Optimal Delay Region for Cross-Layer Resource Allocation

Wooyul Lee, Kibeom Seong, and John M. Cioffi
Department of Electrical Engineering
Stanford University, Stanford, CA 94305
Email: {wylee, kseong, cioffi}@stanford.edu

Abstract—This paper characterizes the delay region that is quite useful for multi-user packet scheduling. In a quasi-static channel, the optimal delay region is defined as every set of total delay vectors achieved by some scheduling policies when there are no further packet arrivals. Each user’s total delay is equivalent to the amount of time until that user’s queue backlog is cleared. The optimal delay region shows the fundamental limit on each user’s achievable queueing delay and describes the trade-off among the users in terms of queueing delay when different scheduling policies are applied. In addition, the concept of the delay region is shown to be extendable to each packet’s average delay as well as total delay.

I. INTRODUCTION

Future wireless networks will be driven by a variety of ubiquitous broadband services that require different quality of service (QoS). In order to guarantee each user’s QoS satisfaction, cross-layer approaches to resource allocation have been actively researched for multi-user communication systems. Cross-layer resource allocation significantly improves throughput, delay, and fairness properties by considering both channel state information (CSI) and queue state information (QSI). Each user’s data rate needs to be carefully determined to satisfy various QoS requirements. In particular, the delay constraint of each service category as well as fairness among the users are important issues to be addressed. Hence, it is essential to design schedulers that effectively control each user’s queueing delay.

Much research in cross-layer resource allocation has focused on designing multi-user packet schedulers that maximize throughput or minimize the overall queueing delay [1]. However, with different delay constraints on each service, which will be fairly common in future networks, precise control of each user’s queueing delay is crucial for the satisfaction of heterogeneous QoS. One good example to this end is the exponential scheduling policy which has been proved to be path-wise optimal in the sense that it minimizes the maximum of weighted queue length under heavy traffic regime [2]. In order to design intelligent schedulers that satisfy different delay constraint imposed on each service, it would be very useful to understand the fundamental limit and trade-off on each user’s achievable queueing delay given the channel and queue states.

From this motivation, this paper proposes the concept of optimal delay region, which is derived from solving draining problems. Like in [3][4][5], this paper addresses how to drain the queue backlogs without further packet arrivals. [4] presents a scheduler that combines channel information with file size information to reduce the total completion time, and [5] studies a buffer draining problem with a hybrid automatic repeat request (H-ARQ) retransmission strategy. This type of draining problems can provide some intuition and serve as a good reference for more general situations with new packet arrivals. We also consider a quasi-static channel where the channel state remains unchanged for the time duration of interest.

This paper characterizes the optimal delay region, which can be useful in designing smart schedulers with a desirable multi-user delay property as well as in evaluating the delay performance of a scheduler. The optimal delay region is defined as every set of total delay vectors achievable by using some scheduling policies, where each user’s total delay is equivalent to the amount of time until that user’s queue backlog is cleared. Thus, the optimal delay region provides the ultimate delay limits when the channel and initial queue states are given, and also it characterizes the trade-off among the users in terms of queueing delay that is essential in designing intelligent schedulers satisfying each user’s different delay requirement.

This paper shows that the boundary of optimal delay region can be achieved by applying queue proportional scheduling (QPS) and extended QPS. QPS is a type of minimum draining time (MDT) scheduling policy, which has been shown to be throughput optimal [6][7][8]. Under the QPS policy, each user’s allocated rate is chosen to be proportional to that user’s packets or bits and lies on the boundary of the achievable rate region. Without further packet arrivals, QPS is shown to minimize the time until all the queue backlogs are cleared. Recently, [8] proved that under the QPS policy, the average queueing delay of each user becomes equalized regardless of channel conditions and input traffic patterns; moreover, by using scaled queue-state vector, each user’s average queueing delay relative to other users can be arbitrarily scaled, an essential property for satisfying the different delay constraint on each user.

Because QPS minimizes the draining time and equalizes every user’s queueing delay when there are no further packet arrivals, QPS always achieves at least one boundary point of the optimal delay region. Other boundary points are shown to be achieved by using a series of QPS that is called the extended QPS policy. This paper presents an efficient optimization
algorithm to find the scheduled rate tuples under the extended QPS policy, which is based on bisection search combined with a convex-feasibility check. Therefore, intelligent scheduling policies can be efficiently designed to achieve any boundary points of optimal delay region for QoS satisfaction. Also, the delay performance of some scheduling policies can be evaluated based on the optimal delay region. The concept of the delay region is extendable to each packet’s average delay as well as total delay, and the achievable delay region in terms of average delay is also provided in this paper.

Notation: Vectors are bold-faced. \( \mathbb{R}^n \) denotes the set of real \( n \)-vectors and \( \mathbb{R}_{+}^n \) denotes the set of nonnegative real \( n \)-vectors. The curled inequality symbol \( \succeq \) (and its strict form \( \succ \)) is used to denote the componentwise inequality between vectors: \( x \succeq y \) means \( x_i \geq y_i \), \( i = 1, 2, \ldots, n \).

II. SYSTEM MODEL AND PROBLEM DEFINITION

This paper considers a quasi-static \( K \) user broadcast channel where the channel gain vector \( \mathbf{h} \) remains fixed for the duration of interest, \( T \). With the sum power constraint of \( P \), the capacity region for the duration of \( T \) is denoted by \( \mathcal{C}(\mathbf{h}, P) \). For example, superposition coding is shown to be an optimal coding scheme that achieves the entire capacity region of Gaussian broadcast channels [9].

The queueing system consists of \( K \) output queues that hold the incoming packets to each user. The queue state vector at time \( t \) is denoted by \( \mathbf{Q}(t) = [Q_1(t), \ldots, Q_K(t)]^T \) (bits) where \( Q_k(t) \) is the number of bits in user \( k \)'s queue at time \( t \). The scheduler is assumed to have both CSI and QSI, and it determines the rate vector \( \mathbf{R}(t) \) (bits/sec) at each scheduling period \( T_s << T \) (sec), based on the scheduling policy \( S(\mathcal{C}(\mathbf{h}, P), \mathbf{Q}(t)) \). The time interval of \([t, t + T_s)\) is considered as a time slot \( t \), and \( \mathbf{R}(t) \) denotes a rate vector served in \([t, t + T_s)\). It is assumed that \( T_s \) can be arbitrarily small in order to obtain a continuous delay region. With perfect CSI at the transmitter, any \( \mathbf{R}(t) \in \mathcal{C}(\mathbf{h}, P) \) can be scheduled as far as \( \mathbf{R}(t)T_s \leq \mathbf{Q}(t) \). This paper considers a draining problem where initial queue backlogs are presented, and no further packets arrive until every queue becomes empty [3] [4] [5]. Then, the queue state at the end of each scheduling period satisfies the following relation.

\[
\mathbf{Q}(t + T_s) = \max \{ \mathbf{Q}(t) - T_s \mathbf{R}(t), 0 \}, t = 0, T_s, 2T_s, \ldots
\]

This policy is deterministic since no random packet arrivals are assumed. Each user’s total delay has the following definition.

Definition 1: With an initial queue state \( \mathbf{Q}(0) \) and a scheduling policy \( S(\mathcal{C}(\mathbf{h}, P), \mathbf{Q}(t)) \), the total delay vector \( \mathbf{D}_T(\mathbf{Q}(0), S) = [D_{T,1}, \ldots, D_{T,K}]^T \) is defined as

\[
D_{T,k} = \min_{t, \ s.t.} Q_k(t) = 0, \ k = 1, \ldots, K.
\]

In other words, \( D_{T,k} \) is the time until user \( k \)'s queue is cleared. Thus, the achievable delay region of \( S \) can be defined as follows.

Definition 2: For an initial queue state \( \mathbf{Q}(0) \) and a capacity region \( \mathcal{C}(\mathbf{h}, P) \), the delay region of a scheduling policy \( S \) is defined as

\[
\mathcal{D}_T(\mathbf{Q}(0), S) = \bigcup_{\mathbf{D} \in \mathbb{R}_+^K} \{ \mathbf{D}_T(\mathbf{Q}(0), S) + \mathbf{D} \}.
\]

Denote a set of all the possible scheduling policies by \( S \). Then, the optimal delay region is defined as

\[
\mathcal{D}_T(\mathbf{Q}(0)) = \bigcup_{S \in S} \mathcal{D}_T(\mathbf{Q}(0), S).
\]

Likewise, each user’s average delay can be defined.

Definition 3: For an initial queue state \( \mathbf{Q}(0) \) and a scheduling policy \( S \), the average delay vector \( \mathbf{D}_A(\mathbf{Q}(0), S) = [D_{A,1}, \ldots, D_{A,K}]^T \) is defined as

\[
D_{A,k} = \frac{1}{Q_k(0)} \sum_{i=0}^{D_{T,k}/T_s - 1} iT_s \{ R_k(iT_s)T_s \}, k = 1, \ldots, K.
\]

where \( R_k(iT_s)T_s \) is the number of bits served during \( iT_s \leq t < (i + 1)T_s \), and \( Q_k(0) = \sum_{i=0}^{D_{T,k}/T_s - 1} R_k(iT_s)T_s \). For simplicity, it is assumed that \( iT_s \) is the delay of the bits that are served during \( iT_s \leq t < (i + 1)T_s \). Thus, the average delay \( D_{A,k} \) is equivalent to the average sojourn time of user \( k \)'s packets. Then, the optimal delay region in terms of average delay or the optimal average delay region has the following definition.

Definition 4: For an initial queue state \( \mathbf{Q}(0) \) and a capacity region \( \mathcal{C}(\mathbf{h}, P) \), the average delay region of a scheduling policy \( S \) is defined as the union of achievable average delay

\[
\mathcal{D}_A(\mathbf{Q}(0), S) = \bigcup_{\mathbf{D} \in \mathbb{R}_+^K} \{ \mathbf{D}_A(\mathbf{Q}(0), S) + \mathbf{D} \}.
\]

With a set of all possible scheduling policies \( S \), the optimal average delay region is defined as

\[
\mathcal{D}_A(\mathbf{Q}(0)) = \bigcup_{S \in S} \mathcal{D}_A(\mathbf{Q}(0), S).
\]

Any delay points in \( \mathcal{D}_T \) and \( \mathcal{D}_A \) are achievable by using some schedulers, and delay points outside \( \mathcal{D}_T \) and \( \mathcal{D}_A \) cannot be achieved by any schedulers. The optimal delay region characterizes a fundamental trade-off among each user’s queueing delay or QoS satisfaction. Thus, the appropriate operating delay point can be determined from this region so that each user’s different QoS requirement is met. In addition, the delay tuples at each time slot.

\[
S(\mathcal{C}(\mathbf{h}, P), \mathbf{Q}(t)) = [\mathbf{R}(0), \mathbf{R}(T_s), \mathbf{R}(2T_s) \ldots].
\]
performance of a scheduler can be evaluated by comparing the achievable delay region of that scheduler with the optimal delay region.

III. OPTIMAL DELAY REGION FOR TOTAL DELAY

In this section, the characteristics of the optimal delay region in terms of total delay are addressed. Also, optimization problems to characterize the optimal delay region are formulated, and efficient methods are presented to solve these problems.

A. QPS and Extended QPS Policy for DT

For any $\alpha \in \mathbb{R}_+^K$ and an initial queue state $Q(0)$, consider the following minimization problem.

$$\min_{S \in S} D_{T,1}(Q(0), S)$$

s.t. $\alpha_1 D_{T,1}(Q(0), S) = \alpha_k D_{T,k}(Q(0), S), k = 2, \ldots, K,$

where $\alpha_1 > 0$ is assumed without loss of generality. The constraints of (5) form a straight line starting from the origin in $D_T$ space. Thus, if a solution exists, the line intersects $D_T$, and the solution is on the boundary of $D_T$. By solving (5) for every $\alpha \in \mathbb{R}_+^K$, the boundary of $D_T$ can be completely characterized. The following theorem proves the delay optimality of queue proportional scheduling (QPS) when $D_{T,1} = D_{T,2}, \ldots = D_{T,K}$.

**Theorem 1:** If $\alpha_1 = \alpha_2, \ldots = \alpha_K$, QPS achieves a boundary point of $D_T$, which is the solution of (5).

**Proof:** At each scheduling period, QPS allocates $	ext{R}_{QPS}(t)$ such that

$$\text{R}_{QPS}(t) = D(t) \left(\frac{\max_{x \in \mathcal{C}} x}{Q(0)}\right).$$

Since $D_{QPS}(0) = Q(0)x_0$ and $D_{QPS}(T_s) = Q(T_s)x_1 = (Q(0) - T_s)^{-1}x_0$, $D_{QPS}(T_s) = (Q(0) - T_s)^{-1}x_0$, and $\text{R}_{QPS}(T_s) = \text{R}_{QPS}(0)$ thus, in the draining problem with no further arrivals, $D_{QPS}(t) = D_{QPS}(0)$ for all $t$. By the definition of QPS, $\text{R}_{QPS}(t) \in \text{bd}(\mathcal{C})$, i.e. on the boundary of $\mathcal{C}$. Hence, $D_{T,1} = D_{T,2}, \ldots = Q(0)/D_{QPS}(0) = I_t$, which means every queue backlog is simultaneously emptied at $t = I_t$. Assume there exist some other scheduling policies that can empty all the queue backlogs at $t \leq (l - 1)T_s$. With $\text{R}(iT_s) \in \mathcal{C}$ allocated during $iT_s \leq t < (i + 1)T_s$, $Q(0) = \sum_{i=0}^{l-2} \text{R}(iT_s)$. Since $Q(0) = I_t$, $D_{QPS}(0)$.

$$\text{R}_{QPS}(0) = \frac{1}{t} \sum_{i=0}^{l-2} \text{R}(iT_s)$$

Because of the convexity of $\mathcal{C}$, $\frac{1}{t} \sum_{i=0}^{l-2} \text{R}(iT_s) \in \text{C}$, and $\frac{1}{t} \sum_{i=0}^{l-2} \text{R}(iT_s) \in \text{int}(\mathcal{C})$, i.e. interior of $\mathcal{C}$, which is contradictory to $D_{QPS}(0) \in \text{bd}(\mathcal{C})$.

The above theorem shows that when every user has the same delay requirement, a scheduling policy $S = [\text{R}_{QPS}(0), \text{R}_{QPS}(0), \text{R}_{QPS}(0), \ldots]$ is an optimal solution of (5). In addition, QPS minimizes the time to drain all the queues as shown below.

**Theorem 2:** For an initial queue state $Q(0)$, QPS minimizes the time to empty all the queue backlogs. Thus, QPS provides an optimal solution to the following problem.

$$\min_{S \in S} \max_{s \in S} \{D_{T,1}, D_{T,2}, \ldots, D_{T,K}\}.$$  

**Proof:** This theorem is equivalent to Theorem 4 in [10], whose proof is as follows: Without loss of generality, assume $T_s = 1$. Then, let $T_X$ denote the time until a scheduling algorithm $X$ empties all the queue backlogs. Over the time interval $[0, T_X]$, the total supported data vector in bits is $Q(0)$. Thus, the average data vector allocated per each scheduling period is given by $\text{R}_{X,avg} = Q(0)/T_X$. The achievable rate region is a convex set since the time-sharing operation can be always performed. Thus, $\text{R}_{X,avg} \in \mathcal{C}$ is always satisfied, and $T_X$ is minimized by assigning $\text{R}_{opt} = (\max_r)Q(0)$ such that $rQ(0) \in \mathcal{C}$ for every scheduling period. Under the QPS policy, without new packet arrivals, the direction of the queue state vector is preserved over time since the scheduled rate vector is always proportional to the queue state vector. Therefore, by definition, QPS allocates a data rate vector $\text{R}_{QPS} = (\max_r)Q(0)$ such that $rQ(0) \in \mathcal{C}$ at each scheduling time. It can be easily seen that $\text{R}_{opt} = \text{R}_{QPS}$ and this completes the proof of the theorem.

When each user’s delay requirement is different, the following theorem proves that the number of distinct rate vectors required to achieve the boundary of the optimal delay region is equal to the number of distinct values in $\alpha$. For example, in Theorem 1, all the elements of $\alpha$ have the identical value; thus, only one rate vector is required, which is $\text{R}_{QPS}(0)$.

**Theorem 3:** Given $\alpha_1 > \alpha_2, \ldots, > \alpha_K$, denote an optimal delay of user $k$ by $D_{T,k}$ where $D_{T,1} < D_{T,2} < \ldots < D_{T,K}$. Then, $D_T = [D_{T,1}, \ldots, D_{T,K}]^T$ can be achieved by using a scheduling policy that assigns only one rate vector over each time interval of $D_{T,k} < t < D_{T,k+1}$. Hence, this optimal scheduling policy can be represented as $[\text{R}(0), \text{R}(D_{T,1}^*, \text{R}(D_{T,2}^*), \ldots, \text{R}(D_{T,k}^*)], which is called extended QPS.

**Proof:** Assume that a scheduling policy $S^*$ achieves $D_T$. Let $D_{T,0} = 0$, then, queue states at $t = D_{T,0}, D_{T,1}, \ldots, D_{T,K}$ under $S^*$ are, respectively, $Q^*(D_{T,0}), Q^*(D_{T,1}), \ldots, Q^*(D_{T,K}), Q^*(D_{T,K})$. Since $Q^*_1, \ldots, Q^*_k$ are empty at $D_{T,k}$ by definition of $S^*$, $Q^*(D_{T,k}) = [0, \ldots, Q_{k+1}(D_{T,k}), \ldots, Q_k(D_{T,k})]^T$. Let $\Delta Q(T_{k}) = Q^*(T_{k}) - Q^*(T_{k+1})$, i.e. $\Delta Q^*(T_{k})$ represents the amount of emptied backlogs during $D_{T,k} \leq t < D_{T,k+1}$. Then, consider a scheduling policy $S^* = [\text{R}(0), \text{R}(D_{T,1}^*), \text{R}(D_{T,2}^*), \ldots, \text{R}(D_{T,K}^*)]$ that achieves $D_T^* = \Delta Q^*(T_{k})$, where $\text{R}(D_{T,k}) = \Delta Q^*(T_{k})\max_{\Delta Q^*(D_{T,k}) \in \mathcal{C}}(x(t))$ is assigned over the time interval of $D_{T,k} \leq t < D_{T,k+1}$ to empty $\Delta Q^*(T_{k})$. By Theorem 2, QPS minimizes the time to empty $\Delta Q^*(T_{k})$, which leads to $D_{T,k+1} - D_{T,k} \leq D_{T,k+1} - D_{T,k}$. Thus, $D_T^* \leq D_T^*$, and by Definition 2, $D_T^* \in D_T(S^*)$. $S^*$ is equivalent to applying a series of QPS on each $\Delta Q^*(D_{T,k})$, which is called extended QPS.
Theorem 3 can be extended to the case when $\alpha_k = \alpha_{k+1} = \ldots = \alpha_{K+1}$. In this case, $K - m$ distinct rate vectors are sufficient to optimally solve (5) where $R(D_{T,k})$ is used for the scheduling period of $D_{T,k} \leq t < D_{T,k+1}$. If $\alpha_1 = \alpha_2 = \ldots = \alpha_K$, one rate vector can solve (5), which is $R_{QPS}(0)$ of Theorem 1.

B. Efficient Methods for Optimizing Extended QPS

From Theorem 3, solving (5) results in $K$ optimal rate vectors. Then, assuming $\alpha_1 > \alpha_2, \ldots, > \alpha_K$, (5) can be converted to the following optimization problem.

$$\min R(D_{T,k-1}) \quad D_{T,1}(Q(0), S)$$

s.t. $\alpha_1 D_{T,1}(Q(0), S) = \alpha_k D_{T,k}(Q(0), S)$,

$R(D_{T,k-1}) \in C, 1 \leq k \leq K,$

(8)

where $D_{T,0} = 0$. The difference between (5) and (8) is that the optimization is performed only for $K$ rate vectors. Since $R(D_{T,k-1})$ is used for the scheduling period of $D_{T,k-1} \leq t < D_{T,k}$, the following relations need to be satisfied and added to (8) as constraints.

$$D_{T,1} = \frac{Q_1(0)}{R_1(0)},$$

$$D_{T,2} = D_{T,1} + \frac{Q_2(0) - D_{T,1}R_2(0)}{R_2(D_{T,1})},$$

$$\vdots$$

$$D_{T,K} = D_{T,K-1} + \frac{Q_K(0) - \sum_{j=1}^{K-1}(D_{T,j} - D_{T,j-1})R_k(D_{T,j-1})}{R_K(D_{T,K-1})},$$

(9)

where $R_k(D_{T,j})$ is the service rate of queue $k$ for the scheduling period of $D_{T,j} \leq t < D_{T,j+1}$. The first equality of (9) comes from the fact that $Q_1(t)$ should be empty at $t = D_{T,1}$ with the service rate $R_1(t)$. For the second equality, $D_{T,2}$ is the sum of $D_{T,1}$ and the time to empty $Q_2(0) - D_{T,1}R_2(0)$ that is the remained backlog at time $D_{T,1}$ with the service rate $R_2(D_{T,1})$. The last equality can be interpreted in the same way. For the two user case, (8) can be converted to the following simple problem by removing $D_{T,1}$ and $D_{T,2}$ in the constraints.

$$\max R(0), R(D_{T,1})$$

s.t. $R_1(0) = \frac{Q_1(0)}{Q_2(0)} [R_2(0) + R_2(D_{T,1})(\alpha_1 - 1)],$

$R(0), R(D_{T,1}) \in C,$

(10)

where $\alpha_1 > \alpha_2 = 1$ is assumed. Since $Q_1(t) = 0$ for $t \geq D_{T,1}, R_1(D_{T,1}) = 0$ and $R_2(D_{T,1})$ can be found from the capacity region. Thus, the first constraint of (10) is a single line in $C$, and the optimal $R(0)$ is the point where the line meets with the boundary of $C$. Once $R(0)$ is found, $D_{T,1}$ and $D_{T,2}$ are obtained from (9). If $\alpha_1 = 1$, the constraint of (10) becomes $R_1(0) = Q_1(0) R_2(0)$, which is exactly the same with the QPS policy. However, the aforementioned method is not applicable to the case of $K > 2$. Instead, a bisection search combined with a convex-feasibility check can be used to efficiently determine the optimal $R(D_{T,k})$ as follows.

- **Initialize**: $l = 0, u = D_{max}, \delta > 0$
- **Repeat until** $u - l < \delta$:
  a. $D_{T,1} = (l + u)/2$
  b. Feasibility check of $D_{T,1}$ with (11)
  c. If $D_{T,1}$ is feasible, then set $u = D_{T,1}$; otherwise, set $l = D_{T,1}$

$D_{max}$ can be also determined by a feasibility check. First, choose any positive value $d$ and run a feasibility check of $d$. If $d$ is feasible, $D_{max}$ uses that value of $d$; otherwise, double the value of $d$ and repeat until $d$ is feasible. The above bisection is performed only for $D_{T,1}$ since $D_{T,k}(k \geq 2)$ are determined by $\alpha_1 D_{T,1} = \alpha_2 D_{T,2} = \ldots = \alpha_K D_{T,K}$. For the given $D_{T,k}$, a convex-feasibility check is formulated as the following.

$$\min R(D_{T,k-1})$$

s.t. $D_{T,k} = D_{T,k-1} + \frac{Q_k(0) - \sum_{j=1}^{k-1}(D_{T,j} - D_{T,j-1})R_k(D_{T,j-1})}{R_k(D_{T,k-1})},$

$R(D_{T,k-1}) \in C, 1 \leq k \leq K.$

(11)

Since $C$ is a convex set and equality constraints are linear in $R(D_{T,k-1})$, (11) can be solved by using general convex optimization tools [11]. For example, in a Gaussian broadcast
channel, by expressing $C$ in terms of rates, (11) can be formulated as geometric programming (GP), which is efficiently solvable [8].

Fig. 2 and Fig. 3 illustrate how to characterize the optimal delay region of two user systems. Fig. 2 shows the capacity region and the initial queue state vector $[Q_1(0), Q_2(0)]^T$. The rate supported by the QPS policy is the boundary point of the capacity region, which is proportional to the queue state vector. This point is the only rate vector supported by QPS until both queues are empty. On the other hand, to meet different delay requirements such as $D_{T,1} > D_{T,2}$, the slope of an initial rate vector should be larger than that of QPS. In this case, point (a) is supported first until Q2 becomes empty, and then point (b) is supported after Q2 becomes empty. The slope of point (a) is the same with $[Q_1(0) - Q_1(D_{T,2}), Q_2(0)]^T$ that is emptied during $D_{T,2}$. However, these rate vectors and queue states cannot be obtained directly with the constraint of $\alpha_1D_{T,1} = \alpha_2D_{T,2}$. The exact point should be determined by (10) for two user system and a bisection search with convex-feasibility check for more than two users. Fig. 3 illustrates the queue draining process. The queue-draining process of QPS is a straight line from the initial queue state $[Q_1(0), Q_2(0)]^T$ to the empty state $[0, 0]^T$, which is the reason why $D_{T,1} = D_{T,2}$ under the QPS regardless of the initial queue states and channel states. On the other hand, the extended QPS drains $Q_2$ faster than $Q_1$, which makes $D_{T,1} > D_{T,2}$. Since the extended QPS achieves the boundary of the optimal delay region as proved in Theorem 3, the optimal delay region can be characterized by sweeping the point $Q_1(D_{T,2})$ from $Q_1(0)$ to 0. Likewise, for the case of $D_{T,1} < D_{T,2}$, the algorithm sweeps the point $Q_2(D_{T,1})$ from $Q_2(0)$ to 0.

IV. NUMERICAL RESULTS

In this section, the optimal delay region for the total delay and the achievable delay region for the average delay are presented by simulation. Consider a two-user broadcast system with a quasi-static channel and no further packet arrivals with initial backlogs. The quarter-circle of Fig. 2 is assumed to be the channel capacity region. Let the initial queue state be $Q(0) = [x, 40]^T$ bits where $x = 20, 40, 60, 80$. Fig. 4 shows the boundaries of the optimal total delay region $D_T$ when $Q_1(0)$ takes four different values of 20, 40, 60, and 80. All the points where $D_{T,1} = D_{T,2}$ are achieved by QPS irrespective of the initial queue states, and other points are achieved by extended QPS from solving (10). As seen in the figure, the optimal delay region is not convex. Even though time-sharing of two scheduling policy is possible, resultant delay is not the linear sum of two delay points; thus, convexification of two delay points is not possible. Fig. 5 shows the achievable delay region in terms of the average delay for the same capacity region and initial queue states. This region is characterized by applying the extended QPS policy where its optimality for the average delay region is not covered in this paper.

V. CONCLUSION

This paper presents the optimal delay region for multi-user packet scheduling and develops optimal scheduling polices that achieve its boundary points. The optimal delay region demonstrates the fundamental trade-off among the queueing delays of all the users, which is useful in designing intelligent schedulers to satisfy heterogeneous QoS requirements. As capacity regions provide a crucial design guideline for communication systems, an optimal delay region can play a vital role in cross-layer resource allocation.

REFERENCES