ISSN: 0976-2876 (Print)
ISSN: 2250-0138(Online)

HYBRID SCHEDULING FOR IMPATIENT CLIENTS WITH RENEGING AND BALKING

R. KAVITHA[a1] AND S. KRISHNA MOHAN RAO[b]

[a]Rayalaseema University, Kurnool, Andhra Pradesh, India
[b]Gandhi Institute of Technology, Bhubaneshwar, Khurda, India

ABSTRACT

The efficiency of scheduling algorithm lies in providing multimedia applications and web applications with data processing abilities. This paper introduces hybrid scheduling model which efficiently combines push and pull (on-demand) models for heterogeneous environment. The proposed algorithm considers variable sized data items and computes access patterns of data items by dividing the database at the server into two sets, pull and push. In order to deliver data items of push set hashing technique is used and for pull data items M/M/1 and M/M/C queuing models are employed for single pull channel and multiple pull channels respectively. The essence of this algorithm lies in reducing the access time by considering the dynamics of the system. Dynamics of the system considers impatient clients who leave the system before they are served, with the required data item. The proposed algorithm analyzes the system performance in terms of average access time in retention of impatient clients. The analytical model shows that the proposed algorithm performs well in handling impatient clients.

KEYWORDS: Average Access Time, Balking, Client's Impatience, Renege

The advancement in current technology in web has given rise to serve the needy efficiently with the required data. To extract the required data and transmit it there is a need for broadcasting strategies In the mobile computing environment, information dissemination no longer requires a user to maintain a fixed position in the network. The wireless medium enables virtually unrestricted mobility and portability. A mobile computing environment requires two major components: a wireless network, and a portable computing platform. Currently used wireless networks are cellular networks, wireless LAN, wide-area wireless networks and paging networks. Mobile computing platforms include palmtops, Personal Digital Assistants (PDA); and personal computers. There are unique characteristics of wireless networks have significant impact on wireless data management. These include - channel limitation, asymmetry in the communications: the bandwidth of downlink (server-to-client) is greater than the uplink (client-to-server), frequent disconnection and relocation (move from one place to another): mobile users tend to switch their units on and off regularly and, power limitation: most of the portable units use batteries that require frequent recharge.

Information Dissemination is the most important application of mobile computing. In general, there are two basic methods to provide users with information. They are pull and push methods [Acharya and Muthukrishnan, 1998] [Acharya et. al., 1995]. In this paper, hybrid method is addressed with a variable data item size, which is the combination of both push and pull models.

In a hybrid push-pull environment, popular data items are broadcasted in the form of a broadcast disk, and less popular items are pulled from the server via explicit client requests. To retrieve individual records of interest, mobile users can either monitor the broadcast channels for the arrival of desired data or issue a pull request to the server. The broadcast disk can be viewed as storage on the air or as an extension of the server's memory, which alleviates pull-based requests considerably. Since the wireless broadcast channel is a limited resource in a mobile environment, data distribution should be carefully managed to achieve efficient utilization of the resources. This paper investigates a method to efficiently broadcast data. The existing algorithms [Acharya et. al., 1995], [Hameed and Vaidya, 1999], [Wong, 1988] and [Aksoy and Franklin, 1999] are not capable of handling heterogeneous environments where the data items have different sizes. The algorithm defined in [Acharya and Muthukrishnan, 1998] and [Yiqiong and Guohong, 2001], focuses on these heterogeneous environments but uses single channel for broadcasting the data items. The above mentioned problem is solved by proposing a scheduling algorithm that is made available not only to the heterogeneous environment where the data items are of variable sized but also reduces the access time by effectively utilizing multiple channels for broadcasting the data items. Multiple channels [Prabhakara et. al., 2000] have capabilities and applications that cannot be mapped on to single channels. As they have several advantages which include better fault tolerance, configurability and scalability.

For achieving optimal average access time, the

[1] **Corresponding Author**

schedule related to packet fair queuing model discussed in [Hameed and Vaidya, 1999] is used. For asymmetric communication, a three-player model [Bar-Noy et. al., 2003], defined logically divides the clients, servers and service providers. The authors of [Peng et. al., 2003] and [Hu and Chen, 2003] derive a method to be adaptive which considers changes in the mobile client's requests. On the other hand, many pull algorithms with preemption and non-preemption are proposed in [Acharya et. al., 1995], [Bar-Noy et. al., 2004] and [Hameed and Vaidya, 1999] but lacks in addressing the heterogeneity of data Items, dynamic computation of access patterns (probabilities) of the data items and lack of patience in the clients due to waiting for required data item.

```
HYBRID SCHEDULING FOR RENEGING CLIENTS
While true do
Begin
    Phase I:
        1. Broadcast the data items according to hashing
           technique
        2. Handling the requests coming in during broadcast
            If the request received for push data item
            then ignore the request
            else
             compute the reneging rate(dropping rate) and
             decrease the access probability of the data item
             update the stretch value of the data items in pull queue;
        3. if $Q_{pull}$ not empty
    Phase II:
       4. extract an item from the $Q_{pull}$ with max stretch
       5. if there is a tie
            extract the item whose access probability is high
        5. clear all the pending requests for that item
        Go to step 2
End
```

Figure 1: Hybrid Scheduling For Reneging Clients

Section II introduces the problems with client's impatience.

Section III starts with modeling the proposed system and ends with defining the algorithm. Simulation results in Section IV show the performance of the proposed algorithm. Finally, Section V concludes the paper with pointing out the future enhancement.

CLIENT IMPATIENCE

In most of the cases involving practical application, it has been observed that the clients lose patience whilst waiting for the required data item. This, after a certain limit of tolerance, results in the client moving away from the system ensuing in a drop of access requests. This conjectures the performance of the system which in turn reflects the immediate need for corrective behavior. This is profoundly found in cases reflecting the false situation of the system. Considering the scenario where numerous requests for a single data item is received by the server due to client's impatience; in turn increases the access probability, even when not requested by different clients . The access rate multiplies. In obtainable systems, the server remains unaware of the information and hence considers the data item as popular and thereby inserts it in the push or pull set at the expense of other popular items.

So as a summary the problems with client impatience are

1) Leaving the system without being severed (Reneging)
2) Does not join the system after finding it as busy (Balking)

HYBRID SCHEDULING

Hybrid scheduling is obtained by combing push and pull models. This scheduling partitions the database at server into two sets push and pull. The push set (queue) contains the data items to be broadcasted. Hashing technique is applied to broadcast the data items. For pull, M/M/1 queuing model is considered with reneging [Kumar and Sharma, 2012].

Modeling the System

The proposed algorithm uses Hybrid model. Let us specify the parameters and assumptions used in our model:

1) We assume that there are many clients who are been served by a single server. Figure1 shows the asymmetric communication environment where there is a single server and the downlink capacity of the channel is more than that of the uplink channel capacity.
2) The server maintains a database and it is assumed that it consists of N number of data items, each with different lengths (size). Let server push M data items and (N-M) are pulled. The service time is assumed to be dependent on the size of the data item. Larger the

size of data item higher its service time. The size of the data item i, is denoted as s_i. The stretch of the data items are calculated as $S(i) = \frac{P_i}{s_i}$.

3) The server supports C channels and each channel has a bandwidth b. Among these channels,$C_{pull}$are number of on-demand or pull (uplink) channels, for accepting client's requests and C_{push} are the broadcast channels. So, $C = C_{push} + C_{pull}$.

4) Clients send requests to the server for their required data item. The requests from the client are accumulated as access probabilities.

5) The access probabilities of data items are skewed and modeled by ZipF distribution, which is typically used to model non-uniform access probabilities [Aksoy and Franklin, 1999]. We use the same mathematical formulation for the ZipF distribution. That is, if we label the items in decreasing order of popularity, the probability of particular request for item i is given by $p_i = \frac{(1/i)^\theta}{\sum_{j=1}^{N}(1/j)^\theta}$ where θ is access coefficient.

As server broadcasts M data items, so the size of Q_{push} is M. And the sizeofQ_{pull} is (N-M). Thus the total access probabilities of data items in $Q_{push}$is $\sum_{i=1}^{M} p_i$and in pull queue is $\sum_{i=M+1}^{N} p_i$

6) The time taken between successive requests and the service times of the server are exponentially distributed. Thus the service rate of Q_{push} is $\mu_{push} = \sum_{i=1}^{M} p_i \times s_i$ and Q_{pull} is$\mu_{pull} = \sum_{i=M+1}^{N} p_i \times s_i$.

7) Let λ be the arrival rate of the system and the requests that are send by the clients be $\lambda_{pull} = \left(1 - \sum_{i=M+1}^{N} p_i\right) \times \lambda$

8) Client upon putting request will wait for certain amount of time before the required data item is broadcasted. If not began, client get impatient or reneged and leave the system with probabilityq. The reneging time follows exponential distribution with ξ as parameter. When $\xi = 0$ means that the clients never leave the system before its service. This indicates that it is retention of reneged clients

9) On arriving client finds the server busy and does put request i,e may balk with q' or join with probability $1 - q'$. If $q' = 0$ represents client join the system.

As hybrid broadcast results from the combination of push and pull models, we will start with formulating the push and pull models and then combine those models to calculate the overall access time for the proposed hybrid algorithm.

Analytical Model for Push Model to Minimize Expected Delay

The server has N number of data items where for M items are for broadcasting and remaining N-M are the requests put by the clients. The server picks the data items from the Q_{push} and transmits them on to the C_{push} broadcast channels. The number of data items that have to be broadcasted on to each channel is determined by hashing function. The hashing function is defined as: $f(s_i) = s_i \mathrm{mod} C_{push}$ where s_i is the size of the data item i. Store these $f(s_i)$ values and arrange them in ascending order. All data items with same $f(s_i)$ are broadcasted onto the same channel. So as a result there will be m_i , number of data items in each broadcast channel i where i = 1,2,3,…C_{push}. These m_i values differ from channel to channel. Each channel has bandwidth, b. As C_{push} channels are used for broadcast, the total bandwidth of these channels is bC_{push}. The average access time for a broadcast data item is the average time spent on waiting for that item. As the bandwidth of each push channel is assumed to be b and the average size of a data item in a channel i, $S_{tot} = \sum_{j=1}^{m_i} s_j$, the propagation time for each data item of a single broadcast channel is equal to S_{tot}/b. The average probe time for all channels is $S_{tot}/2bC_{push}$ In skewed broadcast system the expected waiting time ($E[T_{\mathbf{C_{push}}}]$) for a data item in the push channel i given as:

$$E[T_{\mathbf{C_{push}}}](\mathbf{i}) = \frac{\mathbf{s_i}}{\mathbf{b}} + \frac{\mathbf{S_{tot}}}{\mathbf{2bC_{push}}} \qquad (1)$$

The Average access time (AAT) of all the push or broadcast channels will be derived as

$$A[T_{C_{push}}] = \sum_{i=1}^{M} pi \times E[T_{\mathbf{C_{push}}}](\mathbf{i}) \qquad (2)$$

$$A[T_{C_{push}}] = \left(\sum_{i=1}^{M} pi \times \frac{\mathbf{s_i}}{\mathbf{b}}\right) + \frac{\mathbf{S_{tot}}}{\mathbf{2bC_{push}}} \qquad (3)$$

Analytical Model for Pull Model to Minimize waiting Time

When Reneging is Considered

For pull, M/M/1 queuing model with reneging is used where n is the number of requests from the clients and N is pull system capacity. According to [Kumar and Sharma, 2012] queuing model with retention of reneged clients with

$$\mathrm{Pr}_{\mathrm{n}} = \left[\prod_{\mathrm{k=M+1}}^{\mathrm{n}} \frac{\lambda_{\mathrm{pull}}}{\mu_{\mathrm{pull}}+(\mathrm{k}-1)\xi \mathrm{q}}\right] \mathrm{Pr}_0; 1 \le \mathrm{n} \le \mathrm{N} \qquad (4)$$

$$\mathrm{Pr}_0 = \frac{1}{1+\sum_{\mathrm{n=M+1}}^{\mathrm{N}}\left[\prod_{\mathrm{k=M+1}}^{\mathrm{n}} \frac{\lambda_{\mathrm{pull}}}{\mu_{\mathrm{pull}}+(\mathrm{k}-1)\xi \mathrm{q}}\right]}$$

As the clients become inpatient and leave the system there by dropping the request for the data item and is calculated as

$$\mathrm{R}_{\mathrm{drop}} = \lambda_{\mathrm{pull}} \sum_{\mathrm{n=M+1}}^{\mathrm{N}} \mathrm{Pr}_{\mathrm{n}} - \sum_{\mathrm{n=M+1}}^{\mathrm{N}} \mathrm{n}\mu_{\mathrm{pull}} \left[\prod_{\mathrm{k=M+1}}^{\mathrm{n}} \frac{\lambda_{\mathrm{pull}}}{\mu_{\mathrm{pull}}+(\mathrm{k}-1)\xi \mathrm{q}}\right] \mathrm{Pr}_0 \qquad (5)$$

The above is the model defined for single pull(on-demand) channel .

For multi pull channel M/M/C$_{\mathrm{pull}}$ model defined as

$$\mathrm{Pr}_{\mathrm{n}} = \frac{1}{\mathrm{n}!}\left(\frac{\lambda_{\mathrm{pull}}}{\mu_{\mathrm{pull}}}\right)^{\mathrm{n}} \mathrm{Pr}_0 + \left[\prod_{\mathrm{k=C_{pull}+1}}^{\mathrm{n}} \frac{\lambda_{\mathrm{pull}}}{\mathrm{C_{pull}}\mu_{\mathrm{pull}}+(\mathrm{k}-\mathrm{C_{pull}})\xi \mathrm{q}}\right] \frac{1}{\mathrm{C_{pull}}!}\left(\frac{\lambda_{\mathrm{pull}}}{\mu_{\mathrm{pull}}}\right)^{\mathrm{C_{pull}}} \mathrm{Pr}_0;\ 1 \le 1 \le \mathrm{n} \le \mathrm{N} \qquad (6)$$

Where

$$\mathrm{Pr}_0 = \left[1+\sum_{\mathrm{n=1}}^{\mathrm{C_{pull}}} \frac{1}{\mathrm{n}!}\left(\frac{\lambda_{\mathrm{pull}}}{\mu_{\mathrm{pull}}}\right)^{\mathrm{n}} + \sum_{\mathrm{n=C_{pull}+1}}^{\mathrm{N}} \prod_{\mathrm{k=C_{pull}+1}}^{\mathrm{n}} \frac{\lambda_{\mathrm{pull}}}{\mathrm{C_{pull}}\mu_{\mathrm{pull}}+(\mathrm{k}-\mathrm{C_{pull}})\xi \mathrm{p}} \frac{1}{\mathrm{C_{pull}}!}\left(\frac{\lambda_{\mathrm{pull}}}{\mu_{\mathrm{pull}}}\right)^{\mathrm{C_{pull}}}\right]^{-1}$$

Dropping rate is defined for multi-channel pull system as

$$\mathrm{R}_{\mathrm{drop}} = \lambda_{\mathrm{pull}} \sum_{\mathrm{n=1}}^{\mathrm{N}} \mathrm{Pr}_{\mathrm{n}} - \sum_{\mathrm{n=1}}^{\mathrm{C_{pull}}} \mathrm{n}\mu_{\mathrm{pull}} \frac{1}{\mathrm{n}!}\left(\frac{\lambda_{\mathrm{pull}}}{\mu_{\mathrm{pull}}}\right)^{\mathrm{n}} \mathrm{Pr}_0 + \left[\sum_{\mathrm{n=C_{pull}}}^{\mathrm{N}} \mathrm{C_{pull}}\,\mu_{\mathrm{pull}} \prod_{\mathrm{k=C_{pull}+1}}^{\mathrm{n}} \frac{\lambda_{\mathrm{pull}}}{\mathrm{C_{pull}}\mu_{\mathrm{pull}}+(\mathrm{k}-\mathrm{C_{pull}})\xi \mathrm{p}}\right] \frac{1}{\mathrm{C_{pull}}!}\left(\frac{\lambda_{\mathrm{pull}}}{\mu_{\mathrm{pull}}}\right)^{\mathrm{C_{pull}}} \mathrm{Pr}_0; \qquad (7)$$

Where $\sum_{\mathrm{n=1}}^{\mathrm{N}} \mathrm{Pr}_{\mathrm{n}} = 1$

Expected pull system size is obtained as

$$\mathrm{Pull}_{\mathrm{SysSize}} = \sum_{\mathrm{n=M+1}}^{\mathrm{N}} \mathrm{nPr}_{\mathrm{n}} \qquad (8)$$

If $\mathrm{Pr}_{\mathrm{n}}$is substituted in (4) then the expected size of the pull system for single channel is obtained. If Pr_{n} is substituted in (6) and expected size for multichannel is derived.

$$\mathrm{E}[\mathrm{TR}_{\mathrm{pull}}] = \frac{\mathrm{Pull}_{\mathrm{Syssize}}}{\lambda_{pull}}\ X \sum_{i=M+1}^{N} p_i \qquad (9)$$

The case of retention of clients comes when reneging parameter is set to 0 ($\xi = 0$).

When Balking with Reneging is Considered

According to the authors of [Kumar and Sharma, 2012] model the system for retaining impatient clients. The M/M/1 queuing model is considered and devised a convincing method to retain the clients. Thus, there is a probability say, q not to retain in system and (1-q) for retention. An arriving requests from client finds the server busy on arrival, may balk with probability q' or join the system with $1 - q'$

So Pr_0 *and* Pr_n are defined as follows:

$$Pr_n = \frac{\lambda_{pull}}{\mu_{pull}}\left[\frac{(\lambda_{pull} \times q')^{n-1}}{\prod_{k=M+1}^{n-1} \mu_{pull}+k\xi q'}\right] Pr_0\,; 1 \le n \le N \qquad \text{and}$$

$$Pr_{n0} = 1 + \frac{\lambda_{pull}}{\mu_{pull}} + \sum_{n=M+1}^{N} \frac{\lambda_{pull}}{\mu_{pull}}\left[\frac{(\lambda_{pull} \times q')^{n-1}}{\prod_{k=M+1}^{n-1} \mu_{pull} + \mathrm{k}\xi q'}\right]$$

Expected pull system when balking and reneging is considered is defined as $\mathrm{Pull}_{\mathrm{SysSizerb}} = \sum_{\mathrm{n=M+1}}^{\mathrm{N}} \mathrm{nPr}_{\mathrm{n}}$

And waiting time is derived as

$$\mathrm{E}[\mathrm{TRB}_{\mathrm{pull}}] = \frac{\mathrm{Pull}_{\mathrm{Syssizerb}}}{\lambda_{\mathrm{pull}}}\ \mathrm{X} \sum_{\mathrm{i=M+1}}^{\mathrm{N}} \mathrm{p_i} \qquad (10)$$

Hybrid Model

Thus the expected access time for the proposed hybrid model, $\mathrm{E}[\mathrm{T}_{\mathrm{hybrid}}]$, for any data item can be obtained by the combination of both push and pull models and is given by adding (3) with (9).

$\mathrm{E}[\mathrm{T}_{\mathrm{hybrid}}] = \mathrm{A}[\mathrm{T}_{\mathrm{C_{push}}}] + \mathrm{E}[\mathrm{TR}_{\mathrm{pull}}]$ for Reneging

$\mathrm{E}[\mathrm{T}_{\mathrm{hybrid}}] = \mathrm{A}[\mathrm{T}_{\mathrm{C_{push}}}] + \mathrm{E}[\mathrm{TRB}_{\mathrm{pull}}]$ for Reneging and balking

SIMULATION AND EXPERIMENTS

This section performs simulation experiments on the proposed system. The only and important goal is to reduce average access time.

The below are the assupmtions and parameters used to evaluate the simulation.

1. Total number of data items or the Database size N=1000
2. The size of the data items vary from 1 to 5
3. The arrival rate λ is varied 1 to 4. λ_{pull} is assigned accordingly. The values $\mu_{\text{push}} = \sum_{i=1}^{M} p_i \times s_i$ and $\mu_{\text{pull}} = \sum_{i=M+1}^{N} p_i \times s_i$ are estimted where p_i and s_i are access probabilities and size of data item i.
4. The skew coefficient, θ, vary from 0.2 to 1.5
5. The reneging parameter, ξ, varied from 0.1 -0.5. And when it is set to 0. Probability of reneging set to 0.1 We retain the clients requests.
6. The probability of balking q' varied from 0.1 to 0.5
7. The number of channels varied from 5 to 20 .
8. To compare the performance with the proposed hybrid system with clients reneging the work proposed in [Saxena and Pinotti, 2005] is used , as per our knowledge, this is the algorithm considered client impatience but not how to retain them.

In the following discussion we evaluate the results obtained to show the effiiency of our proposed method.

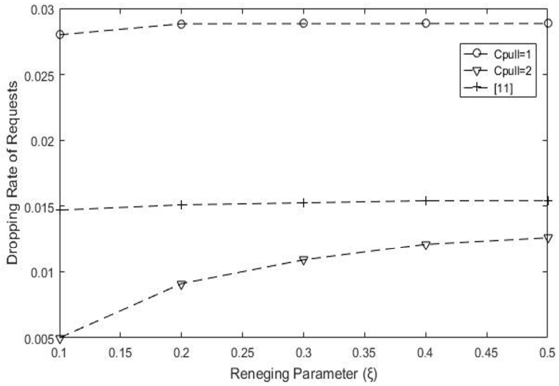

Figure 2: Impact Of Reneging Parameter on Dropping rate

Figure 3 shows that as the reneging parameter increase dropping rate of the requests increase. The algorithm defined in [Saxena and Pinotti, 2005] shows the impact of dropping rate on reneging parameter. But the dropping rate of it is more when compared to our proposed algorithm. And it can be also concluded that as the number of channels for pull increase the dropping rate decreases drastically.

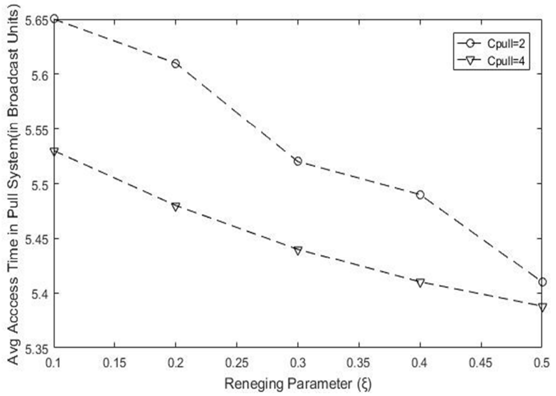

Figure 3: Minimum access Time with variation of reneging Parameter

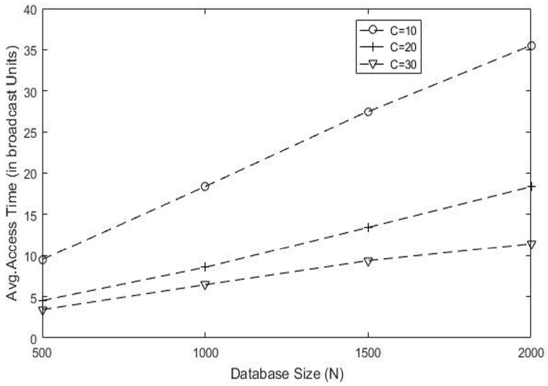

Figure 4: Varying access time with increasing in Database Size

It is observed that the as database size and number of channels increase there will be increase in average access time but increase with slower rate .

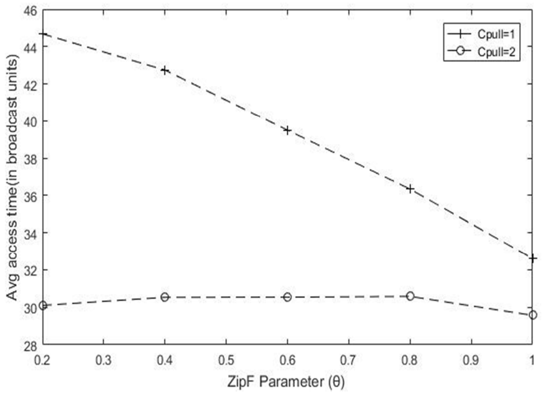

Figure 5: Expected access time with ZipF parameter

As ZipF or Skew Parameter increases the average access time decreases it is depicted in Fig 5.

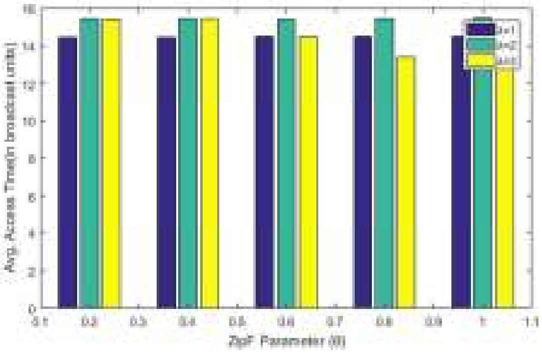

Figure 6: Impact of Arrival times on Avg Access Time

The varying arrival rates of data items have significant affect on the average access time of the system. From Fig 2 it is evident that for different access skewness and with increase in arrival rate the average access time also increases.

Table 1: Values of different parameters when reneging factor is varied.

ξ	Pull System Size	Dropping Rate	Avg Access time
0.1	13.4	0.2103	6.64
0.2	6.775	0.2576	3.96
0.3	4.5	0.26	3.7
0.4	3.4	0.2686	2.6
0.5	2.72	0.27	2.62

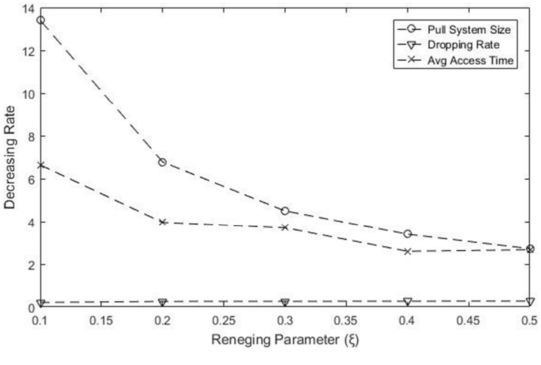

Figure 7: Affect of Reneging Parameter

From Fig 7 and table 1, it can be observed that increasing rate of reneging causes decrease in pull system size. Thus, resulting in increase in dropping rate of requests, it is concluded that more and more requests are dropped. Increase in drop rate will decrease in number of requests causing decrease in average access time.

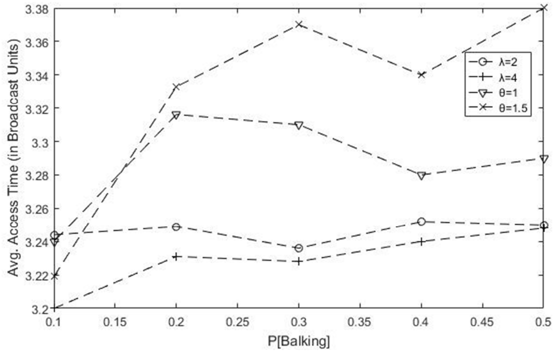

Figure 8: Comparision of Avg Access time with varying arrival rate and skew parameter

For evaluation of balking for the system we have choosen Database size to be, N=100 and M=60. Probability of the balking is varied from 0.1 to 0.5.Skew Parameter varies from 1 to 1.5 and arrival rate varies from 2 to 4.

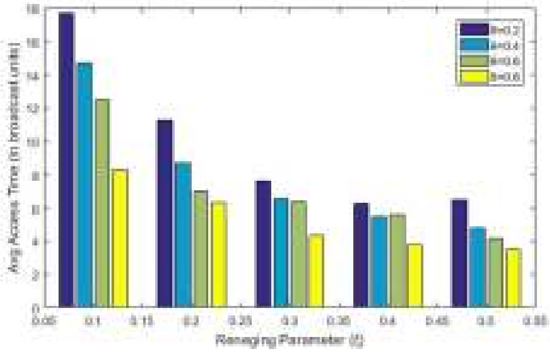

Figure 9: Impact of Reneging parameter

It is concluded from Fig 9 that increase in reneging parameter with varying access coefficient will decrease the access time of the data items requested by the clients.

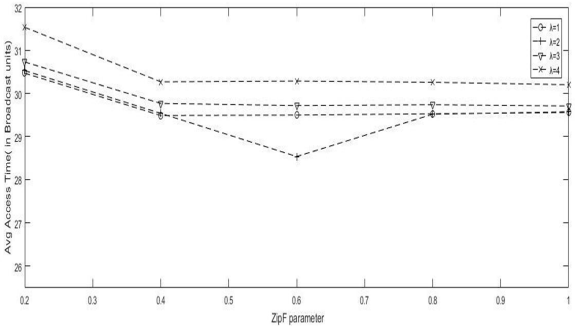

Figure 10: Impact of Retention of Reneged Client's Request

To evaluate the Fig 10 , reneging parameter set to 0 (ξ = 0). And it is evident that the avgerage access time decreases with increase in ZipF (skew) parameter reduced to pull model with push as the impatient clients are retained.

Similarly the model reduces to push model if the probability of balking is set to 0.

[Note : ZipF parameter and Skew Parameter are used Interchangeably. Both are one and the same]

CONCLUSION

It is a tendency of the clients to become impatient while they are waiting for the required data item. They become impatient and leave the system before they are served. This causes changes in the access probabilities and affects the average access time. So this paper has proposed a hybrid scheduling algorithm of to minimize average access time with retention of impatient clients. This work can be extended to handle multiple requests for data item arise from same client.

REFERENCES

Acharya S., Alonso R., Franklin M. and Zdonik S., 1995. Braodcast Disks: Data Management for Asymmetric Communication Environments. Proc.of ACM SIGMOD, San Jose.

Peng W.C., Huang J.L. and Chen M.S., 2003. Dynamic Levelling: Adaptive Data Broadcasting in a Mobile Computing Environment. In Mobile Networks and Applications, **8**: 355-364.

Bar-Noy A., Patt-Shamir B. and Ziper I., 2004. “Broadcast Disks with Polynomial Cost Functions”, ACM/Kluwer Wireless Networks, **10**: 157-168.

Acharya S. and Muthukrishnan S., 1998. Scheduling on-demand broadcasts: New metrics and algorithms. In Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’98), items 43–54, Dallas, TX, USA.

Hameed S. and Vaidya N.H., 1999. Efficient Algorithms for Scheduling Data Broadcast Wireless Networks, **5**: 183-193.

Hu C.-L. and Chen M.-S., 2003. “Adaptive Information Dissemination: An Extended Wireless Data Broadcasting Scheme with Loan-Based Feedback Control, IEEE Trans. on Mobile Computing, **2**(4).

Wong J. W., 1988. Broadcast delivery. Proceedings of the IEEE, **76**(12):1566–1577.

Bar-Noy A., Naor J.S. and Schieber B., 2003. Pushing Dependent Data in Clients-Providers-Servers Systems. In Mobile Networks and Applications, **9**: 421-430.

Aksoy D. and Franklin M., 1999. R x W: A scheduling approach for large scale on-demand data broadcast. IEEE/ACM Transactions on Networking, **7**(6):846–860.

Kumar R. and Sharma S.K., 2012. “Queuing with reneging, balking and retention of Reneged Customers”, International Journal of Mathematical Models and Methods in Applied Sciences, **6**(7).

Saxena N. and Pinotti M.C., 2005. A Dynamic Hybrid Scheduling Algorithm with Clients' Departure for Impatient Clients in Heterogeneous Environments" 5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN).

Yiqiong Wu and Guohong Cao, Stretch-optimal scheduling for on-demand data broadcasts, Computer Communications and Networks, 2001. Proceedings. Tenth International Conference, 2001.

Prabhakara K., Hua K.A. and Oh J., 2000. Multi-level multi-channel air cache designs for broadcasting in a mobile environment. Proceedings of the IEEE International Conference on Data Engineering.