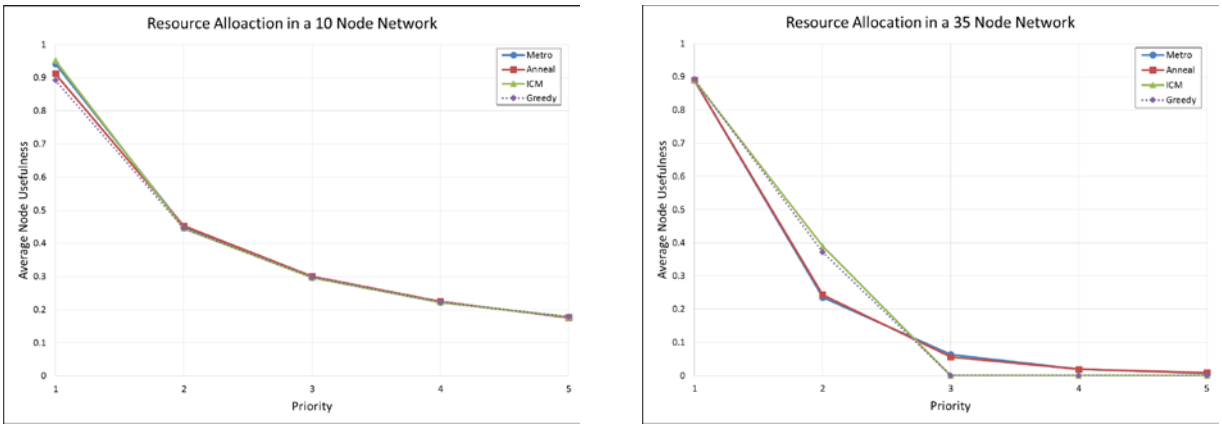
Examining the trace files, both of these behaviors can be explained. Remember that the initial environment for this simulation was uniform resource allocation to all nodes in the network. If the environment is large enough, then the initial starting bitrate of each node will be less than the critical bitrate of the utility function, which in these simulations occurred when $|\mathcal{N}| = 25$.

When starting below where the critical bitrate occurs, the outcome is largely dependent on the algorithm used. In Metro, the sampler selects a new point on the overall solution space using a fixed normal distribution, which corresponds to a new resource allocation for the network (and $f' \in \mathcal{F}, f' \neq f$). Due to the small standard deviation of the distribution (which we previously showed increased the overall performance of Metro), the likelihood of the newly selected distribution containing a node with a resource allocation greater than the critical bitrate is small (and decreases as $|\mathcal{N}|$ increases). With the step utility function being horizontal below the critical bitrate, and thus giving all nodes a fixed utility regardless of any reallocation of network resources, we are left with $U(f') = U(f) = 0$, thus accounting for the steep drop-off in utility.

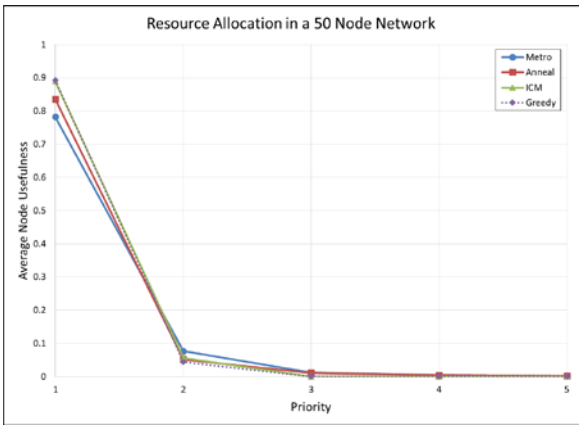
The Anneal algorithm is able to mitigate this effect somewhat because the standard deviation of its sampler begins at a much larger value, thus increasing the probability of it allocating a node enough network resources to get it over the critical bitrate. This is why we see the gradual decrease in overall network usefulness as the number of nodes increases, since the difference between the starting bitrate and the critical bitrate grows larger and thus outside the range of probabilistically being selected by the sampler (in the Metro case, due to the small standard deviation, for all practical circumstances this occurs immediately). However, the Anneal sampler's standard deviation decreases with time, thus gradually falling into the same issue that Metro encounters. The ramp utility function's behavior suffers from similar effects as the step function, although not as pronounced due to the non-horizontal natural of the function below the critical bitrate.

### 6.3 Resource Distribution by Priority

An important feature of optimizing the network resources is the algorithm's ability to, in situations of high resource contention, maximize the usefulness of each individual network resource. Remember that in our ideal network environment, the only feature that differentiates nodes is their priority. Thus, as contention increases, lower priority nodes have their resources allocated to higher priority ones. We examine how each algorithm performs in three different network conditions: low contention ($N = 10$), medium contention ($N = 35$), and high contention ($N = 50$). Figures 12(a), 12(b), and 12(c) show the allocation by node priority of usefulness and utility respectively. Looking closely, ICM does the best job at cutting off lower priority nodes from being assigned resources as resource competition increase. In fact, ICM successfully drives all lower priority nodes to zero. Metro and Anneal do a good job at removing resources from lower priority nodes as resource contention increases.




(a) (b)




(c)

Figure 12. Resource allocation by priority for a network with (a) low resource contention, (b) medium resource contention, and (c) high resource contention.

### 6.3.1 Stopping Criteria

While Anneal and Metro perform slightly worse in higher contention environments, this can be compensated for by increasing the number of iterations the algorithms execute before terminating. Remember that Anneal and Metro are iterative algorithms that don't have a defined halting criteria—they execute for a defined number of iterations than halt. ICM, on the other hand, halts once it identifies a solution that cannot be improved by reallocating a resource block from one node to another. This means that increasing the number of iterations for Anneal and Metro improves the results of the solution they identify—albeit at diminishing returns.

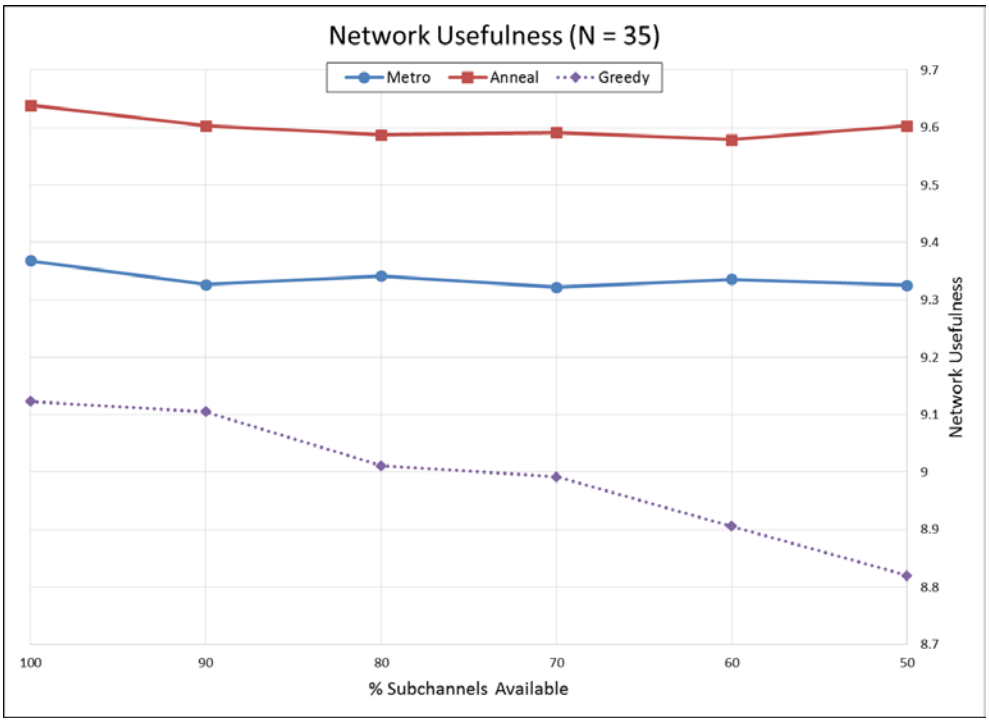
## 6.4 Network Degradation

We now look to test the scheduling algorithms' robustness to network degradation. Up until this point, all nodes in our network environment were permitted to receive data on any subchannel

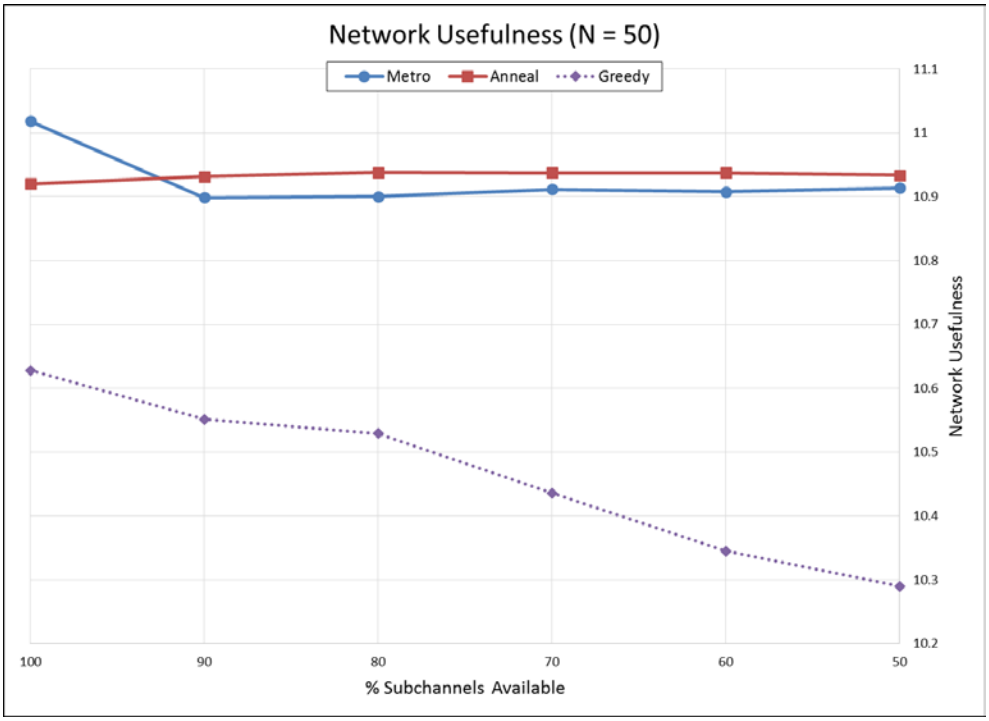
(ideal network conditions). In this subsection, we gradually degrade the network conditions by randomly disabling a percentage of each nodes' subchannels. In addition, we compare two different network sizes: 35 nodes and 50 nodes. Note that even though we used these network sizes in the previous subsection as a way of looking at the effects of network contention, these two subsections are not directly comparable since we are controlling network contention by both network size and subchannel availability (the previous section only used network size). As before, the priority of the nodes ranges from 1 to 5 and is uniformly distributed.

Looking at network usefulness, we see that in both the 35 node network (Figure 13(a)) and the 50 node network (Figure 13(b)), Metro and Anneal are robust against network degradation. They accomplish this by driving lower priority nodes to zero resources and allocating them to maximize their use. We saw in Figure 12 how, in an ideal network, some resources were still being allocated to lower priority nodes. In this test, those resources are being better allocated and thus the scheduling algorithms are more robust. Greedy, for comparison, is unable to make smart choices on resource allocation and thus as the total number of network resources decrease, so does the network usefulness.

We can also look at the same situation in terms of the average number of nodes receiving service. We say a node receives service if it is allocated enough resources that its video stream has a utility of 0.85. As expected, Figure 14 shows that in both a 35 node and 50 node environment, Metro and Anneal are robust against network degradation, with Anneal providing better overall performance.

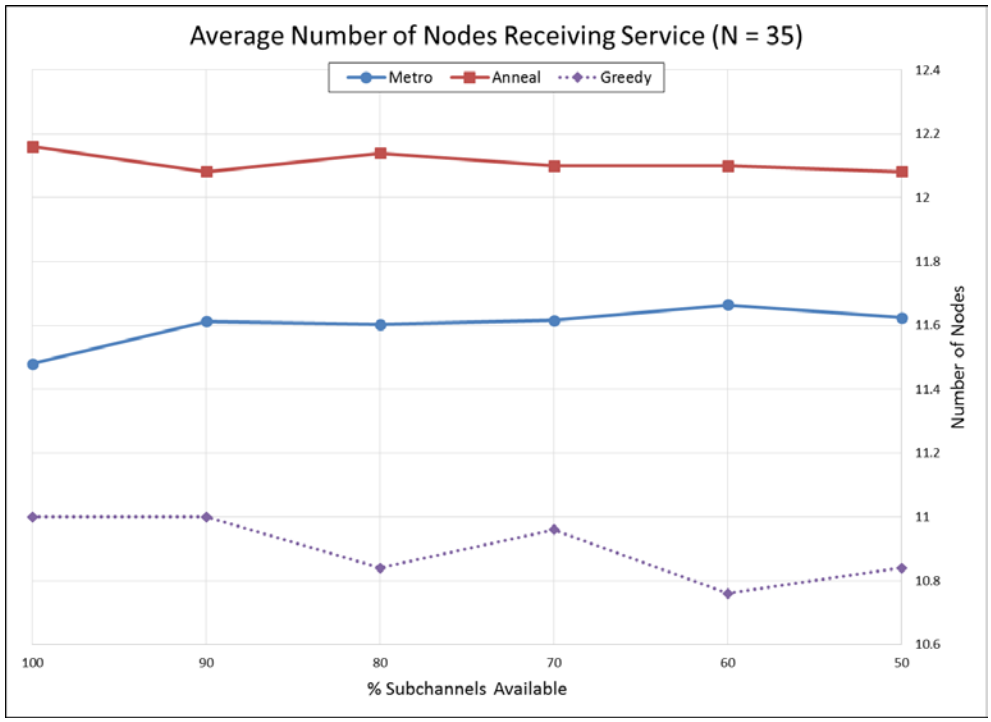



(a)

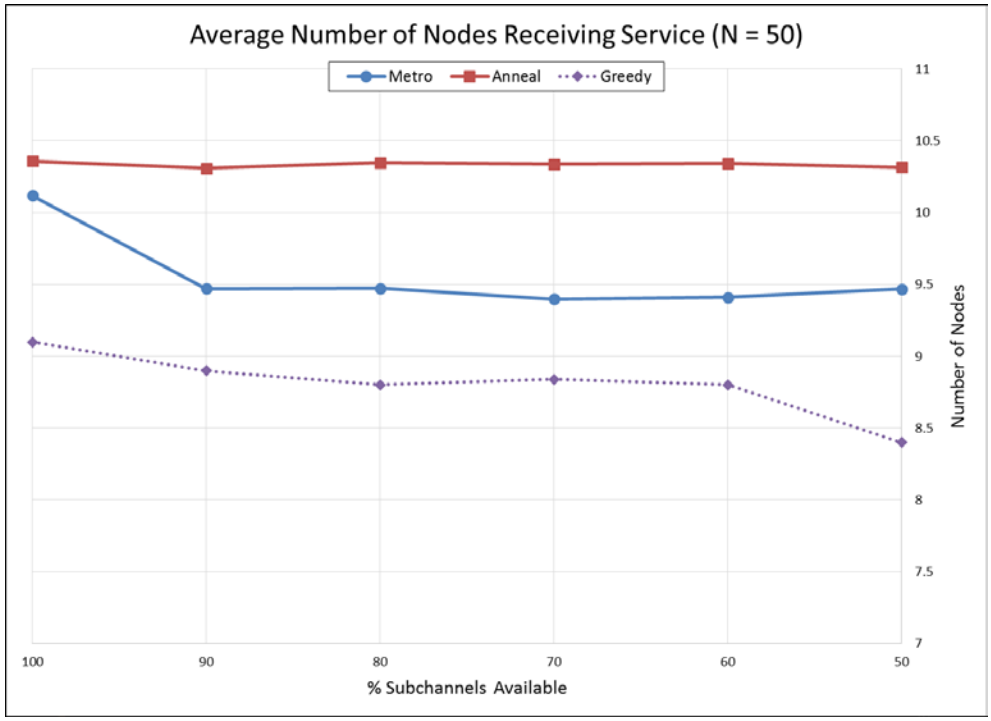



(b)

Figure 13. Network usefulness for varying degraded network environments. (a) 35 node network. (b) 50 node network.




(a)




(b)

Figure 14. Average number of nodes receiving service for varying degraded network environments. (a) 35 node network. (b) 50 node network.

### 6.5 Balancing Priority and Utility

Lastly, we test how well the scheduling algorithms balance the properties of priority and utility. In this test, we defined a 35 node network environment in which 25% of the nodes have Priority = 1 and are requesting a video stream (Dim/Moving/Large Target) for which a high bitrate is need in order to provide high utility. The other 75% of the users have Priority = 2 and are requesting a video stream (Light/Static/Large Target) for which high utility can be achieved with a much lower bitrate. Thus we are left with a scenario in which a small set of high priority nodes want to consume all network resources and crowd out the much large set of lower priority nodes requesting minimal resources. As before, each data point is an average of 25 simulations. We select the sigmoid utility function for all algorithms in these simulations.

We begin by looking at network usefulness, shown in Table 1. Metro, Anneal, and ICM all greatly outperform the Greedy algorithm. This is because Greedy allocates resources based on priority alone, so the high priority nodes are allowed to crowd out the more numerous lower priority nodes. Metro and Anneal perform best because they compute that greater network usefulness is achieved by allocating resources to the lower priority nodes, thereby allowing more users to utilize the network.

Table 1. Network usefulness for different scheduling algorithms in a priority vs. utility environment test.

| **Algorithm** | **Network Usefulness** |
|---|---|
| **Metro** | 7.68284 |
| **Anneal** | 6.70197 |
| **ICM** | 5.75250 |
| **Greedy** | 1.71347 |

We can see this in better detail if we look at Table 2. Here, we see that using Greedy only allows two high bandwidth nodes on the network, crowding out all other nodes. Metro, on the other hand, allocates resources to, on average, 14.24 nodes of priority 2. Yet, if one of those high priority nodes must receive network resources at all costs, then this can easily be achieved through the allocation of priorities to nodes, since usefulness is partially computed as a weighted product of utility and the inverse of priority.

Table 2. Average number of nodes receiving 0.85 utility in a priority vs. utility environment test.

| **Algorithm** | **Average Nodes** |
|---|---|
| **Metro** | 14.24 |
| **Anneal** | 13.52 |
| **ICM** | 12 |
| **Greedy** | 2 |

# 7. CONCLUSIONS AND FUTURE WORK

In this work, we looked at defining an algorithmic LTE resource scheduler that optimizes the usefulness of the network configuration. We began by developing a mathematical model of an LTE network and defining a new metric called usefulness. By computing the usefulness of a network's resource allocation, we can compare multiple resource allocations against each other in a quantitative manner. This results in the ability to apply known optimization algorithms to the problem of LTE network resource scheduling in order to identify the resource allocation that maximizes the network's usefulness.

We presented three different optimization algorithms: Iterated Conditional Modes (ICM), the Metropolis-Hastings Algorithm (Metro), and the Metropolis-Hastings Algorithm with Simulated Annealing (Anneal). We showed that although ICM appears to perform slightly better in the general case, Metro and Anneal allow for better customization as well as offering scaling and performance benefits.

We incorporated the concept of local control, an important issue for public safety practitioners. Not only does this support their desired ability to control their own networks, it can leverage their insights during incident response by creating a flexible scheduling algorithm that optimizes to the specific situation at hand, instead of a rigid algorithm that can only be optimized for the general case—thus achieving better results that can be tailored to specific emergency response scenarios.

We defined four different utility functions to test with our scheduling algorithms. After simulating each of the four functions with the scheduling algorithms, we concluded that using a sigmoid function provides the best performance. Its continuous, non-linear features are successful in allocating resources to the nodes where they are most useful.

Further testing of the scheduling algorithms showed that Metro and Anneal outperformed both ICM and Greedy. From a computational standpoint, both scale efficiently with network size. While most of the time Anneal performed better than Metro, due to its annealing ability while searching the solution space, this was not definitive. In our test of priority vs. utility, Metro was able to consistently achieve better results in terms of both network usefulness and average number of nodes receiving service. We suspect that further tuning of the multiple variables in Anneal may remove this discrepancy, but we leave this for a future effort.

In the future, we look to further refine the sigmoid utility function to allow for even better customization. We recognize that there are still efficiencies to be gained, such as driving resource allocation to nodes below the defined minimum utility to zero. Additionally, we look to further study the effects of the multiple parameters within the model and how they can be tuned, along with giving recommendations for certain defined scenarios.

Beyond this, we would like to implement our resource scheduling algorithms into a more robust and capable network simulator, such as ns3. With this, we could study more advanced topics such as the effects of mobility and the transitions from transient to steady-state when users enter or leave a cell sector.

We also received strong positive feedback from public safety personnel when we presented initial results of this work at the Fifth Video Quality in Public Safety (VQiPS) Workshop in Houston during the week of April 28th. Included in this was allowing feedback from end users into the algorithm instead of only relying on the backend system operations, using upcoming multicast standards, and enhancing priority functions to contain multiple variables. More advanced suggestions included the ability to make our model network agnostic so that it could be applied to future technologies not based on LTE and defining advanced network metrics and visualization tools.

## REFERENCES


[1] 3GPP. n.d. *www.3gpp.org*. http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core.

[2] 3GPP, ETSI. 2013. *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description; Stage 2.* Technical Specification, Sophia Antipolis Cedex: ETSI3GPP. http://www.etsi.org/deliver/etsi_ts/136300_136399/136300/10.11.00_60/ts_136300v101100p.pdf.

[3] 3GPP, ETSI. 2012. *Quality of Service (QoS) concept and architecture.* Technical Specification, Sophia Antipolis Cedex, France: ETSI 3GPP. http://www.etsi.org/deliver/etsi_ts/123100_123199/123107/10.02.00_60/ts_123107v100200p.pdf.

[4] Besag, Julian. 1986. "On the Statistical Analysis of Dirty Pictures." *Journal of the Royal Statistical Society. Series B* 45 (3): 259-302.

[5] Cerny, V. 1985. "Thermodynamical approach to the traveling salesman problem: An efficientcient simulated algorithm." *Journal of Optimization Theory and Applications* 45: 41-51.

[6] Dumke, Joel, Carolyn G Ford, and Irena W Stange. 2011. "The effects of scene characteristics, resolution, and compression on the ability to recognize objects in video." *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series.*

[7] Geman, S, and D Geman. 1985. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (5): 721-741.

[8] Hastings, W K. 1970. "Monte Carlo Sampling Methods Using Markov Chains." *Biometrika* 57 (1): 97-109.

[9] Kirkpatrick, S, C D Gelatt Jr, and M P Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220 (4598): 671-680.

[10] Metropolis, Nicholas, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. 1953. "Equations of State Calculations by Fast Computing Machines." *Journal of Chemical Physics* 21 (6): 1087-1092.

[11] Technologies, Agilent. n.d. *LTE Physical Layer Overview*. http://wireless.agilent.com/wireless/helpfiles/89600B/WebHelp/subsystems/lte/content/lte_overview.htm.

[12] US Department of Homeland Security. 2012. "Assessing Video Quality for Public Safety Applications Using Visual Acuity." Public Safety Communications Technical Report DHS-TR-PSC-12-11. http://www.firstresponder.gov/TechnologyDocuments/Assessing Video Quality for Public Safety Video.pdf

[13] US Department of Homeland Security. 2011. “Recorded-Video Quality Tests for Object Recognition Tasks.” Public Safety Communications Technical Report DHS-TR-PSC-11-01. http://www.pscr.gov/outreach/safecom/vqips_reports/RecVidObjRecogn.pdf

[14] US Department of Homeland Security. 2010. “Video Quality Tests for Object Recognition Applications.” Public Safety Communications Technical Report DHS-TR-PSC-10-09. http://www.pscr.gov/outreach/safecom/vqips_reports/Video_Quality_Tests_for_Object_Recognition_Applications.pdf

[15] Zyren, Jim. 2007. *Overview of the 3GPP Long Term Evolution Physical Layer*. White Paper, Freescale Semiconductor. http://www.freescale.com/files/wireless_comm/doc/white_paper/3GPPEVOLUTIONWP.pdf.

NTIA FORM 29
(4-80)

U.S. DEPARTMENT OF COMMERCE
NATIONAL TELECOMMUNICATIONS AND INFORMATION ADMINISTRATION

# BIBLIOGRAPHIC DATA SHEET

| 1. PUBLICATION NO. | 2. Government Accession No. | 3. Recipient's Accession No. |
|---|---|---|
| TR-15-511 | | |

| 4. TITLE AND SUBTITLE | 5. Publication Date |
|---|---|
| An Algorithmic Approach to Optimizing Network Resources for Public Safety LTE Networks | December 2014 |
| | 6. Performing Organization Code<br>NTIA/ITS.P |

| 7. AUTHOR(S) | 9. Project/Task/Work Unit No. |
|---|---|
| William Kozma Jr., Joel Dumke, Brent Johnson | 6736000-302 |

| 8. PERFORMING ORGANIZATION NAME AND ADDRESS | 10. Contract/Grant Number. |
|---|---|
| Institute for Telecommunication Sciences<br>National Telecommunications & Information Administration<br>U.S. Department of Commerce<br>325 Broadway<br>Boulder, CO 80305 | |

| 11. Sponsoring Organization Name and Address | 12. Type of Report and Period Covered |
|---|---|
| National Telecommunications & Information Administration<br>Herbert C. Hoover Building<br>14th & Constitution Ave., NW<br>Washington, DC 20230 | |

14. SUPPLEMENTARY NOTES

15. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)

This report examines the problem of how to allocate limited network resources in a public safety LTE network for the purpose of disseminating video streams to end users. We develop a mathematical model of the network environment, including the definition of a new metric called usefulness. We then apply known optimization techniques and algorithms to our model and analyze the results. We incorporate the concept of local control for public safety practitioners into our model through the use of a priority function and simulate its performance.

16. Key Words (Alphabetical order, separated by semicolons)

LTE, resource scheduling, video, optimization

| 17. AVAILABILITY STATEMENT | 18. Security Class. (This report) | 20. Number of pages |
|---|---|---|
| ☒ UNLIMITED. | Unclassified | 49 |
| ☐ FOR OFFICIAL DISTRIBUTION. | 19. Security Class. (This page)<br>Unclassified | 21. Price: |

# NTIA FORMAL PUBLICATION SERIES

## NTIA MONOGRAPH (MG)

A scholarly, professionally oriented publication dealing with state-of-the-art research or an authoritative treatment of a broad area. Expected to have long-lasting value.

## NTIA SPECIAL PUBLICATION (SP)

Conference proceedings, bibliographies, selected speeches, course and instructional materials, directories, and major studies mandated by Congress.

## NTIA REPORT (TR)

Important contributions to existing knowledge of less breadth than a monograph, such as results of completed projects and major activities. Subsets of this series include:

## JOINT NTIA/OTHER-AGENCY REPORT (JR)

This report receives both local NTIA and other agency review. Both agencies' logos and report series numbering appear on the cover.

## NTIA SOFTWARE & DATA PRODUCTS (SD)

Software such as programs, test data, and sound/video files. This series can be used to transfer technology to U.S. industry.

## NTIA HANDBOOK (HB)

Information pertaining to technical procedures, reference and data guides, and formal user's manuals that are expected to be pertinent for a long time.

## NTIA TECHNICAL MEMORANDUM (TM)

Technical information typically of less breadth than an NTIA Report. The series includes data, preliminary project results, and information for a specific, limited audience.

For information about NTIA publications, contact the NTIA/ITS Technical Publications Office at 325 Broadway, Boulder, CO, 80305 Tel. (303) 497-3572 or e-mail info@its.bldrdoc.gov.