Broadcast Traffic Queueing in OFDMA Systems for Spectrum Efficiency and Improved QoS Provisioning

Maduranga Liyanage
Keio University, Japan.

Iwao Sasase
Keio University, Japan.

Abstract—A queueing model is proposed for multiuser systems for increased spectrum efficiency and better QoS provisioning. Broadcast services are quickly finding their way into the IP domain and service providers need to introduce different traffic servicing models to handle these bandwidth intensive multimedia services. In an Orthogonal-frequency-division-multiple-access system, using conventional subcarrier per-user resource allocation method is inefficient. Here we present a two-stage queueing model where broadcast streaming traffic are aggregated by separate queues and processed in batches in the first-stage before propagating to the root queue. Modeling the general user traffic as an exogenous traffic source is also considered, which provides more delay performance improvement. The queueing framework is mathematically analyzed for average delay performances. Simulation results validate that not only resource consumption is reduced, but system queue delays also show substantial reductions. The introduced queueing model provides a framework for handling broadcast traffic and to reduce its impact on total system throughput.

I. INTRODUCTION

Orthogonal-frequency-division-multiple-access (OFDMA) has been chosen as the PHY/MAC technology for the next-generation broadband wireless communication systems. Recent explosion of smartphones has changed the traffic usage patterns of the existing networks that it has become difficult to provide the guaranteed quality-of-service (QoS) to the end users. Migration of traditional broadcast services like internet radio and TV into the IP domain has made things worse due to their high bandwidth consumption. It is our objective to address an efficient serving of broadcast services to end users of a multiuser system such as OFDMA, in a resource efficient manner.

OFDMA is capable of very high throughput at its’ maximum theoretical performance. But in practical deployments it is very difficult to achieve these performance levels. Therefore, resource allocation in OFDMA networks has been a deeply researched area. The optimal resource allocation is a NP-hard problem that is mathematically difficult to solve [1], and hence researchers are resolving to sub-optimal schemes. The main objectives of any resource allocation scheme must be: maximizing system throughput, minimizing transmit power and maintaining QoS across all users. Earlier resource allocation research worked mainly in the PHY of OFDMA [2]. But it was realized that resource allocation techniques need to consider both PHY and MAC of OFDMA to provide a sustainable solution to the stringent QoS requirements of recent services which are not only bandwidth sensitive but also delay sensitive.

A comprehensive analysis is done in [3] where utility functions are defined to bridge the PHY and MAC of OFDMA. In [4] authors try to find an unified methodology for radio link level queuing. A discrete Markov modulated Poisson process (dMMPP) model is used for each user. This is an important adoption since it gives the ability to individually control the data rate and delay constraints of the different applications. A similar queue level and radio link level model is proposed in [5] for users with heterogeneous delays by transforming delay constraints in to rate constraints. Queue delays are explored in [6] where a M/G/1 queue delay constraint of a user is related to an outage probability and then solved through Lagrangian.

With the migration of broadcast services into the IP domain, a different resource allocation method is needed to be implemented, since existing methods assume that each users’ data are independent, but in the case of broadcast services this is not the case. In this paper we consider the delivery of these traffic with shared subcarriers. We use a dMMPP model to have the flexibility to address the different QoS requirements of different services of each user. In the conventional cross-layer resource allocation schemes each users’ queue delay is determined individually disregarding the other users in the system. In our model we consider the traffic patterns of every user in the system when analyzing the system delay.

A two-stage queueing model is used. First-stage consists of streaming queues, which are the queues that hold the traffic of different broadcast services from all the users. A streaming queues’ packets are aggregated and processed as a batch and sent to the second-stage of the queue, the root queue, where they are allocated to subcarriers in the order of arrival. We determine the delays associated with each of the queues and calculate the batch sizes of the streaming queues so an optimal delay-resource efficiency can be reached. This queueing model lets us take in to account system traffic intensity when determining individual delays. Use of individual queues for each service gives the flexibility to assign priorities to queues if required, further increasing the QoS.

The rest of the paper is organized as follows. Section II discusses the system model and introduces the problem. The
A general traffic queueing and resource allocation scheme of an OFDMA system can be depicted as shown in Fig. 1. Each users’ traffic arrives at an individual queue and then propagates to a root queue where each packet is processed by a server and then sent for resource allocation. In this context we use the term resource allocation loosely to mean subcarrier, power, modulation/coding rate allocation, etc. This step is depicted in the figure as a black-box to indicate that the resource allocation algorithm can be any specific allocation scheme and that the nature of that algorithm does not impact the traffic queueing scheme in consideration.

A system with \( K \) users and \( N \) data carrying (i.e. excluding control/guard) subcarriers is considered. Furthermore, we assume that each users’ traffic can be categorized in to \( S \) types, with one general traffic type and \( S-1 \) streaming traffic types. Streaming traffics are broadcast streams we discussed in the introduction. Users who request data from these traffic sources receives the same stream of packets. In this type of traffic, if there are multiple users receiving this stream, the packets being transmitted to each of the users are similar and hence bandwidth can be saved if this traffic is transmitted on the subcarriers to each of the users. Each users’ traffic is considered as a discrete Markov process and can be represented as by a state-transition probability matrix for each user \( k, k = 0, 1, \ldots, K-1 \) as [7]

\[
P_k = \begin{bmatrix}
    P_{(k),0,0} & P_{(k),0,1} & \cdots & P_{(k),0,S-1} \\
    P_{(k),1,0} & P_{(k),1,1} & \cdots & P_{(k),1,S-1} \\
    \vdots & \vdots & \ddots & \vdots \\
    P_{(k),S-1,0} & P_{(k),S-1,1} & \cdots & P_{(k),S-1,S-1}
\end{bmatrix}
\]  

(1)

The matrix \( P_k \) is a \( S \)-state Markov state transition matrix showing each users’ traffic type changes probabilistically, given by:

\[
P_{(k),s_i,s_j} = \text{prob} [S[n] = s_j | [S[n-1] = s_i]].
\]  

(2)

Here \( S[n] \) is the traffic state at time instant \( n \), and hence, \( P_{(k),s_i,s_j} \) represents the conditional probability of traffic in state \( s_j \) given it was in state \( s_i \) in the previous time instant. Defining \( P_k \) lets one find the the state probabilities for each user. Using the Markov property, the probability of states of user \( k \) at time \( n \) can be written iteratively as follows:

\[
P_k^T[n] = P_k^T[n-1] P_k.
\]  

(3)

where \( P_k[n] \) is a \( S \times 1 \) vector with traffic probabilities at time \( n \). If we define the steady-state probability vector as \( \pi \), then as \( n \rightarrow \infty \),

\[
\pi^T_k = P_k^T[0] \pi^0_k.
\]  

(4)

Average performances of the system are analyzed in this study, and thus steady-state traffic probabilities \( \pi_i \) are considered in the throughput calculations. The \( i \)-th entry of \( \pi \) denotes the probability of traffic being in state \( i \). If we denote users \( k \)’s data arrival rates by matrix

\[
\lambda_k = \begin{bmatrix}
    \lambda_{k,0} & 0 & \cdots & 0 \\
    0 & \lambda_{k,1} & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & \lambda_{k,S-1}
\end{bmatrix},
\]  

(5)

with each entry on the diagonal indicating the traffic arrival rate for each state. If the cumulative traffic rate of user \( k \) is \( \lambda_{\text{cum},k} \), we then have

\[
\lambda_k = \text{diag}(\lambda_{\text{cum},k}\pi_k).
\]  

(6)

From the merging property of Poisson processes we have the total system traffic at the root queue to be a Poisson process with the rate \( \lambda_{\text{sys}} = \sum_{k=0}^{K-1} \lambda_{\text{cum},k} \).

A. Queueing delay - Conventional System

Queueing delay for the conventional system depicted in Fig. 1 is determined here. The average system queueing delays for a packet is [8]

\[
W_{\text{sys}}^{(1)} = \frac{\rho^2}{2\lambda_{\text{sys}}(1-\rho)},
\]  

(7)

and

\[
W_{\text{sys}}^{(1)} = W_{\text{que}}^{(1)} + \frac{1}{\mu_R},
\]  

(8)

Here \( W_{\text{que}}^{(1)} \) is the average delay in the queue, while \( W_{\text{sys}}^{(1)} \) is the average total system delay, i.e. including the processing delay. Queue is modelled as a M/D/1 queue with the server processing delay taken to be a deterministic value of \( \mu_R \) with the subscript \( R \) denoting the root queue. For steady-state operation we have that

\[
\mu_R > \sum_{k=0}^{K-1} \lambda_{\text{cum},k}.
\]  

(9)

The traffic intensity is defined by \( \rho = \frac{\lambda_{\text{sys}}}{\mu_R} \). We model the server delay as deterministic since the packets, specially in IP networks, are of fixed size and thus processing them takes a fixed amount of time.
III. TWO-STAGE QUEUING MODEL

In this subsection we introduce the proposed queueing system. As defined in (1), each users’ traffic is categorized in to S − 1 streaming traffic types with one general traffic type. The proposed queueing model to achieve this streaming traffic routing is shown in Fig. 2.

The system is modeled as a two-stage queue. In the first stage, each users packets are categorized to their corresponding traffic types and sent to streaming queues, S1, i = 1, 1,..., S− 1. It should be noted that general traffic packets should not go through a streaming queue, but to keep the analyses complete and simple we will assume that streaming queue S0 is assigned for general traffic of users. Each streaming queue is modeled as a M/D/1 queue with every streaming queue having the same deterministic processing rate of μR. The functionality of the streaming queues is to batch process the packets, i.e. a certain number (a batch) of packets is processed and sent as a single stream. The second-stage of the queue, the root queue, functions similar to the root queue of the conventional system (Fig. 1). Packets are processed and sent to the resource allocation system. As defined in (1), each users’ traffic is categorized in to S− 1 streaming traffic types with one general traffic type. The functionality of the streaming queues is to batch process the packets, i.e. a certain number (a batch) of packets is processed and sent as a single stream. The second-stage of the queue, the root queue, functions similar to the root queue of the conventional system (Fig. 1). Packets are processed and sent to the resource allocation system.

The system is modeled as a two-stage queue. In the first stage, each users packets are categorized to their corresponding traffic types and sent to streaming queues, S1, i = 1, 1,..., S− 1. It should be noted that general traffic packets should not go through a streaming queue, but to keep the analyses complete and simple we will assume that streaming queue S0 is assigned for general traffic of users. Each streaming queue is modeled as a M/D/1 queue with every streaming queue having the same deterministic processing rate of μR. The functionality of the streaming queues is to batch process the packets, i.e. a certain number (a batch) of packets is processed and sent as a single stream. The second-stage of the queue, the root queue, functions similar to the root queue of the conventional system (Fig. 1). Packets are processed and sent to the resource allocation system.

The system is modeled as a two-stage queue. In the first stage, each users packets are categorized to their corresponding traffic types and sent to streaming queues, S1, i = 1, 1,..., S− 1. It should be noted that general traffic packets should not go through a streaming queue, but to keep the analyses complete and simple we will assume that streaming queue S0 is assigned for general traffic of users. Each streaming queue is modeled as a M/D/1 queue with every streaming queue having the same deterministic processing rate of μR. The functionality of the streaming queues is to batch process the packets, i.e. a certain number (a batch) of packets is processed and sent as a single stream. The second-stage of the queue, the root queue, functions similar to the root queue of the conventional system (Fig. 1). Packets are processed and sent to the resource allocation system.

The system is modeled as a two-stage queue. In the first stage, each users packets are categorized to their corresponding traffic types and sent to streaming queues, S1, i = 1, 1,..., S− 1. It should be noted that general traffic packets should not go through a streaming queue, but to keep the analyses complete and simple we will assume that streaming queue S0 is assigned for general traffic of users. Each streaming queue is modeled as a M/D/1 queue with every streaming queue having the same deterministic processing rate of μR. The functionality of the streaming queues is to batch process the packets, i.e. a certain number (a batch) of packets is processed and sent as a single stream. The second-stage of the queue, the root queue, functions similar to the root queue of the conventional system (Fig. 1). Packets are processed and sent to the resource allocation system.

The system is modeled as a two-stage queue. In the first stage, each users packets are categorized to their corresponding traffic types and sent to streaming queues, S1, i = 1, 1,..., S− 1. It should be noted that general traffic packets should not go through a streaming queue, but to keep the analyses complete and simple we will assume that streaming queue S0 is assigned for general traffic of users. Each streaming queue is modeled as a M/D/1 queue with every streaming queue having the same deterministic processing rate of μR. The functionality of the streaming queues is to batch process the packets, i.e. a certain number (a batch) of packets is processed and sent as a single stream. The second-stage of the queue, the root queue, functions similar to the root queue of the conventional system (Fig. 1). Packets are processed and sent to the resource allocation system.

The system is modeled as a two-stage queue. In the first stage, each users packets are categorized to their corresponding traffic types and sent to streaming queues, S1, i = 1, 1,..., S− 1. It should be noted that general traffic packets should not go through a streaming queue, but to keep the analyses complete and simple we will assume that streaming queue S0 is assigned for general traffic of users. Each streaming queue is modeled as a M/D/1 queue with every streaming queue having the same deterministic processing rate of μR. The functionality of the streaming queues is to batch process the packets, i.e. a certain number (a batch) of packets is processed and sent as a single stream. The second-stage of the queue, the root queue, functions similar to the root queue of the conventional system (Fig. 1). Packets are processed and sent to the resource allocation system.

The system is modeled as a two-stage queue. In the first stage, each users packets are categorized to their corresponding traffic types and sent to streaming queues, S1, i = 1, 1,..., S− 1. It should be noted that general traffic packets should not go through a streaming queue, but to keep the analyses complete and simple we will assume that streaming queue S0 is assigned for general traffic of users. Each streaming queue is modeled as a M/D/1 queue with every streaming queue having the same deterministic processing rate of μR. The functionality of the streaming queues is to batch process the packets, i.e. a certain number (a batch) of packets is processed and sent as a single stream. The second-stage of the queue, the root queue, functions similar to the root queue of the conventional system (Fig. 1). Packets are processed and sent to the resource allocation system.
Here $A_{S.1}(t)$ equals the number of new packet arrivals to the streaming queue $S_i$ during time $t$ and $I(x)$ is the unit step function

$$I(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \end{cases}.$$  

(18)

Summing all equations in (16) and (17) and equating to (15), the state equation for the complete model is given as:

$$Q(t) = Q(t-1) + A(t-1) - I(Q(t-1)),$$

(19)

where $A(t) = A_{S.1}(t) + A_{S.2}(t) + \ldots + A_{S.S-1}(t)$. Comparing with (16) we see that (19) is the state equation for a M/D/1 queue. Since $Q_R = Q - Q_{S.1} - Q_{S.2} - \ldots - Q_{S.S-1}$, and using the fact that a queue size of a M/D/1 queue is given by:

$$Q_R/D_R = \frac{\lambda^2}{2\mu_1 - \lambda},$$

(20)

the queue size of the root queue can then be determined as:

$$Q_R = \frac{\sum_{i=0}^{S-1} \left( \lambda^{\text{eff}}_{(a),i} \right)^2}{2\mu R} \left( \mu R - \sum_{i=1}^{S-1} \lambda^{\text{eff}}_{(a),i} \right) - \sum_{i=0}^{S-1} \frac{\left( \lambda^{\text{eff}}_{(a),i} \right)^2}{2\mu R} \left( \mu R - \lambda^{\text{eff}}_{(a),i} \right),$$

(21)

where $\lambda^{\text{eff}}_{(a),i}$ is the effective traffic rate of the $S_i$ as seen by the root queue in the second stage, defined as

$$\lambda^{\text{eff}}_{(a),i} = \frac{\lambda_{(a),i}}{B_i}, \quad i = 0, 1, \ldots, S - 1.$$  

(22)

This is because the traffic entering the root queue is a $B_i$ fraction of the actual traffic in queue $C_i$. Using Little’s formula, the delay at the root queue is then simply found to be

$$W_{3,\text{que}} = \frac{\sum_{i=0}^{S-1} \lambda^{\text{eff}}_{(a),i}}{2\mu R} \left( \mu R - \sum_{i=0}^{S-1} \lambda^{\text{eff}}_{(a),i} \right) - \sum_{i=0}^{S-1} \frac{\left( \lambda^{\text{eff}}_{(a),i} \right)^2}{2\mu R} \left( \mu R - \lambda^{\text{eff}}_{(a),i} \right).$$

(23)

Thus the total delay at the root queue including the server delay is

$$W_3 = W_{3,\text{que}} + \frac{1}{\mu_R}.$$  

(24)

2) Two-Stage Model with Exogenous General Traffic: In the queueing system above we treated the general traffic as streaming traffic for simplicity and routed them through a streaming queue, $S_0$. But general traffic cannot be batch processed as we discussed in the beginning of section II. We can avoid this unnecessary processing by routing the general traffic as an exogenous traffic stream in to the root queue directly. Using priority based approach of [9], this queueing structure can be seen as a special case of a two-stage model, where the traffic from the streaming queues $S_i$, $i = 1, \ldots, S - 1$ as a single in route queue with aggregated traffic rate $\lambda_S = \sum_{i=1}^{S-1} \lambda^{\text{eff}}_{(a),i}$, and the other queue as the general traffic queue. Assuming the exogenous traffic is of low-priority, i.e. the exogenous traffic will be processed when there are no packets coming from the in-route queue, the average queue size of the exogenous traffic is shown to be [9]

$$Q_{\text{EXO}} = \frac{(\lambda_0 + \lambda_{(a),0})^2}{2\mu_R (\mu_R - \lambda_0 - \lambda_{(a),0})} - \frac{\lambda_0^2}{2\mu_R (\mu_R - \lambda_0)}.$$  

(25)

Note here that $\lambda_{(c),0}$ is the the aggregated general traffic rate from (11). The priority based approach is for analytical purposes only and need not be implemented for the above formula to hold true. Then the total traffic at the root queue is the sum of the exogenous queue size and the in-route queue size, which was calculated in (21). The root queue size in the exogenous input model is then

$$Q_{R,\text{EXO}} = \frac{(\lambda_0 + \lambda_{(a),0})^2}{2\mu_R (\mu_R - \lambda_0 - \lambda_{(a),0})} - \sum_{i=1}^{S-1} \frac{\left( \lambda^{\text{eff}}_{(a),i} \right)^2}{2\mu_R (\mu_R - \lambda^{\text{eff}}_{(a),i})}.$$  

(26)

The root queue delay for the exogenous traffic is then simply given from Little’s formula as

$$W_{3,\text{EXO}} = \frac{Q_{\text{EXO}}}{\lambda_S + \lambda_{(a),0}}.$$  

(27)

Note that in this case the only delay incurred on the exogenous traffic is the delay at the root queue $W_{3,\text{EXO}}$ and the processing delay. For the streaming traffic, the total delay is $W_{1,i} + W_{2,i} + W_{3,\text{EXO}}$. The queueing delay of the general traffic can be significantly reduced by this approach.

3) Optimum Queuing Parameters: It is necessary to ensure that this proposed model does not introduce a performance degradation in terms of delay. This section determines the optimum system parameters are to increase the performance without degrading delay performance. The cumulative delay in the two-stage model is $W_{\text{two}}(2) = W_{1,i} + W_{2,i} + W_{3,\text{EXO}}$ for each corresponding streaming queue. The only parameter adjustable is the batch size $B_i$ of each queue. This parameter defines how much transmission units can be reduced, hence increasing spectrum efficiency, but in the process also increases the delay. Parameter $B_i$ also controls delay $W_3$, but it is apparent that increasing $B_i$ reduces $W_3$, i.e. higher values of batch sizes reduces the traffic at the root queue. Therefore, our objective is to find an optimum $B_i$.

This problem has two objectives and one optimization parameter vector $B = [B_1, B_2, \ldots, B_S]$. This is a multi-objective optimization problem with a convex Pareto front. To make this optimization tractable we will make it an inequality-constrained maximization problem by transforming an objective in to a constraint. We make the objective of minimization of the delay a constraint, by realizing a reference a metric. For this reference delay we take the conventional
We further simplify the problem by taking

\[ \sigma \] 

If \( \lambda \) case \( \lambda \) solution \( \sigma \) where the first is the stationarity condition while the later two

\[ \sum_{i=1}^{S} B_i \] 

Then the Lagrangian is,

\[ L(B, \lambda, \sigma) = \sum_{i=1}^{S} B_i + \sum_{i=1}^{S} \lambda_i (W_{sys}^{(1)} - W_{1,i} - W_{2,i}) + \sum_{i=1}^{S} \sigma_i (B_i - 1), \] 

where \( \lambda \) and \( \sigma \) are the vectors of Lagrangian multipliers. The KT conditions are then,

\[ \frac{\partial L}{\partial B_i} (B, \lambda, \sigma) = 1 - \lambda_i \frac{\alpha_i}{2\lambda_i} + \sigma_i = 0, \] 

\[ \frac{\partial L}{\partial \lambda_i} (B, \lambda, \sigma) = \lambda_i (W_{sys}^{(1)} - W_{1,i} - W_{2,i}) = 0, \] 

\[ \frac{\partial L}{\partial \sigma_i} (B, \lambda, \sigma) = \sigma_i (B_i - 1) = 0. \] 

where the first is the stationarity condition while the later two are the complementary slackness conditions. From (32), if \( \sigma_i = 0 \), then from (30) \( \lambda_i = \frac{2\lambda_i}{\alpha_i} > 0 \), which gives the solution

\[ \left( W_{sys}^{(1)} - W_{1,i} - W_{2,i} \right) = 0 \) \) from (31). The \( f(B) \) maximizing solution is then,

\[ B_i^* = \frac{2\lambda_i}{\alpha_i} \left( W_{sys}^{(1)} - W_{1,i} \right). \] 

If \( \sigma_i > 0 \), (31) gives \( \lambda_i \left( W_{sys}^{(1)} - W_{1,i} - \frac{\alpha_i}{2\lambda_i} \right) = 0 \). The case \( \lambda_i = 0 \) leads to \( \sigma_i = -1(< 0) \), which is false, or else

\[ \left( W_{sys}^{(1)} - W_{1,i} - \frac{\alpha_i}{2\lambda_i} \right) = 0. \] 

Therefore, we can finally write the solution of \( B_i \) that maximizes \( f(B) \) as follows:

\[ B_i^* = \begin{cases} \frac{2\lambda_i}{\alpha_i} \left( W_{sys}^{(1)} - \frac{\lambda_i}{2\alpha_i} \right) - \frac{2}{j} & \text{Cond.-1} \\ \frac{2\lambda_i}{\alpha_i} \left( W_{sys}^{(1)} - \frac{\alpha_i}{2\lambda_i} \right) & \text{Cond.-2} \end{cases} \] 

where Cond-1 is \( W_{sys}^{(1)} > W_{1,i} + \frac{\alpha_i}{2\lambda_i} \), and Cond.-2 is the case otherwise.

IV. SIMULATION RESULTS AND DISCUSSION

This section evaluates the performance of the two-stage queueing model in terms of the average queue delay, transmission power and spectrum efficiency. Fig. 3 shows the queuing delays for the conventional single queueing system and the proposed two-stage queueing system, for both in-route and exogenous general traffic. The delay for the streaming queues are similar for both the types of two-stage queues, and for the single-queue system all traffic experience same delay. \( \mu_R \) is kept constant throughout the simulation.

The delay showed in the figure is for the general traffic, for two values of batch sizes, 2 and 4, as the number of users are increased. Two-stage queue with exogenous general traffic gives the lowest delay, with the single-queue system also giving lower delays for small number of users. The two-stage queue with in-route general traffic incur a slightly higher delay than the exogenous case. This is to be expected since the exogenous traffic by-passes the first stage of the queue. For example, in the exogenous case, when the number of users is 40, having a batch size of 4 yields a 6.5% increase in delay relative to the batch size of 2, but when the number of users are doubled to 80, B=4 shows a reduced delay of 7%.

Transmission power and the spectrum efficiency of the two systems as the batch size is increased are shown in Fig. 4. Transmission of a single transmission unit to a batch of \( B \) users is determined and only long-term path loss is considered using Frii’s formula. A maximum cell range of 2km is used in the simulation. For the conventional scheme, we simulated two methods, adaptive modulation and single modulation level transmission. In adaptive modulation we assumed the transmission range is divided in to 3 zones, where each zone uses 4, 16 and 64- QAM modulation levels, and the single modulation with 4-QAM only. Plot on the top shows the transmission power while the bottom plot shows the corresponding transmission units required. Use of adaptive modulation required the highest transmission power followed by the single modulation scheme. Batch transmission scheme of the two-stage queue model requires the lowest power as expected. For the single modulation level, the required average transmission power doubles when the batch size is increased from 3 to 6, while a further 50% is required when it is 9. For the batch transmission scheme of the two-stage queue model, the transmission power is only increased by 50% when the batch size is increased to 6 from 3, while only 20% increment is required for the batch size of 9. In terms of spectrum efficiency, single modulation scheme requires the most number of transmission units, one for each user. Adaptive modulation is more spectrum efficient as it can transmit more than a single transmission unit using higher level modulations. No power restrictions are imposed in this simulation. These simulation results show that the batch processing of streaming data using the two-stage model can, first, reduce the delay for general packets through the system queues, providing better QoS, and secondly, substantially reduce the transmission power and the resource utilization.
It is understood that the proposed two-stage queueing model needs assistance from the upper layers of the stack to be implemented. Different streaming services need to flag their packets in some form to efficiently differentiate them and to be routed in to streaming queues. This can be implemented fairly simply in the header of the packets. This model is not perfect in the sense that a batch of packets arriving will be of the same user or packets can arrive out of sequence. Also it is intuitive that optimal values $B^*_i$ cannot be used always and it depends also on the number of active users in queue $S_i$, because the packets of a batch could be of different sequence numbers. If $K_{S_i} < B^*_i$, then the nominal batch size should be $B^* = K_{S_i}$, otherwise if $K_{S_i} > B^*_i$, batch size should be the largest integer less than $B^*_i$ that completely divides $K_{S_i}$. In these situations the queue needs to adopt a different strategy. For example, the queueing system can choose to send particular users streaming data to the general queue or if packets arrive out of sequence the queue can simply send it as an unicast or store and process as a batch as other packets arrive. Of course in these cases it is very difficult to derive a definite formulation for the queueing model. What we have presented here is a general framework that can be incorporated to treat streaming broadcast services. A model such as this is needed to service broadcast services due to their higher bandwidth and the growing demand. The described model can be very efficient, for example in a 'live' streaming of a hugely popular event where there will be a significant number of users tuning in to watch it. This could take the bandwidth demand well over the usual levels and could degrade the QoS for every user in the system. A model like this could fit very well in to such situations.

V. CONCLUSION

A two-stage queueing model is analyzed for the servicing of streaming services. The presented method aggregates and batch processes streaming traffic, thus reducing the resource consumption and transmission power. Moreover, this frame-