A New Service Classification Strategy in Hybrid Scheduling to Support Differentiated QoS in Wireless Data Networks

Navrati Saxena, Kalyan Basu, Sajal K. Das
Comp. Sc. & Engg. Dept.
University of Texas at Arlington
Arlington, Texas, USA
{nsaxena, basu,das}@cse.uta.edu

Cristina M. Pinotti
Maths and Informatics Dept.
University of Perugia
Perugia, Italy
pinotti@unipg.it

Abstract

The wireless telecommunication industry is now slowly shifting the paradigm from circuit-switched voice-alone applications to a new audio-visual world. Diversification of personal communication systems (PCS) and gradual penetration of wireless Internet have generated the need for *differentiated services*. The set of clients (customers) in the wireless PCS networks is generally classified into different categories based on their power and importance. Activities of the customers having higher importance have significant impact on the system and the service providers. The goal of the service providers lies in minimizing the cost associated in the maintenance of the system and reducing the loss incurred from the clients' churn rate. Deployment of such differentiated services calls for efficient scheduling and data transmission strategies. In this paper we have developed a new service classification strategy in hybrid scheduling scheme to support differentiated quality of service (QoS) among the different set of clients. The scheme dynamically computes the data access probabilities and amalgamates the push and pull scheduling schemes to develop the hybrid scheduling framework. While a flat scheduling is used for push system, the major novelty of the work lies in differentiating the clients based on their priority-classes and incorporating the effect of priority in selecting an item from the pull-system. Modeling and analysis of the system is performed to get an average behavior of the QoS parameters like delay in our hybrid scheduling framework. Simulation results points out that the average waiting time for the highest priority clients can be kept very low, while simultaneously minimizing the number of requests dropped by assigning appropriate fraction of available bandwidth. It also demonstrates that by intelligent selection of the cut-off point, used to segregate push and pull systems, the overall cost associated with the system can be minimized.

1 Introduction

Historically, cellular telephone networks were the first radio access networks to be developed and widely deployed. The major objective behind the initial deployment of cellular wireless networks was to provide only an un-interrupted, circuit-switched voice communication. Increasing popularity of hand-held mobile devices, deregulation of wireless services and existence of multiple network operators introduced an era of *competitive* wireless market. This aids in rapid deployment of enhanced wireless communication technologies, thus improving the customer's satisfaction. The gradual deployment of the Internet applications added a new paradigm by introducing the concept of *data services* and *packet technologies*. While the growth in wireless voice communication is almost attaining its saturation, the primary target and attention of competitive service providers is slowly shifting towards the packet-switched, data services in cellular, wireless networks. The popularity of short messaging services (SMS), I-mode (Japan) and *push-to-talk* services over the legacy cellular systems along with the proliferation of IETF (Internet Engineering Task Force) standardized protocols and the speculation behind the deployment of UMTS (Universal Mobile Telecommunications System) is probably the first step of this movement towards wireless data services.

However the most challenging question at this point becomes "what is the level of Quality of Service (QoS) guarantee the wireless systems can provide for these newly introduced data services ?". Recent researches in QoS [8, 11] reveal that the resource (bandwidth) constraints, high bit-error rate, channel fading and interference in wireless channels, along with the hand-off generated from the user-mobility are the major impairments behind meeting the QoS guarantee needed for real-time, wireless services. The inherent asymmetry in wireless systems arising from difference in uplink and downlink channel capacity, number of clients and server, and uplink and downlink message-size makes the problem even more complex and challenging. Hence, in order to meet these stringent QoS requirements of wireless data services, one needs an efficient and scalable data broadcasting and scheduling strategy. Current cellular systems and its data transmission strategies do not differentiate the QoS among the clients, i.e., the sharing and management of resources does not reflect the importance of the clients. A close look into the existing hybrid scheduling strategy for wireless systems reveals that most of the scheduling algorithms aims at minimizing the overall average access time of all the clients. However, we argue that this is not sufficient for future generation cellular wireless systems which will be provid-

Proceedings of the 2005 International Conference on Parallel Processing (ICPP'05)
0190-3918/05 $20.00 © 2005 IEEE

ing QoS differentiation schemes. The items requested by clients having higher priorities might need to be transmitted in a fast and efficient manner, even if the item has accumulated less number of pending requests. Hence, if a scheduling considers only popularity, the requests of many important (premier) clients' may remain unsatisfied, thereby resulting in dissatisfaction of such clients. As the dissatisfaction crosses the tolerance limit, the clients might switch the service provider. In the anatomy of today's competitive cellular market this is often termed as *churning*. This churning has adverse impacts on the wireless service providers. The more important the client is, the more adverse is the corresponding effect of churning. The data transmission and scheduling strategy for cellular wireless data networks thus needs to consider not only the probability of data items, but also the priorities of the clients.

In this paper, we propose a new service classification strategy for hybrid broadcasting to support the differentiated QoS in wireless data networks. The hybrid scheduling that effectively combines broadcasting of more popular (i.e., push) data and disseminating (upon-request) the less popular (i.e., pull data) in asymmetric (where asymmetry arises because the number of clients is more than the number of servers), heterogeneous (different items have different lengths) environments. At any instance of time, the item to be broadcast is selected by applying a *flat* scheduling. However, the selection strategy for a pull-item is significantly influenced by the influence of the clients and the corresponding service classification scheme. The major novelty of our work lies in separating the clients into different classes and introducing the concept of a new selection criteria, termed as *importance factor*, by combining the clients' priority and the *stretch* (i.e, *max-request min-service-time*) value. The item having the maximum importance factor is selected from the pull queue. We argue that is a more practical and better measure in the system where different clients have different priorities and the items are of variable lengths. The performance of our heterogeneous hybrid scheduler is analyzed using suitable priority queues to derive the expected waiting time. The bandwidth of the wireless channels is distributed among the client-classes to minimize the request-blocking of highest priority clients. The cut-off point, used to segregate the push and pull items is efficiently chosen such that the overall costs associated in the system gets minimized. We argue that the strict guarantee of differentiated QoS, offered by our system, generates client-satisfaction, thereby reducing their churn-rate.

The rest of the paper is organized as follows. Section 2 reviews existing work in the literature of data broadcasting and scheduling. The new hybrid algorithm, which introduces the concept of service classification into hybrid scheduling to provide differentiated QoS is produced in Section 3. A suitable performance model, based on priority queuing, is developed in Section 4 to analyze and estimate the average delay and blocking in the hybrid system. Simulation results in Section 5 supports the performance analysis and points out that the resultant delay and blocking of the highest priority clients can be kept sufficiently low, thereby reducing the overall cost of the system. Finally, Section 6 concludes the paper.

2 Existing Works

A survey into the existing literature reveals the existence of a wide variety of solutions for data broadcasting. However, the solutions can be broadly categorized into two parts: (1) push-based broadcasting and (2) pull-based dissemination. While the *flat, round-robin* scheduling provides the most simple and basic transmission strategy, it always suffers from a fixed average delay, which is half of the sum of the length of all the data items. The work of Acharya and Franklin [1] is perhaps the first attempt to remove this disadvantage by introducing the role of access probability (popularity) in the selection of data items. This problem, better known as the *Broadcast Disk Problem*, groups data items in disks, thus assigning items in the same range of access probabilities to the same disk. The broadcast schedule is then generated by interleaving one item from each disk. The disks having higher access probabilities are of smaller size and have higher rotation speed. This results in the disks having higher access probabilities providing more instances of their data to the broadcast schedule. The Square-Root-Rule (SRR) [5] provides an optimal solution for the uniform-length broadcasting problem. This produces a broadcast schedule, where each data item appears with equally spaced replicas, having frequency directly proportional to the square root of its access probability and inversely proportional to the square root of its length.

A hybrid approach that use the flavors of both push-based and the pull-based scheduling algorithms in one system, appears to be more attractive. Perhaps the first hybrid technique for scheduling and data transmission in asymmetric environment is proposed in [2]. In this work, the server pushes all the data items according to some push-based scheduling, but simultaneously the clients are provided with a limited back-channel capacity to make requests for the items. In our previous work [10], we have developed a hybrid scheduling strategy for transmission of heterogeneous, variable length data items.

A close look into the existing hybrid scheduling strategy reveals that none of the existing works have considered service differentiation and client priorities into account. According to our knowledge, we are the first to develop the new service classification scheme in hybrid scheduling strategy, which is capable of offering differentiated QoS for wireless data networks.

Proceedings of the 2005 International Conference on Parallel Processing (ICPP'05)
0190-3918/05 $20.00 © 2005 **IEEE**

3 Hybrid Scheduling with Service Classification

We assume an environment with a single server serving multiple clients, thus imposing asymmetry. The server-database consists of a total D distinct items, out of which K items are pushed and the remaining $(D-K)$ items are pulled. All the items have variable lengths. The access probability P_i, of an item i is governed by the Zipf's distribution. Every client is also associated with certain priority. These priorities provides the influence and importance of the clients to the service providers. The push-based broadcasting ignores the clients' requests, and uses a *Flat* round-robin scheduling strategy for cyclic broadcasting of popular data items.

The pull-scheduling, on the other hand, is based on a linear combination of the number of clients' requests accumulated and priorities. It should be noted that items with pending requests for higher priority clients should be serviced faster than the items having requests from lower priority clients. However, this scheme might suffer from *un-fairness* to the lower priority clients and also does not consider the number of clients' requests. A data item, requested by many clients having lower importance, might remain in the pull queue for a long time. Eventually, all the pending requests for that item might be lost (blocked). Hence, a better option is to consider both the number of pending requests and the priorities of all clients requesting the particular data item. A close look into the system reveals that, the service time required to serve an item is dependent on the size of that item. The larger the length of an item the higher is its service time. We introduce a new scheduling strategy that combines *stretch optimal or max-request min-service-time first* schedule with the *priority scheduling* to select an item from the pull-queue. Formally if, $\mathcal{S}_i$ represents the stretch associated with item i and $\mathcal{Q}_i$ represents the total clients' priority associated with item i, then the item selected from the pull-queue is determined by the following condition:

$$\gamma_i = \max\left[\alpha\mathcal{S}_i + (1-\alpha)\mathcal{Q}_i\right], \qquad (1)$$

where α is a fraction $0 \leq \alpha \leq 1$, which determines the relative weights between the priority and the stretch value. Clearly, $\alpha = 0$ and $\alpha = 1$ makes the schedule priority-scheduling and stretch-optimal scheduling respectively.

When a client needs an item i, it requests the server for item i and waits until it listens for i on the channel. Note that the behavior of the client is independent of the fact that the requested item belongs to the push-set or the pull-set. Depending on the priorities, the server first classifies the clients into different service classes. The server goes on accumulating the set of requests from the clients. The algorithm starts with a fixed cutoff-point which separates the push and pull set. For any item arrived, it first determines if the item belongs to the push or the pull set. If the request is for a push item, the server simply ignores the request as the item will be pushed according to the online Flat, round-robin algorithm. However, if the request is for a pull item, the server inserts it into the pull queue with the arrival time, and updates its stretch value and total priority of all the clients' requesting that item. After every push, if the pull queue is not empty, the server chooses the item having maximum importance factor (γ_i) from the pull-queue. The bandwidth required by the data item is assumed to follow Poisson's distribution. If the required bandwidth of the data item is less than the bandwidth available for the corresponding service class, then the data item and the corresponding requests are lost. Otherwise, the server assigns the required bandwidth and transmits the item. Once the transmission is complete, the pending requests for that item in the pull-queue is cleared and the bandwidth used is released to update the available bandwidth. Figure 1 provides the pseudo-code of the hybrid scheduling algorithm executing at the server-side. Periodically the algorithm is executed for different cutoff-points and obtains the optimal cutoff-point which minimizes the overall access time (delay).

```
Procedure HYBRID SCHEDULING;
divide the clients among different service-classes;
while true do
      begin
      consider the access/requests arriving;
      ignore the requests for push item;
      append the requests for the pull item in the pull-
      queue with its arrival time and importance-factor;
      take out an item from push part and broadcast it;
      if the pull-queue is not empty then
        extract the item having maximum importance-factor
        (γ_i) from the pull-queue;
        clear the number of pending requests for that item;
        free the amount of required bandwidth and update
        the amount of available bandwidth;
      end-if
end-while
```

Figure 1: Service Classification in Hybrid Scheduling

4 Delay and Blocking in Differentiated QoS

In this section we study the performance evaluation of our hybrid scheduler system by developing suitable models to analyze its behavior. The prime concern of this analysis is to obtain an estimate of the minimum expected waiting time (delay) of the hybrid system. Since, this waiting time is dependent on the cutoff point K, investigation into the delay dynamics with different values of K is necessary to get the optimal cutoff point. As explained before in Section 3, the selection criteria in the pull system is dependent on both the stretch-value associated with the item and the priority of the clients requesting that particular

Proceedings of the 2005 International Conference on Parallel Processing (ICPP'05)
0190-3918/05 $20.00 © 2005 **IEEE**

item. Hence, the performance analysis also needs to consider the clients priority along with the stretch-value associated with every data item. We divide the entire analysis into two parts. In the first part, we consider the system without any role of the client's priority and obtain the expression for average number of items present in the system. In the second part, we introduce the explicit role of priorities in determining the average system performance.

4.1 Average Number of Elements in the System

Assumptions: The arrival rate in the entire system is assumed to obey the Poisson's distribution with mean λ'. The service times of both the push and pull systems are exponentially distributed with mean μ_1 and μ_2, respectively. Let C, D and K respectively represents maximum number of clients, total number of distinct data items and the cut-off point. The server pushes K items and clients pull the rest $(D-K)$ items. Thus, the arrival rate in the pull-system is given by: $\lambda = \sum_{i=K+1}^{D} \mathcal{P}_i \times \lambda'$, where $\mathcal{P}_i$ denotes the access probability of item i. We have assumed that the access probabilities P_i follow the *Zipf's distribution* with *access skew-coefficient* θ, such that $\mathcal{P}_i = \frac{(1/i)^\theta}{\sum_{j=1}^{n}(1/j)^\theta}$.

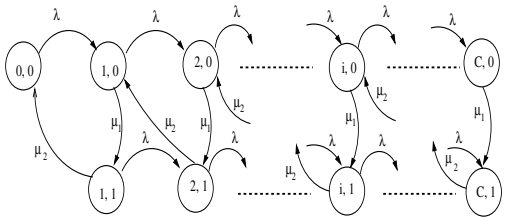

Figure 2: Performance Modelling of Hybrid System

Figure 2 illustrates the birth and death model of our system, where the arrival rate in the pull-system is given by λ. Any state of the overall system is represented by the tuple (i,j), where i represents the number of items in the pull-system and $j = 0$ (or 1) respectively represents whether the push-system (or pull-system) is being served. The arrival of a data item in the pull-system, results in the transition from state (i,j) to state $(i+1,j), \forall i \in [0,C]$ and $\forall j \in [0,1]$. The service of an item in the push system results in transition of the system from state $(i, j=0)$ to state $(i, j=1), \forall i \in [0,C]$. On the other hand, the service of an item in the pull results in transition of the system from state $(i, j=1)$ to the state $(i-1, j=0), \forall i \in [1,C]$. The details of steady-state flow balance equations and their solutions are explained in our previous work [10]. For the sake of clarity, we briefly highlight the major steps here. The steady-state behavior of the system (without considering priority) is represented by the equations given below:

$$p(0,0)\ \lambda = p(1,1)\ \mu_2$$

$$p(i,0)(\lambda+\mu_1) = p(i-1,0)\lambda + p(i+1,1)\mu_2 \quad (2)$$

$$p(i,1)(\lambda+\mu_2) = p(i,0)\mu_1 + p(i-1,1)\lambda \quad (3)$$

where $p(i,j)$ represents the probability of state (i,j). Dividing both sides of Equation (2) by μ_2, letting $\rho = \frac{\lambda}{\mu_2}$, $f = \frac{\mu_1}{\mu_2}$, performing subsequent z-transform and using Equation (2), we get

$$\begin{aligned} P_2(z) &= \rho\, p(0,0) + z(\rho+f)[P_1(z) - p(0,0)] - \rho z^2 P_1(z) \\ P_2(z) &= \frac{f[P_1(z) - p(0,0)]}{(1+\rho-\rho z)} \end{aligned} \quad (4)$$

Now, estimating the system behavior at the initial condition, we can state that the occupancy of pull and push states is given by: $P_2(1) = \sum_{i=1}^{C} p(i,1) = \rho$ and $P_1(1) = \sum_{i=1}^{C} p(i,0) = (1-\rho)$. Using these two relations in Equation (4), we can obtain the idle probability, $p(0,0)$ as: $p(0,0) = 1 - \rho - \frac{\rho}{f}$. Differentiating both sides of Equation (4) with respect to z at $z=1$, we estimate the expected number of elements in the pull-system ($E[\mathcal{L}_{pull}]$) as follows:

$$\begin{aligned} \left[\frac{\partial P_2(z)}{\partial z}\right]_{z=1} = E[\mathcal{L}_{pull}] &= (\rho+f)\mathcal{N} + (1-\rho) - \\ &\quad (\rho+f)\times(1-\rho-\frac{\rho}{f}) - \rho\mathcal{N} \end{aligned} \quad (5)$$

where $\left[\frac{\partial P_1(z)}{\partial z}\right]_{z=1} = \mathcal{N}$ represents the average number of elements in the pull queue when a push request is being serviced.

4.2 Priority-based Service Classification

Every client j is associated with a certain priority q_j, which reveals the importance or class of that client. Obviously, this influences the arrival rate associated with every item. The arrival rate associated with i^{th} item for j^{th} priority-client is given by: $\lambda_i = \lambda\, p_i\, q_j$. Now, L_i and R_i represents the length and number of pending requests associated with the i^{th} item, then the stretch-value S_i associated with that item is given by the expression: $S_i = \frac{R_i}{L_i^2}$. If $E[\mathcal{L}_{pull}]$ represents the average length of the pull queue, then average number of i^{th} items present in the queue is given by $E[\mathcal{L}_{pull}]p_i$. Hence, average importance of i^{th} item requested by j^{th} client is given by: $E[\mathcal{L}_{pull}]\, p_i\, q_j$. Representing the influence of the set of clients $\mathcal{S}$ requesting for item i by $\mathcal{Q}_i = \sum_{j=1}^{\mathcal{S}} q_j$, the selection criteria of that element is now given by the following equation:

$$\varrho_i = \left(\alpha \frac{E[\mathcal{L}_{pull}]p_i}{L_i^2} + (1-\alpha)E[\mathcal{L}_{pull}]\, p_i\, \mathcal{Q}_i\right) \quad (6)$$

Proceedings of the 2005 International Conference on Parallel Processing (ICPP'05)
0190-3918/05 $20.00 © 2005 **IEEE**

It should be noted that the above equation actually resembles Equation 1. However, Equation 1 does not consider the number of i^{th} items present in the pull queue. Thus, Equation 6 actually generalizes Equation 1 and boils down to Equation 1, when $E[\mathcal{L}_{pull}]p_i = 1$. This condition provides the position of every item in the priority queue. In order to distinguish this measure with the client priority q_j, we term ϱ_i as the *importance-factor* of item i. We first analyze the system performance with clients belonging to two different classes [4], having two different importance factors. Subsequently, we extend the framework to incorporate clients having multiple importance factors.

4.2.1 Delay Estimation for Two Different Service Classes

Let, λ_1 and λ_2 represents the average arrival rate of the data items having importance factors 1 and 2, i.e., $\lambda = \lambda_1 + \lambda_2$. We also assume that the most important items have the right to get service before the second important item without *preemption*. Now, the probability of every state should incorporate the number of items belonging to both important factors and the class of item currently getting service. We denote it by $p(m, n, r, 1)$, such that: $p(m, n, r, 1) = Pr[m \text{ and } n \text{ units of importance factor 1 and 2 are present in the system and a unit of importance factor } r = 1(or\ 2) \text{ is in service, the system is in the pull mode}]$. Proceeding in a similar manner as shown in Section 4.1, we can obtain the steady state balanced equations of the prioritized pull-system as:

$$\begin{aligned}
(\lambda_1+\lambda_2+\mu_2)p(m,n,2,1) &= \lambda_1 p(m-1,n,2,1) \\
&\quad +\lambda_2 p(m,n-1,2,1) \\
(\lambda_1+\lambda_2+\mu_2)p(m,n,1,1) &= \lambda_1 p(m-1,n,2,1) \\
&\quad +\lambda_2 p(m,n-1,2,1) \\
&\quad +\mu_2[p(m+1,n,1,1) \\
&\quad +p(m,n+1,1,1)] \\
(\lambda_1+\lambda_2+\mu_2)p(m,1,2,1) &= \lambda_1 p(m-1,1,2,1) \\
(\lambda_1+\lambda_2+\mu_2)p(1,n,1,1) &= \lambda_2 p(1,n-1,1,1) \\
&\quad +\mu_2[p(2,n,1,1) \\
&\quad +p(1,n+1,2)] \\
(\lambda_1+\lambda_2+\mu_2)p(0,n,2,1) &= \lambda_2 p(0,n-1,2,1) \\
&\quad +\mu_2[p(1,n,1,1) \\
&\quad +p(0,n+1,2,1)] \\
(\lambda_1+\lambda_2+\mu_2)p(m,0,1,1) &= \lambda_1 p(m-1,0,1,1) \\
&\quad +\mu_2[p(m+1,0,1,1) \\
&\quad +p(m,1,2,1)] \\
(\lambda_1+\lambda_2+\mu_2)p(0,1,2,1) &= \lambda_2 p(0,0,0,1) \\
&\quad +\mu_2[p(1,1,1,1) \\
&\quad +p(0,2,2,1)] \\
(\lambda_1+\lambda_2+\mu_2)p(1,0,1,1) &= \lambda_1 p(0,0,0,1) \\
&\quad +\mu_2[p(2,0,1,1) \\
&\quad +p(1,1,2,1)] \\
(\lambda_1+\lambda_2)p(0,0,0,1) &= \mu_2[p(1,0,1,1) \\
&\quad + p(0,1,2,1)] \qquad (7)
\end{aligned}$$

It should be noted that the probability of the idle state, i.e., $p(0,0,0,0) = p(0,0)$ remains same as before. The reason behind this is that the ordering of service does not affect the probability of idleness; i.e., $p(0,0) = 1 - \rho - \frac{\rho}{f}$. Now, the occupancy of the pull states is ρ. Hence the fraction of time, the pull-system is busy with type-1 and type-2 items is given by: $\rho\lambda_1/\lambda$ and $\rho\lambda_2/\lambda$. Thus we have,

$$\begin{aligned}
\sum_{m=1}^{C}\sum_{n=0}^{C} p(m,n,1,1) &= \frac{\lambda_1}{\mu} \quad (a) \\
\sum_{m=0}^{C}\sum_{n=1}^{C} p(m,n,2,1) &= \frac{\lambda_2}{\mu} \quad (b) \qquad (8)
\end{aligned}$$

Obtaining a reasonable solution to these set of stationary equations is almost impossible. All we can is to achieve an expected measure of the system performance. We perform two successive z-transforms over the Equations 8 (a)–(b), to get one and two dimensional z-transformed equations in the following way:

$$\begin{aligned}
P_{m1}(z) &= \sum_{n=0}^{\infty} z^n p(m,n,1,1) \\
P_{m2}(z) &= \sum_{n=1}^{\infty} z^n p(m,n,2,1) \qquad (9) \\
H_1(y,z) &= \sum_{m=1}^{\infty} y^m P_{m1}(z) \\
H_2(y,z) &= \sum_{m=1}^{\infty} y^m P_{m2}(z) \qquad (10)
\end{aligned}$$

Combining the above two-dimensional z-transforms we have:

$$\begin{aligned}
H(y,z) &= H_1(y,z) + H_2(y,z) + p(0,0,0,1) \\
&= \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} y^m z^n (p_{m,n,1,1} + p_{m,n,2,1}) \\
&\quad + \sum_{m=1}^{\infty} z^n p(m,0,1,1) \\
&\quad + \sum_{n=1}^{\infty} z^n p(0,n,2,1) + p(0,0,0,1) \qquad (11)
\end{aligned}$$

Solution of the above equations results in:

$$\begin{aligned}
H(y,z) &= H_1(y,z) + H_2(y,z) + p(0,0,0,1) \\
&= \frac{p(0,0,0,1)(1-y)}{1-y-\rho y(1-z-\lambda_1 y/\lambda + \lambda_1 z/\lambda)} \\
&\quad + \frac{(1+\rho-\rho z+\lambda_1 z\mu_2)(z-y)P_{0,2}(z)}{A \times B} \\
A &= z[1+\rho-\lambda_1 y/\mu_2 - \lambda_2 z/\mu_2] \\
B &= [1-y-\rho y(1-z-\lambda_1 y/\lambda + \lambda_1 z/\lambda)] \qquad (12)
\end{aligned}$$

The above equation provides the final solution of the z-transforms associated with the two different pri-

Proceedings of the 2005 International Conference on Parallel Processing (ICPP'05)
0190-3918/05 $20.00 © 2005 **IEEE**

ority classes of clients. This equation will help us in obtaining the average performance of both the priority classes and also the overall expected system performance. As discussed earlier in the previous subsection, differentiating this equation will provide the average number of items present in the system. If L_1 and L_2 represents the average number of items for both the classes then,

$$L_1 = \left[\frac{\partial H(y,z)}{\partial y}\right]_{y=z=1} \quad \text{and} \quad L_2 = \left[\frac{\partial H(y,z)}{\partial z}\right]_{y=z=1} \tag{13}$$

The expected waiting time of the data items having two different importance factors now can be easily found by using the Little's formula as: $E[W_1] = L_1/\lambda_1$ and $E[W_2] = L_2/\lambda_2$.

4.2.2 Effect of Multiple Service Classes

The outline of the above procedure however fails to capture the expected system performance when number of importance-factors increase over 2. Thus a better way is to follow a *direct expected value approach* [4]. Considering a non-preemptive system with many importance-factors, let us assume the data items with importance-factor ϱ_j have an arrival rate and service time of λ_j and $\mu_{2\,j}$ respectively. The occupancy arising due to this j^{th} data item is represented by $\rho_j = \frac{\lambda_j}{\mu_{2\,j}} (1 \leq j \leq max)$, where max represents maximum possible value of importance-factor. Also let σ_j represents the sum of all occupancy factors ρ_i, i.e., $\sigma_j = \sum_{i=1}^{j} \rho_i$. In the boundary conditions we have, $\sigma_0 = 0$ and $\sigma_{max} = \rho$. If we assume that a data item of importance-factor i arrives at time t_0 and gets serviced at time t_1, then the wait is $t_1 - t_0$. Let at t_0 there are n_j data items present having priorities j. Also let, S_0 be the time required to finish the data item already in service, and S_j be the total time required to serve n_j. During the waiting time of any data item, n'_j new items having higher importance-factor can arrive and go to service before the current item. If S'_j be the total service time required to service all the n'_j items, then the expected waiting time for the i^{th} item will be,

$$E[W_{pull}^{(i)}] = \sum_{j=1}^{i-1} E[S'_j] + \sum_{j=1}^{i} E[S_j] + E[S_0] \tag{14}$$

In order to get a reasonable estimate of $W_{pull}^{(i)}$, three components of Equation 14 needs to individually evaluated.

(i) *Estimating* $E[S_0]$: The random variable S_0 actually represents the remaining time of service, and achieves a value 0 for idle system. Thus, the computation of $E[S_0]$ is performed in the following way:

$$\begin{aligned} E[S_0] &= Pr[\text{Busy-System}].E[S_0|\text{Busy-System}] \\ &= \rho.\sum_{j=1}^{max} E[S_0|\text{Serving item, importance-factor} = j] \\ &\quad \times Pr[\text{item having importance-factor} = j] \\ &= \rho \times \sum_{j=1}^{max} \frac{\rho_j}{\rho\mu_{2j}} = \sum_{j=1}^{max} \frac{\rho_j}{\mu_{2j}} \end{aligned} \tag{15}$$

(ii) *Estimating* $E[S_j]$*:* The inherent independence of Poisson's process gives the flexibility to assume the service time $S_j^{(n)}$ of all n_j customers to be independent. Thus, an estimate of $E[S_j]$ can be obtained using the following steps:

$$\begin{aligned} E[S_j] &= E[n_j S_j^{(n)}] = E[n_j]E[S_j^{(n)}] \\ &= \frac{E[n_j]}{\mu_{2j}} = \rho_j E[W_{pull}^{(j)}] \end{aligned} \tag{16}$$

(iii) *Estimating* $E[S'_j]$*:* Proceeding in a similar way and assuming the uniform property of Poisson's,

$$E[S'_j] = \frac{E[n'_j]}{\mu_{2j}} = \rho_j E[W_{pull}^{(i)}] \tag{17}$$

The solution of Equation 14 can be achieved by combining the results of Equations 15–17 and using Cobham's iterative induction [4]. The expected waiting time of the i^{th} item and the overall expected waiting time of the pull system is given as:

$$\begin{aligned} E[W_{pull}^{(i)}] &= \frac{\sum_{j=1}^{max} \rho_j/\mu_{2j}}{(1-\sigma_{i-1})(1-\sigma_i)} \\ E[W_{pull}^{q}] &= \sum_{i=1}^{max} \frac{\lambda_i E[W_{pull}^{q(i)}]}{\lambda} \end{aligned} \tag{18}$$

The overall expected access time is obtained by combining the time taken to service the push and pull items. Since, the push set contains K items of heterogeneous lengths $L_1, L_2, \ldots, L_K$, the average length of the push (broadcast) cycle is $\frac{1}{2}\sum_{i=1}^{K} L_i\mathcal{P}_i$. Thus, the expected access-time ($E[T_{hyb-acc}]$) of our hybrid system is now given by:

$$E[T_{hyb-acc}] = \frac{1}{2\mu_1}\sum_{i=1}^{K} L_i\mathcal{P}_i + E[W_{pull}^{q}] \sum_{i=k+1}^{D} \mathcal{P}_i, \tag{19}$$

where K is the cutoff-point used to segregate push and pull components of the hybrid system. It should be noted that one major objective of our proposed algorithm is to find out an optimal cutoff-point K such that this delay is minimized. The above expression provides an estimate of the average delay (waiting time) for different class of clients in our hybrid scheduling system. The service providers always try to reduce the delay of the high priority clients, in order to ensure their satisfaction. Apart from this delay, we would like get an estimate of the prioritized cost associated with each class of client. This cost is

Proceedings of the 2005 International Conference on Parallel Processing (ICPP'05)
0190-3918/05 $20.00 © 2005 **IEEE**

actually obtained as $q_j \times E[T_{hyb-acc}]$. Intuitively this cost provides an estimate of the client's influence on the service provider and the overall system.

5 Simulation Experiments

In this section we validate the performance analysis of our prioritized hybrid system by performing simulation experiments. We first enumerate the set of assumptions used in our simulation. Subsequently, we provide the series of simulation results obtained.

5.1 Assumptions

1. The simulation experiments are evaluated for a total number of data items $D = 100$.
2. The overall average arrival rate λ' is assumed to be 5. The value of μ_1 and μ_2 is estimated as: $\mu_1 = \sum_{i=1}^{K}(\mathcal{P}_i \times L_i)$ and $\mu_2 = \sum_{i=K+1}^{D}(\mathcal{P}_i \times L_i)$.
3. The length of the data items are varied from 1 to 5, with an average of 2.
4. In order to keep the access probabilities of the items from similar to very skewed, θ is dynamically varied from 0.20 to 1.40. More specifically, we have assumed $\theta = \{0.20, 0.60, 1.0, 1.40\}$.
5. The entire set of clients is divided into three classes: *Class-A, having highest priority*, *Class-B with medium priority* and *Class-C with lowest priority*. The priorities are taken in the ratio $1 :: 2 :: 3$. The fraction α associated in deriving the importance-factor is assumed to be in the range $[0, 1]$, where $\alpha = 1$ indicates the system ignoring the effect of priority and $\alpha = 0$ indicates the system ignoring the effect of stretch.
6. The distribution of clients among different classes is also assumed to obey Zipf's distribution, with lowest number of highest priority (Class-A) clients and highest number of lowest priority clients.

Now we describe the set of simulation results obtained from our simulation experiments.

5.2 Overall Expected Delay

The goal of the first set of experiments is to investigate into the overall delay experienced by each class of clients. Figures 3– 4 demonstrate the dynamics of total delay with the cut-off point experienced by three different classes of clients for $\alpha = \{0, 0.25, 0.50, 0.75, 1.0\}$ respectively. This is performed for different values of access skewness. The delay associated with the Class-A (highest priority) clients is very low (within 5–10 broadcast units). The delay experienced by the Class-B clients remains in the range 20–40 broadcast units. The highest delay (40–70 broadcast units) is experienced by the Class-C clients. However, for all the classes of clients the delay is higher for low values of cut-off point (K). The reason is that for low values of K, the system deviates from the hybrid nature and can not achieve a good balance between push and pull set.

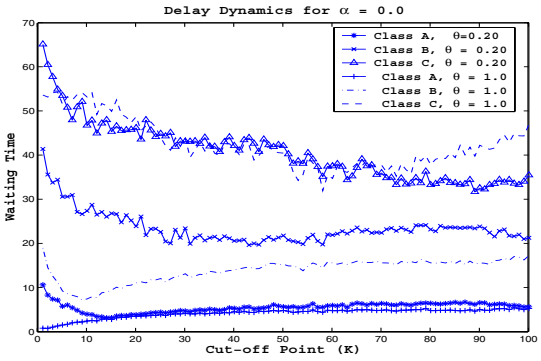

Figure 3: Delay Variation with $\alpha = 0.0$

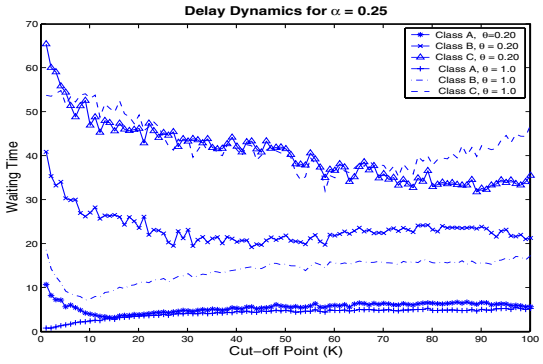

Figure 4: Delay Variation with $\alpha = 1.0$

5.3 Prioritized Costs

The major objective of the second set of experiments is to look into the variation of the prioritized cost associated with each class of clients. As mentioned earlier, the system assigns the costs to each class of clients in proportion to the priority of that particular class. These costs are actually computed by multiplying the priority of the client-class with the expected delay. Figure 5 demonstrates the variation of prioritized costs with the cut-off point, associated with each class of clients for $\alpha = \{0.25, 0.75\}$ and $\theta = 0.60$. The overall objective is to pick up the particular value of cut-off point such that the total prioritized cost is minimized. Figure 6, on the other hand, shows the changes in total optimal prioritized cost of all the client-classes, with different values of α for $\theta = \{0.20, 0.60, 1.40\}$. With decreasing values of α the influence of priority increases and the prioritized cost reduces. The underlying reason is that for lower values of α the increased influence of priority results in serving the important clients first, thereby

Proceedings of the 2005 International Conference on Parallel Processing (ICPP'05)
0190-3918/05 $20.00 © 2005 **IEEE**

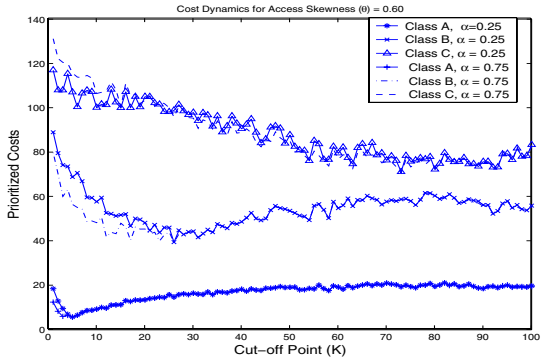

Figure 5: Cost Dynamics for Service Classes

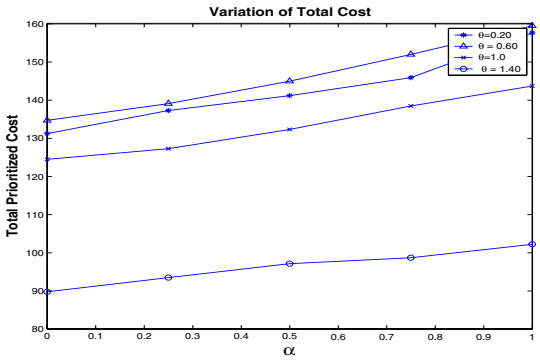

Figure 6: Variation of Prioritized Cost

reducing the overall cost of the system.

5.4 Simulation and Analytical Results

Figure 7 demonstrates the comparison between analytical and simulation results for $\theta = 0.60$ and $\alpha = 0.75$. The analytical results are obtained using the Equation 19. We have chosen the values of α and θ so that these values are almost in the middle of their range. Analytical results closely match simulation results for all the three set of clients, with a minor 10% deviation. The minor deviation is attributed to the memory-less assumption in the system modelling.

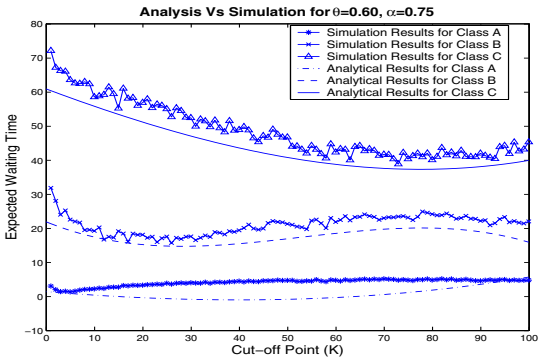

Figure 7: Analytical Vs. Simulation Results

6 Conclusion

In this paper we have proposed a new service classification strategy for hybrid scheduling to support differentiated services in wireless data networks. The hybrid scheduling effectively combines the push and pull systems. The major novelty of the work lies in differentiating the clients into various classes based on their priorities. Subsequently, it uses a linear combination of the clients' priorities and the probabilities of the data items to form a new selection criteria for the pull-system. This is more practical as the system should pay more attention towards the clients having higher importance than the clients having lower importance. By obtaining an optimal cut-off point between the push and pull items the framework minimizes the overall prioritized costs associated to maintain the clients in the system. Performance analysis and simulation results demonstrate that the average waiting time for the premium clients can be significantly reduced, thereby minimizing the total overall cost associated in the system and increasing an overall efficiency of the system and profit of the service providers.

References

[1] S. Acharya, R. Alonso, M. Franklin and S. Zdonik. Braodcast Disks: Data Management for Asymmetric Communication Environments, *Proceedings of ACM SIGMOD Conf.*, pp. 199-210, May 1995.

[2] S. Acharya, M. Franklin, and S. Zdonik. Balancing push and pull for data broadcast. *Proceedings of the ACM SIGMOD Conference*, pp. 183–193, May, 1997.

[3] D. Aksoy and M. Franklin. RxW: A scheduling approach for large scale on-demand data broadcast. *IEEE/ACM Transactions on Networking*, Vol. 7, No. 6, pp. 846-860, Dec. 1999.

[4] D. Gross and C. M. Harris, Fundamentals of Queuing Theory, *John Wiley & Sons Inc.*

[5] S. Hameed and N. H. Vaidya. Efficient algorithms for scheduling data broadcast In *WINET*, Vol. 5, pp. 183-193, 1999.

[6] G. Lee and S. C. Lo. Broadcast Data Allocation for Efficient Access of Multiple Data Items in Mobile Environments. *Mobile Networks and Applications*, Vol. 8, pages 365-375, 2003.

[7] C-W Lin, H. Hu and D-L Lee, "Adaptive Realtime Bandwidth Allocation for Wireless Date Delivery", *ACM/Kluwer Wireless Networks*, (WINET), vol. 10, pp. 103-120, 2004.

[8] M. Mahajan and M. Pashar "Managing QoS for Multimedia Applications in a Differentiated Services Environment", *Active Middleware Services (AMS)*, 2002.

[9] M. C. Pinotti and N. Saxena. Push less and pull the current highest demanded data item to decrease the waiting time in asymmetric communication environments. *4th International Workshop on Distributed and Mobile Computing, Springer-Verlag, (LNCS)*, (IWDC), pp. 203–213, 2002.

[10] N. Saxena, K. Basu and S. K. Das, "Design and Performance Analysis of a Dynamic Hybrid Scheduling for Asymmetric Environment", *IEEE Intl. Workshop on Mobile Adhoc Networks*, WMAN, 2004.

[11] Z. Wu and D. Raichaudhuri, " D-LSMA: Distributed Link Scheduling Multiple Access Protocol for QoS in Ad-Hoc Networks" *Proc. of IEEE GlobeCom*, 2004.

Proceedings of the 2005 International Conference on Parallel Processing (ICPP'05)
0190-3918/05 $20.00 © 2005 **IEEE**