Congestion Control and User Utility Function for Real-Time Traffic

Hengky Susanto and Byung Guk Kim

Abstract — Usage of multimedia communication in everyday life has seen a remarkable growth. Vast in Bandwidth demand often results in lower network performance and reduced quality of user experience during congestion. In this paper, we will address congestion control problem while providing quality of service (QoS) of delay for inelastic flows like video streaming, IPTV, etc. First, we propose an extension of an existing user utility function to capture user’s satisfaction over bandwidth allocation, QoS, and the cost to acquire the service. Second, we will study the relationship between these three entities.

Index Terms — Congestion control, network utility optimization, real time network.

I. INTRODUCTION

The increase of real-time applications, such as video streaming, VoIP, etc. in today’s Internet usage means the quality of service (QoS) is becoming more important in assuring user satisfaction with network performance. In this paper, we are extending network utility maximization (NUM) problem [18] to address QoS issues for real-time traffic, particularly relating to network delay. Real-time traffic is also known as inelastic traffic with sigmoidal like utility function [16], because in response to network congestion, the user utility may drop significantly when the transmission rate is decreased below the set threshold. This means when the bandwidth requirement or QoS is met, the application performance is constant. However, as soon as the transmission rate or the QoS drops below the threshold, the performance falls sharply to zero. For this reason, the utility maximization for real-time traffic is also known as non-convex optimization of which the problem of maximizing polynomial is an NP-hard problem [2] and finding the optimal solution is hard because the local minimum may not be the global maximum. In [1], J. W. Lee extends Kelly’s model [18] to address the non-convex problem and proposes sub-differential based solution with feedback. Thus, when congestion occurs, network influences users to decrease their transmission rate by adjusting the network price.

There are a number of other developments in NUM for real-time traffic. In [8], second order utility optimization is used in the construction of user utility function in real time environment. This framework, however, is tied to the active queue management mechanism used in the router and it is not an end-to-end solution. In [4], non-convex utility maximization is solved by decomposing the dual function into a number of sub-problems and solving it with Surrogate Subgradient to compute the optimal price. The authors proposed hybrid Particle Swarm Optimization and Sequential Quadratic Programming method to ensure the speed of convergence and the accuracy of zero duality gap. The methodology is centralized and requires too many steps which may not be practical for real life application.

Authors of [5] present a set of conditions to establish a distribution algorithm that converges to globally optimal solution and the dual decomposition also converges. Unfortunately, these conditions may not hold in many cases. While the authors of [5] also propose a price-based admission control and capacity planning to address non-convexity, neither approach provides a theoretically polynomial time and practically efficient algorithm (distributed or centralized) for non-concave utility maximization. In [6] and [7], Chiang et al. use a family of convex semidefinite programming relaxations based on sum of squares relaxation and Positivstellensatz theorem in real algebraic geometry. They apply centralized computational method to bound the total network utility in polynomial time. However, it is not responsive to distributed algorithm [5]. Another way to solve non-convex problem is to provide an approximation to make the problem convex [9]. The authors propose changing the underlying assumption of the architecture protocol which makes it a different and much easier problem to solve or to approximate NUM formulation [11]. The authors of [12] and [13] also discover a change of variables that alter the non-convex problem into a convex problem, defining the condition under which the problem is convex or has unique KKT points. However, changing the hard problem into an easier one contradicts optimization.

This paper aims to provide a framework to understand how QoS influences the network bandwidth distribution while resolving congestion, and achieving maximum user utility. Our previous work considers QoS in user utility function but only limited to elastic traffic [19]. This understanding enables the development of a practical solution to assure the network quickly solves congestion problem.

II. NON-CONVEX UTILITY FUNCTION

In this section, we develop user delay and cost utility functions to measure user satisfaction for QoS and the cost to incur the service. We also explore how queuing theory can be incorporated into the delay utility function.

A. User Delay Utility Function

Consider a communication network shared by a set of users $S$. The NUM formulation attempts to maximize the aggregate...
utility of users receiving bandwidth with utility function $U$ subject to limits on the link capacity

$$\text{maximize } \sum_{s \in S} U_s(x_s) \quad (1)$$

$$\text{s.t. } Ax \leq C$$

$$\text{over } x \geq 0$$

where, $C$ denotes a set of capacity of link $l$, for $l \in L$, where $L$ is set of links in the network and let $S$ be the set of users accessing the network. Associated with each user $s$, a single route $r$ is set up, which consist a set of links $l$ in route $r$, $r \in R$. Set $A_{lr} = 1$ if $l \in r$, that is link $l$ lies on route $r$, otherwise $A_{lr} = 0$. We have matrix $A = (A_{lr}, l \in L, r \in R)$, that is matrix $A$ contains information on link $l$ associated to resource $r$ and matrix $R$ is a set of route or paths that connect a source node and a sink node associated with user $s$.

The network rate control for each user $s$ can be derived by solving utility maximization problem with utility function

$$U_s(x_s) = U_{bw}(x_s) + U_{QoS}(x_s) + U_{cost}(x_s), \quad (2)$$

Where $U_{bw}(\cdot)$ is the rate allocation utility function, $U_{QoS}(\cdot)$ is the QoS utility function, and $U_{cost}(\cdot)$ is the cost utility function relative to user’s willingness to pay.

$$U_{cost}(x_s) = 1 - \frac{x_s \lambda_s}{m_s}$$

Also, $m_s$ denotes user’s willingness to pay, which allows users to influence amount of bandwidth allocation, and $\lambda_s$ is the price that network charges to user $s$. Thus, $x_s \lambda_s$ can be interpreted as the total cost that user has to pay. Ideally, the cost ratio is preferable to user when $m_s \geq x_s \lambda_s$, that is, the cost is less than the amount of money user is willing to pay, leading to higher perception of value for money, which is captured in $U_{cost}(x_s)$. Otherwise, user is paying too much.

In this paper, we assume the utility function is continuous, twice differentiable on $(0, +\infty)$, and the following properties are satisfied:

1. $U_{bw}(x_s), U_{QoS}(x_s), U_{cost}(x_s) \geq 0, \quad \forall x_s, \quad 0 \leq x_s \leq C_l$ and $U_{bw}(0), U_{bw}(0) = 0, U_{cost}(0) = 0, \forall x_s, s$, where $C_l$ is link capacity.

2. $U_{bw}(x_s)$ and $U_{QoS}(x_s)$ are twice differentiable.

3. $U_{bw}(x_s)$ and $U_{QoS}(x_s)$ are sigmoidal-like.

4. $\frac{dU_{bw}(x_s)}{dx_s}, \frac{dU_{QoS}(x_s)}{dx_s} \to \infty$, for all $0 \leq x_s \leq c$.

5. $\lim_{x_s \to 0} \frac{dU_{bw}(x_s)}{dx_s}, \lim_{x_s \to 0} \frac{dU_{QoS}(x_s)}{dx_s} < \infty, \forall s, s \in S$.

Assumption 1, 2, and 4 are commonly used in [3][1][19]. Assumption 5 can be interpreted as one that prevent starvation from transmitting data, since it implies the slope of the utility function increases to infinity as the rate approaches zero [14]. Assumption 3 corresponds to real time applications like video, voice services, and other multimedia applications. One utility function of real time application for bandwidth allocation satisfying condition 1-5 is

$$U_{bw}(x_s) = \frac{1}{1 + e^{-a x_s}}, \quad (3)$$

where the positive constant variable $a$ controls the steepness of the sigmoidal between minimum and maximum utility value.

The following user QoS utility function is modeled by incorporating delay function $d(x_s)$ and queuing theory [10] into the utility function; the function $d(x_s)$ is defined as follows:

$$d(x_s) = \sum_{l \in R_s} \frac{1}{x_s(t) - a_s(t)} + \sum_{l \in R_s} \frac{1}{x_s(t) - a_s(t)} \times \frac{|r_s|}{x_s - a_s}, \quad (4)$$

where $a_s(t)$ denotes the arrival rate at link $l$ associated to user $s$ and $x_s > a_s(t)$. Additionally, in the context of queuing theory, the allocated bandwidth $x_s$ can be interpreted as the processing rate and $a_s(t)$ is the minimum required bandwidth. Since the most congested link determines the amount of flow traverse through path $r_s$, for link $l \in r_s$. Furthermore, QoS utility function

$$q_s(x_s) = \frac{x_s}{d_s(x_s)}$$

is assumed to be subject to exponential decay, such that the quality decreases proportionally to increase of delay and congestion. Therefore longer delay implies decrease in quality and vice versa. Thus, the sigmoidal QoS utility function is formulated as follows.

$$U_{QoS}(x_s) = \frac{1}{1 + e^{-k Q_s(x_s)}} \quad (5)$$

where $x_s > a_s > 0$ and the constant variable $b$ controls the steepness of the sigmoidal. Additionally, the arrival rate $a_s$ can be interpreted as user’s minimum demand, hence the condition $x_s - a_s > 0$ must be satisfied. Moreover, to prevent $x_s (x_s - a_s)$ from growing polynomially, it is normalized such that $\sqrt{x_s (x_s - a_s)}$ linear grow can be achieved. Another factor to consider is that rate $x_s$ is proportional to the length of the path $r_s$. The longer the path implies longer end-to-end delay along path $r_s$, so the rate $x_s$ must be increased to reduce the delay. Thus,

$$U_{QoS}(x_s) = \begin{cases} \frac{1}{1 + e^{-\frac{1}{\sqrt{x_s (x_s - a_s)}}}}, & x_s > a_s \\ 0, & \text{otherwise} \end{cases}$$

And, based on Little’s theorem [10], the implication from $x_s < a_s$ is that the average queue length $Q_s(x_s)$ will grow
infinitely and cause a bottleneck, such that, as \( \frac{d}{dx} \geq 1 \), \( \lim_{x \to \infty} Q_s(x_s) = \infty \). Consequently, this will lead to further performance degradation that is when network starts dropping packets. Thus, \( U_{QoS}(x_s) = 0 \) implies user is very unsatisfied.

Generally, the property of function \( U_{pr}(x_s), U_{QoS}(x_s) \), and \( U_{cost}(x_s) \) is between 0 and \( 1 - \epsilon \). User utility is equal to 0 when user is not satisfied, otherwise it is equal to \( 1 - \epsilon \), where constant variable \( \epsilon \leq 0 \). This definition allows us to provide the simplicity for later analysis where we will dissect the relationship between three functions later.

**B. Bandwidth Allocation**

As (2) is a non-concave utility function, problem (1), the primal problem, is a non-convex programming problem, which is usually more difficult to solve [15][1]. However, the lower bound from Lagrange dual function for problem (1) depends upon variable \( \lambda \) and the dual objective function is always concave because it is the infimum of a family of affine function in \( \lambda \) [15]. Thus, the Lagrange dual function is always convex, which is easier to solve, and the separable property of dual problem also allows an easier implementation in distributed manner. However, since the primal problem (1) is a non-convex, the duality gap may exist between the primal and its dual [15]. Although, the dual problem is solvable, solving the dual may not obtain the optimal primal solution.

The Lagrange form associated to problem (1) is

\[
L(x, \lambda) = \sum_{s \in S} U_s(x_s) - \lambda^T (C - Ax),
\]

\[
= \sum_{s \in S} U_s(x_s) - \sum_{s \in S} \lambda_s x_s + \sum_{l \in L} \lambda_l C_l
\]

where \( \lambda \) is known as a set of Lagrangian multipliers of which also interpreted as the link price [10] [1] and

\[
\lambda_s = \sum_{l \in L} \lambda_l.
\]

The dual problem of (1) is defined as follows

\[
\min D(\lambda)
\]

\[
s.t. \ \lambda \geq 0,
\]

where \( \lambda \) is a vector of zeros and dual function \( D(\lambda) = \max_{\lambda \geq 0} L(x, \lambda) \), which is convex function. Nonetheless, it might not be everywhere differentiable because the slope of the sigmoidal utility function at the inflection point is infinite. Thus, a simple gradient based algorithm may not find the solution when \( D(\lambda) \) does not have a gradient at the point where it is not differentiable [1]. Moreover, the authors of [1] further study the property of the dual problem by using theory of the sub differentials.

In the following step, user decide the transmission rate \( x_s(\lambda_s) \) at price \( \lambda_s \) by solving

\[
x_s(\lambda_s) = \arg \max_{0 \leq x_s \leq x_s^{max}} U_s(x_s), \quad (6)
\]

where \( x_s(\lambda_s) \) denotes bandwidth allocation price at \( \lambda_s \). Given the rate allocation, we can solve the dual problem \( D(\lambda) \) with a subgradient projection method, which is formulated using an iterative algorithm, because the subgradient of \( D(\lambda) \) is not unique and differentiable everywhere. Thus, the projection method [1] is

\[
\lambda^{(t+1)} = \left[ \lambda^{(t)} - \sigma^t \left( C - Ax(\lambda^{(t)}) \right) \right]^+, \quad (7)
\]

where \( x(\lambda^{(t)}) \) is the solution of (7) and \( C - Ax(\lambda^{(t)}) \) is a subgradient of \( D(\lambda) \) at link price \( \lambda = \lambda^{(t)} \) and \( x(\lambda^{(t)}) \) denotes the rate allocation at \( \lambda^{(t)} \). The time \( t \) denotes the iteration index, \( 0 \leq t \leq \infty \).

Furthermore, if \( \lambda^{(t)} \) takes on a negative value, then it just returns zero, or minimum price \( \lambda^{min} \), where \( \lambda^{min} \geq 0 \), and \( \lambda^{min} \) can be interpreted as the network’s operation cost, such as the cost of network operators, electricity, etc. [16]. In this case, the link price is never allowed to be negative

\[
\lambda^{(t+1)} = \begin{cases} 
\lambda^{(t+1)} & \lambda^{(t)} > \lambda^{min} \\
\lambda^{min} & \text{otherwise}
\end{cases}
\]

Moreover, to assure \( \lambda^{(t)} \) converges to the optimal price \( \lambda^* \), the optimal solution for dual \( D(\lambda) \), we must have an appropriate *step size* \( \sigma^{(t)} \). The parameter \( \sigma > 0 \) is the step size that controls the tradeoff between a convergence guarantee and the convergence speed. If it is sufficiently small, the algorithm will converge to the right solution, but if it is too small the convergence becomes too slow. Turning this parameter is not easy. Additionally, the subgradient based algorithm may not always converge to the solution if the step size is constant. Thus, the authors of [15] proposed diminishing step size to adjust \( \sigma \) as the following sequence:

\[
\sigma^{(t)} \to 0, \text{as } t \to \infty \text{ and } \sum_{t=1}^{\infty} \sigma^{(t)} = \infty.
\]

For instance, \( \sigma^{(t+1)} = \frac{K_1}{K_2 + t} \), where \( K_1 \) and \( K_2 \) are constant variables, \( K_1, K_2 > 0 \).

Furthermore, in distributed environment, to determine the price of link \( l \in L \), the projection method in (7) can be formulated as follows.

\[
\lambda^{(t+1)}_l = \left[ \lambda^{(t)}_l - \sigma^t \left( C_l - \sum_{s \in S(l)} x_s(\lambda^{(t)}_s) \right) \right]^+ , \quad (8)
\]

for \( l = 1, 2, \ldots, L \), and \( \lambda^{(t+1)}_l \geq \lambda^{min} \). This implies that solving the problem at the local links also solves (7), which means obtaining the optimal price at the link level also solves the dual problem by simply adjusting the price based on its congestion level.

**C. Big Picture on Congestion Control**

The feedback loop in (7) is inspired from an economic
standpoint, where price is adjusted when bottleneck occurs at
\( C_t < \sum_{s \in \mathcal{S}(t)} x_s \), for \( \forall t \in L \). Then the network notifies user \( s \) about the new pricing \( \lambda_s = \sum_{t \in \mathcal{S}_s} \lambda_t \) and user adjusts the transmission rate by solving (6) with the updated price. The loop continues until price converges and the bandwidth allocation is feasible, that is the condition \( \sum_{s \in \mathcal{S}(t)} x_s \leq C_t \), for \( \forall t \in L \), is satisfied [18]. In the final state of the bandwidth allocation implementation, the allocation is determined by the most congested link in its path. Thus, bandwidth allocation for user \( s \) is \( x_s = \min\{x_{r_s}\} \), for \( \in r_s \).

III. RESULTS AND ANALYSIS

This section demonstrates the outcomes from incorporating QoS into user utility function and how network distributes bandwidth to users.

A. Results and Analysis

Here, we plot a simple rate allocation of three sigmoidal flows, associated to user (flow) 0, 1, and 2, over single link as it is shown in fig. 1 below, connecting \( s_0 \), \( s_1 \), and \( s_2 \) with \( r_0 \), \( r_1 \), and \( r_2 \) respectively.

![Fig. 1. Single link network](image)

Fig. 1. Single link network

The setup is the three flows traverse over router \( R_1 \) and \( R_2 \) sharing a congested link \( R_1 R_2 \). Initially, every user (flow) has identical willingness to pay such that \( m_1 \) = \( m_2 \) = \( m_3 \) = 10. Also, the arrival rate of each flow is 2 unit of bandwidth capacity is 10 unit bandwidth and \( \sigma \) = 0.5. The constant variable in (3) \( a=1 \) and in (5) \( b=1 \). All users initially transmit data at \( x_0 = x_1 = x_2 = C = 10 \). The initial price is \( \lambda_t = \lambda_{\text{min}} = 1 \) but \( \lambda_{\text{min}} \leq \lambda_t \) throughout the simulation. Additionally, the entire flows are inelastic traffic.

Initially, in fig. 2.a, the algorithm quickly converges and the bandwidth is equally divided among three flows. Next, at 60th iteration, after the first conversion, user 0 and 1 decide to increase their willingness to pay at \( m_0 = 110 \) and \( m_1 = 60 \), but user 2 maintains the willingness to pay at \( m_2 = 10 \). The rate allocation is adjusted according to user’s willingness to pay and those who pay more receive more bandwidth. However, the convergence takes much longer time than the previous conversion as it is shown in fig. 2.a, because the choice of the step size is too small for this scenario. Furthermore, the rate allocation is reflected in their utility values as it is shown in fig. 2.b. Notice that, during the convergence, around 117th iteration, there is no significant change as the algorithm continues to converge.

B. Dissecting User Utility

In this section, in order to understand the behavior of utility model, we dissect the user utility function in greater detail. The user utility function of user \( s \) (2) and, given \( \lambda_s \) and \( m_s \), bandwidth allocation \( x_s \) is solved with eq. (6). Solving (6) may require trying different values of \( x_s \) to attain maximum \( U_s(x_s) \) from a set of values between 0 and \( x_s^{\text{max}} \). By dissecting \( U_s(x_s) \), it gives us a “glance” of the relationship between the three aspects of \( U_s(x_s) \), that is \( U_{bw}(x_s) \), \( U_{QoS}(x_s) \), and \( U_{\text{cost}}(x_s) \), and provides insight into the impact of network pricing on bandwidth allocation.

![Fig. 3. Scenario 1: \( U_{bw} \), \( U_{QoS} \), and \( U_{\text{cost}} \) comparison.](image)

In the following step, we run a simulation in a single link network environment with three users, as illustrated in fig. 1. The goal of this simulation is to observe the deviation of \( U_{bw}(x_s) \), \( U_{QoS}(x_s) \), and \( U_{\text{cost}}(x_s) \), as well as how each aspect of \( U_s(x_s) \) influences the attainment of \( x_s \). The setup of the simulation work is to experiment with different inputs from the largest value \( x_s^{\text{max}} \) to zero. For simplicity and clarity of the explanation, let set \( x_s^{\text{SET}} = \{v_0, v_1, \ldots, v_n\} \), where value \( v_0 \) is the largest value in \( x_s^{\text{SET}} \) and \( v_n \) be the smallest value in the set. For example, \( v_0 = x_s^{\text{max}} \) and \( v_n = 0 \) respectively.

Fig. 3 depicts the process of achieving \( \arg \max \{U_s(x_s)\} \) by substituting different value from \( x_s^{\text{max}} \) down to zero and \( \arg \max \{U_s(x_s)\} \) is achieved at 176th iteration. Observe that the lines associated with \( U_{bw}(x_s) \) and \( U_{QoS}(x_s) \) in fig. 4 are rather flat compared to the increasing line of \( U_{\text{cost}}(x_s) \). The reason of the occurrence of the flat line is, for \( \{x_0^0, x_1^1, \ldots, x_n^n\} \subseteq x_s^{\text{SET}} \), where \( 0 \leq i \leq j \leq n \), because \( x_j^1, \ldots, x_j^n, \ldots, x_0^0 \geq x_s^{\text{min}} \), and given \( \lambda_s, x_s = \{x_0, \ldots, x_1, \ldots, x_n\} \) attain maximum value from (6).

In different scenario, as illustrated in fig. 4, function \( U_{\text{cost}}(x_s) \) increases as \( x_s^j \) decreases, which causes the
nominator of \( \frac{x_s^j \lambda_s}{m_s} \) in \( U_{\text{cost}}(x_s) \) to decrease also, for \( x_s^j > 0 \). In other words, \( x_s \lambda_s \) can be interpreted as the cost to incur the service; function \( U_{\text{cost}}(x_s) \) measures user utility upon the cost that user must pay to the network. So one explanation for this phenomenon is that the increase in user utility \( U_s(x_s) \) is not necessarily caused by being satisfied for receiving more bandwidth or higher quality, but the increase in satisfaction may be caused by user is spending less money for the bandwidth and quality received. Thus, this utility function captures and measures all three important aspects of bandwidth allocation which are: utility for bandwidth, QoS, and amount of money user must pay for the service.

In another scenario, notice in fig. 4 that utility function \( U_s(x_s) \) is slightly increasing but \( U_{\text{bw}}(x_s) \) and \( U_{\text{QoS}}(x_s) \) are decreasing even though \( U_{\text{cost}}(x_s) \) is increasing. User may seem satisfied with the reduction in the cost to obtain the service, but user appears dissatisfied with bandwidth allocation and the QoS. We observe that the decrease in \( U_{\text{bw}}(x_s) \) and \( U_{\text{QoS}}(x_s) \) caused by \( x_s < x_s^{\text{min}} \). Hereafter, an interpretation of this occurrence is that the dissatisfaction captured in \( U_{\text{bw}}(x_s) \) and \( U_{\text{QoS}}(x_s) \) is compensated by satisfaction captured in \( U_{\text{cost}}(x_s) \). So, after 55th iteration, as it is shown in fig. 4, \( \forall x_s^j \) will result in the decrease of \( U_s(x_s) \). Let \( x_s^j \) be the least value which attains the maximum value of \( U_s(x_s) \). The phenomenon beyond 55th iteration is summarized as

\[
x_s(\lambda_s) < x_s^j = \arg \max_{0 \leq x_s, x_s \leq x_s^{\text{max}}} \left( U_s(x_s^j) \right).
\]

Nevertheless, the satisfaction user derives from low cost is conditioned upon user receiving the required minimum bandwidth and QoS. In essence, the level of user utility \( U_s(x_s) \) is determined by the balance of the cost that user has to incur, the amount of bandwidth, and the quality of user experience.

**IV. ADMISSION CONTROL**

The general objective of this thesis is to resolve network congestion and keep the network traffic under control by solving problem (1). However, network may not always able to meet user’s minimum bandwidth requirement. Therefore, network must take an active role in implementing admission control and select user request that network can support. In our scheme, user’s admission to network is decided by the relations between current pricing, QoS, network revenue, and the price the user is willing to pay measured through function \( \vartheta_s(x_s(\lambda)) \).

\[
\vartheta_s(x_s(\lambda)) \approx x_s(\lambda) \cdot \frac{\lambda_s}{\lambda} \cdot \frac{x_s^{\text{min}}}{x_s^{\text{max}}}, \quad x_s(\lambda) > 0
\]

\[
\vartheta_s(x_s(\lambda)) = \frac{\lambda s}{\lambda} \cdot \frac{x_s^{\text{min}}}{x_s^{\text{max}}} \quad \text{Otherwise}
\]

for \( x_s^{\text{min}} \geq 0 \) and \( \lambda, \lambda_s > 0 \), where the price user is willing to pay \( \hat{\lambda}_s = \frac{m_s}{x_s^{\text{max}}} \) and \( \tau \) is a positive constant. The idea is to find users with the highest \( \vartheta_s \) value from \( \vartheta_s = \vartheta_s(x_s(\lambda)) \). In other word, network tries to admit a set of user who are willing to pay higher price and easier to satisfy. One way to interpret the model is, when a user with \( \frac{\lambda_s}{\lambda} > 1 \) can be interpreted as user is willing to pay higher price than the given network price \( \lambda \), and when \( \frac{x_s^{\text{min}}}{x_s^{\text{max}}} > 1 \) may be interpreted as user is currently satisfied with the current bandwidth allocation \( x_s \) price \( \lambda \). Hence, the maximization problem is defined as follow.

\[
\max \sum_{s \in S} \vartheta_s z_s \quad s.t. \quad Axz \leq C \
\]

\[
z_s \in \{0,1\} \quad \forall s \in S, \text{ over } 0 \leq x_s^{\text{min}} \leq x
\]

where \( z_s = 1 \) if user \( s \) is selected, otherwise \( z_s = 0 \).

User selection for network admission is similar to solving the knapsack decision problem that is, given a set of \( |S| \) users, with each user associated with some value \( \vartheta_s \) and the minimum bandwidth requirement \( x_s^{\text{min}} \). The objective is to select some of the users that result in the maximal total \( \sum_{s \in S} \vartheta_s \), while obeying the fixed maximum total link capacity \( C_l \), such that \( \sum_{s \in S(l)} x_s \leq C_l \). However, Knapsack decision problem turns out to be another NP-hard problem. Thus, finding the resolution is at least as difficult as the decision problem, and there is no known polynomial algorithm which can tell, given the solution, whether it is optimal. Therefore, we propose a greedy based solution to select users in a centralized manner is designed as follows.

**User selection algorithm:**

1. \( \vartheta_s^{\text{max}} = \max \left\{ \vartheta_{\text{SET}(l)} \right\} \)
2. \( x_s^{\text{min}} = \text{get_bandwidth} (\vartheta_s^{\text{max}}) \)
3. If \( (x_s^{\text{min}} + \sum_{s \in S} x_s \leq C_l, \forall s, l \in \text{route } r_s) \)
   then
   
   4. Reserve \( l \) for user \( s \), for \( \forall l, l \in r_s \)
   5. \( C_l = C_l - x_s^{\text{min}} \) for \( \forall l, l \in r_s \)
   6. \( \hat{S} = \hat{S} + s \)
   7. \( \vartheta_{\text{SET}} = \vartheta_{\text{SET}} - \left\{ \vartheta_s^{\text{max}} \right\} \)
   8. Repeat from line 1 until \( \vartheta_{\text{SET}} = 0 \) // until \( \vartheta_{\text{SET}} \) is empty

Let \( \vartheta_{\text{SET}} \) denotes a set of \( \vartheta_s \) that is associated with user and \( \hat{S} \) denotes a set of users admitted into network. In line 1 and 2 of User selection algorithm, given \( \vartheta_s^{\text{max}} \), \( x_s^{\text{min}} \) is retrieved. In line 3, the algorithm verifies whether the link has sufficient capacity to provide at least \( x_s^{\text{min}} \) and \( x_s^{\text{min}} \) has not been
included from previous run, and then executes line 4, 5, and 6. Next, $g_{\text{max}}$ is removed from the set in line 6. We assume that network begins to provide service as soon as the user is admitted into network.

V. SIMULATION

In this section, we demonstrate the rate allocation algorithm asymptotically converges after the admission control. The simulation setup is similar to the setup in the previous section shown in fig. 1, with an additional flow.

<table>
<thead>
<tr>
<th>Table 1. the setup and initialization for the simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wtillness to pay</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Min. Bandwidth Demand ($Q(x)^m$)</td>
</tr>
<tr>
<td>Initial rate Allocation ($x_i$)</td>
</tr>
<tr>
<td>$b_i$</td>
</tr>
</tbody>
</table>

Fig. 3.a and 3.b: Rate allocation and user utility

Initially, the total initial users’ demand exceeds the link capacity, where the total demand is $40 > 10$. Consequently, network increases the price and force users to lower their transmission rate. Note, after the initial bandwidth allocation, user 3 suffers poor performance and achieve zero utility shown in fig. 3.b because network is unable to meet the required minimum bandwidth. Thus, network performs admission control by solving (10) to compute $\theta$ value of each user. Then, at iteration 100, the network makes its selection according to $\theta$ value, which results in not admitting user 3 because user 3 has the least $\theta$ value. After the selection, the spare bandwidth is distributed among to the admitted users, the fig. 3 illustrates that the rate distribution algorithm also converges after user 3 is removed from the network. Moreover, note that user 0, 1, and 2 also achieve higher utility after user 3 is removed, as shown in fig. 3.b. The increase of user utility reflects the increase in bandwidth allocation shown in fig. 3.a. The objectives of this simulation to provide a simple demonstration of how admission control is practiced and to show that the algorithm also achieves convergence even after admission control.

VI. CONCLUSION

With the additional of QoS utility function, minimum bandwidth requirement is not only determined by the required transmission rate of the applications but also by the minimum delay that user wants to achieve. Furthermore, QoS utility may have the strongest influence in deciding how much bandwidth should be allocated because the allocated bandwidth must be greater than the minimum bandwidth requirement determined in the QoS utility function $U_{\text{QoS}}(x)$. Otherwise the delay will grow exponentially [10]. Furthermore, we also introduce cost utility function $U_{\text{cost}}(x)$ to measure user’s satisfaction over the amount of money spent for the service relative to user’s willingness to pay. This utility function provides us with deeper understanding of how the quality of the connection is valued in relation to user’s willingness to pay. In conclusion, as we demonstrated in the simulation, there must be balance between the cost incurred for the service and the quality of the service, where the bandwidth, QoS, and cost utility functions reached an equilibrium. There are open questions to be addressed in congestion control and bandwidth allocation for user utility with multi layers application.

REFERENCES