Decoupling Delay-Bandwidth Guarantees in Hierarchical Fair Queuing

Nirav Parikh, Milind Kopikare, Abhay Karandikar
Indian Institute of Technology, Bombay
{email: karandi@ee.iitb.ernet.in}

ABSTRACT—
An efficient and fair scheduling scheme is one of the key components for guaranteeing Quality of Service (QoS) features for various applications over an Integrated Services Network. However almost all the schemes proposed so far have a strong coupling between delay and bandwidth. We address this limitation by modifying the Fair Queuing mechanism and extending it for a hierarchical link. An end to end delay bound for the packets over the hierarchical link has also been analysed and verified through simulations.

I. Introduction

The diverse applications in the emerging Integrated Services Packet Network (ISPN) require various types of Quality of Service (QoS) guarantees in terms of delay, jitter and bandwidth. Various scheduling schemes have been proposed that guarantee the required QoS for each of these service classes [5], [2], [3]

However, all these algorithms have a limitation. There exists a strong coupling between the delay bound and the bandwidth (rate or weight) allocation. The implication of this coupling is that low rate applications like Telnet and packetized voice will have higher delay bound as well as higher average delay as compared with other high rate applications. We address this problem by modifying the Fair Queuing mechanism. We further extend this mechanism and apply it to a hierarchical link structure. We call this scheme as the Hierarchical Modified Generalised Processor Sharing Decoupled (H-MGPS-D) scheme.

II. Fair Queueing

Fair Queueing is based on the idea of Generalized Processor Sharing (GPS) [2]. A GPS server is work conserving and operates at a fixed rate r. Let N be the number of sessions sharing a server. These sessions are allocated weights $\phi_1, \phi_2, ..., \phi_N$. Let $S_i(\tau, t)$ be the amount of session i traffic served in the interval $(\tau, t]$. Session i is backlogged at time t if a positive amount of traffic for i is queued at time t. For any session i that is continuously backlogged during the interval $(\tau, t]$, the GPS server is defined as [2],

$$\frac{S_i(\tau,t)}{S_j(\tau,t)} \geq \frac{\phi_i}{\phi_j}, \quad j = 1, 2, ..., N$$

Let $B(t)$ be the set of backlogged users at time t. As proved in [2], the session i is guaranteed a rate of $g_i = \frac{\phi_i}{\sum_{j \epsilon B(t)} \phi_j} r$

GPS assumes that the traffic is fluid and serves it in bit-by-bit manner. The packet approximation of GPS is called Packet Generalized Processor Sharing (PGPS) [2]. It tracks the order of transmission of packets in the GPS server.

Let packet k from session i arrive at the server at time t_i^k. Let its length be L_i^k. Let r be the server rate. The system virtual time $v(t)$ is defined as [2],

$$v(t_2) - v(t_1) = \frac{r(t_2 - t_1)}{\sum_{j \epsilon \mathrm{B}(t_1)} \phi_j}, \quad v(0) = 0$$

The time when the packet starts service in the GPS server is (virtual start time),

$$S_i^k = \mathrm{MAX}(F_i^{k-1}, v(t_i^k))$$

and the virtual finish time of the packet from the GPS server is defined as

$$F_i^k = S_i^k + \frac{L_i^k}{\phi_i}$$

The packet is stamped with F_i^k. The PGPS server serves the packets in the increasing order of their F_i^k.

III. Modified Fair Queueing

We modify the GPS based server as follows.

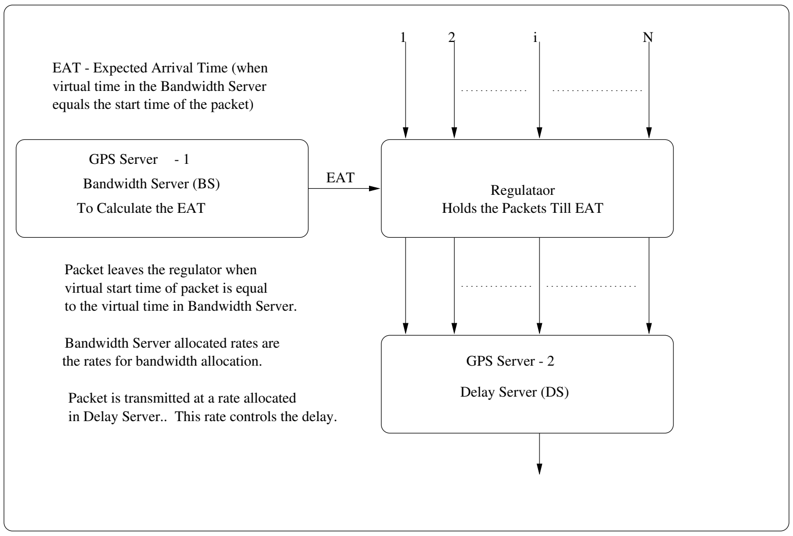

Fig. 1. Scheduler Model for modified GPS

The modified scheduler (server) consists of two GPS servers and a regulator. Both servers are working at the rate r. Session i is allocated a weight of ϕ_i in GPS server 1 (Bandwidth Server) and a weight of θ_i in GPS server 2 (Delay Server). The packets from different sessions that arrive on the regulator are transmitted on the output link from Delay Server (DS). Let us consider that a packet k on session i arrives at the regulator at time t. The same packet also receives service in Bandwidth Server (BS).

- Packet k is held in the regulator till it starts receiving service in BS. This means that the packet k is held in the regulator till the virtual time in BS equals the virtual start time of the packet, *ie*, $S_i^k = v_1(t)$, where $v_1(t)$ is the virtual time in the BS at time t. The virtual times are with respect to the Bandwidth Server with weights ϕ.
- At time t when $v_1(t) = S_i^k$, packet k is released from the regulator and reaches DS. It is served according to the weight θ_i in the DS (Delay Server). The packet is transmitted to the output link from the Delay Server (DS).

In practice, BS weights $\phi's$ can be set according to the average arrival rates for sessions, and the DS weights $\theta's$ can be set according to the promptness (delay) requirements of the sessions. Hence the long term average bandwidth guarantee is satisfied by the BS weights ϕ. The virtual finish time in DS is given by

$$BID_i^k = S_i^k + \frac{L_i^k}{\theta_i} \tag{1}$$

The packets are served in the increasing order of their BID. This ensures that the delay guarantees are also satisfied. We have implemented this scheme for a hierarchical link and verified the performance through analysis as well as simulations. The modified scheme can be shown to decouple delay and bandwidth guarantees to a large extent [1], [6].

We call this modified scheme as the Hierarchical Modified Generalized Processor Sharing-Decoupled scheme (H-MGPS-D). A psuedo code for the implementation of this scheme is given below. n stands for any non root node in the algorithm. Each non root node is connected to its parent node $p(n)$ by a logical queue Q_n. $num.child.(n)$ tells us the number of child queues attached to n and $n.i.queue$ points to the list of packets in the queue of child i of node n and $n.i.share$ is the share of n's link assigned to i and $V(n)$ tells us the virtual time of node n

```
for (t = Δ, t < 10 secs, t + Δ)
send_packet (Root.node,linkbandwidth);
send_packet (n,l){
  If n ≠ leaf node {
  /* Calculation of Virtual time */    for (i=1,
i ≤ num.child(n), i++){
        if (n.i.queue) has packets backlogged
        sum++ = φ_{n.i};    }
    V(n) = V(n) + ((1/sum) * Δ)
if transmitflag=0 {
for (i=1, i ≤ num.child(n), i++){
if BID of first packet of n.i ≤ min.BID
        min.BID = BID of first packet of n.i
}
length = length of packet with min.BID
transmitflag=1
}
length− = l * Δ;   if length ≤ 0 {
enqueue (first packet of n.i that has min.BID) to
p(n).n.queue
/*enque will also put the new VST, VFT and BID
stamps on the packet*/
dequeue (first packet of n.i from n.i.queue)
transmitflag=0
}
for (i=1, i ≤ num.child(n), i++)
send_packet (n.i, (l * φ_{n.i}/sum));
/*This continues the process recursively till we
reach the leaf node*/ }
if n = leaf node {
        current packet = generate packet /*Checks
the source   to see if any packets have arrived*/
        if current packet != null
enqueue (current packet) to p(n).n.queue;
      }
}
```

IV. Delay Analysis of the H-MPGPS-D scheme

The following result can be proved for a single stand-alone server. The result shows that we can achieve desired flexibility in terms of delay and bandwidth allocation [1].

Theorem 1:

1. **For $\theta_i \geq \phi_i$, the maximum end-to-end delay d_i^k for packet k on session i is**

$$d_i^k \leq \frac{\sigma_i - L_i^k}{\frac{\phi_i}{\sum_{j\epsilon Q}\phi_j}r} + \frac{L_i^k}{\frac{\theta_i}{\sum_{j\epsilon Q}\theta_j}r} + \frac{L^{max}}{r} \tag{2}$$

2. **For $\theta_i < \phi_i$, the maximum end-to-end delay d_i^k for packet k on session i is**

$$d_i^k \leq \frac{\sigma_i}{\frac{\phi_i}{\sum_{j\epsilon Q}\phi_j}r} + \frac{L^{max}}{r} \tag{3}$$

provided the total traffic arrival on all the sessions other than i at GPS server 2 during any time interval of duration $\frac{\sigma_i}{\phi_i}$ is

$$\sum_{j\epsilon Q \Lambda j\neq i} A_j(t, t+\frac{\sigma_i}{\phi_i}) \leq \frac{\sigma_i}{\phi_i}\sum_{j\epsilon Q\Lambda j\neq i}\phi_j \tag{4}$$

where Q is the set of all active sessions, $A_i(t_1, t_2)$ is the amount of traffic arrived on session i in the duration (t_1, t_2), and L^{max} is the size of the largest packet among all the sessions.

Theorem 2:
The worst-case Weighted Fair Index (WFI) for the above scheme is [6]

$$\alpha_i = \phi_i L_{max} + L_{max} + 2L_{max}\frac{\phi_i}{\theta_i} \tag{5}$$

Intuitively the WFI is the maximum amount of time a packet has to wait to receive its fair share service when it comes to an empty queue i. For a hierarchical link, the WFI for a session's packet keeps increasing as it passes through each of its ancestor's nodes. Using Equation 5 and from an analysis of the end to end delay for a hierarchical link, we state the following theorem [6]

Theorem 3:
For a session with H ancestors in an H-MPGPS-D system, if it is constrained by a leaky bucket (σ_i, ρ_i), the delay of any packet in the session is bounded by

$$d_i + \sum_{h=1}^{H-1} \frac{\alpha_{p^h}}{r_{p^h}} \tag{6}$$

where α_{p^h} is the WFI guaranteed to the logical queue node p^h by the server node p^{h+1}, $h = 1,....,H$-1 and d_i is the delay for a single session as defined in Equations 2, 3.

V. Performance Evaluation

Simulations have been performed to verify the performance of this algorithm. The average delay and bandwidh allocation of the scheme has been compared with that of the PGPS and the STFQ [3] schemes for various topologies. Average delay is an important performance parameter for many adaptive or elastic applications like packetized voice and coded video. We have used and modified Keshav's REAL network simulator for these simulations. All the packets finally end up in a single sink. The sink collects data from different sources and sends acknowledgements. Each of the nodes of the hierarchy is assumed to have an infinite buffer. We assume error free network and the link latency to be zero. Various sources such as Telnet, FTP, On-Off etc have been used. Our aim here is to show the kind of flexibility we can achieve with H-MGPS-D that is lacking in other Fair Queueing algorithms.

The H-MPGPS-D scheme is succesful in bringing down the delay bound of sessions which were allocated low bandwidth. The reduction in the delay bound is at times as much as 50% as compared to the delay faced by the low bandwidth packets by the PGPS and the STFQ schemes. At the same time, there has been a slight reduction in the average bandwidth of other sessions but this has been very negligible. The computational complexity for the H-MPGPS-D scheme is the same as that for the PGPS scheme and is of the order of log$N$.

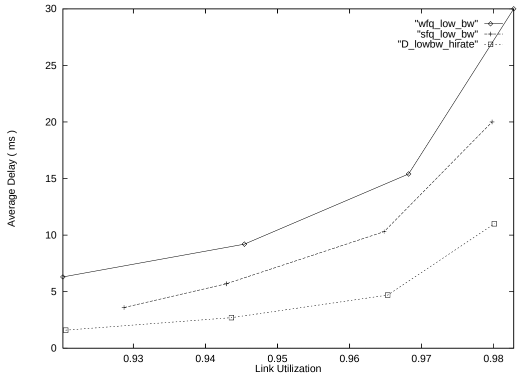

Fig. 2. Average delay for low arrival rate sessions in STFQ, PGPS (WFQ) and MSCFQ-D

Figure 2 shows the comparison of average delay for low rate sessions in STFQ, PGPS (WFQ) and HMGPSD. In HMGPSD, average delay is plotted for the low rate session for which $\theta_i = 20\phi_i$. HMGPSD average delay is even smaller than that of STFQ. In STFQ, we are transmitting all the packets according to their virtual start time. As a result, the advantage given to all the sessions is more or less the same. While in HMGPSD,the sessions with $\theta_i = 20\phi_i$ are given more advantage as compared with other sessions who will be served according to their virtual finish times. In Figure 2, the graph with the lowest delay is for HMGPSD sessions who have been provided $\theta_i = 20\phi_i$. Note that for HMGPSD, the other sessions with $\theta_i = \phi_i$ have their average delay more or less equal to that of PGPS.

In another simulation, we considered the topology shown in Figure 3 Each of the users sends packets which are 8kB long. Each user transmits for a link bandwidth of 0.5Mb. The output link is 1.2Mb and each of the users is given 33users. User 1 and User 3 are Poisson sources while User 2 is an deterministic on-off source with on time 2secs and off time 2 secs. The simulation was run for a total of 8 secs. We simulated for both, the Packetised version of the GPS and the H-MPGPS-D schemes. For the H-MPGPS-D scheme, we tried to improve the delay performance for User 1 by giving it a delay parameter θ of 0.8. We compared the average delay for every 100 ms for User 1 using both the schemes.

In Figure 4, the graph for U1_d is the average delay for the user for the H-MPGPS-D scheme while U1_f is the graph of the average delay for the PGPS scheme. It is apparent that after decoupling, User 1 is having a much lower delay bound. The variation in the delay in both the graphs is due to the on-

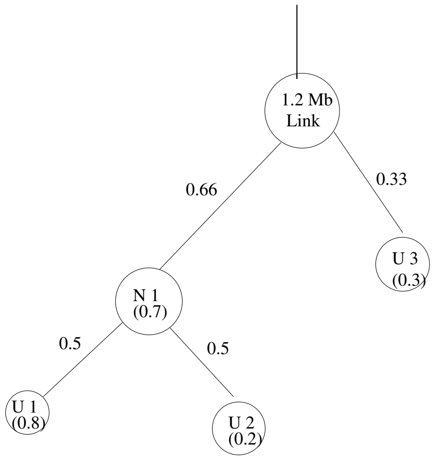

Fig. 3. An example for simulation

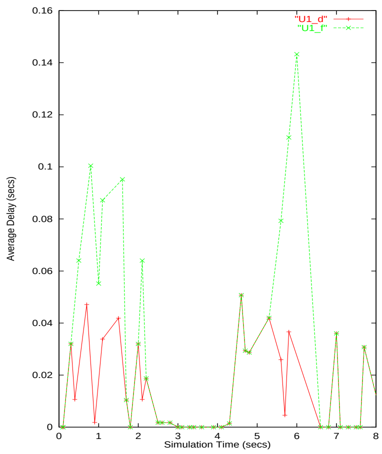

Fig. 4. Average Delay Comparison for PGPS and H-MPGPS-D schemes

off source. Whenever the on-off source becomes on, the output bandwidth of 1.2 Mb starts becoming insufficient and hence the delay starts accumulating. But when User 2 stops sending packets, both User 1 and User 2 experience virtually no delay due to an excess of bandwidth.

The H-MPGPS-D scheme has also been compared with another suggested scheme for decoupling called the HFSC scheme [4]. The H-MPGPS-D scheme ensured nearly the same and at times even a lesser delay for packets that were guaranteed a low bandwidth, the rest of the conditions remaining the same.

REFERENCES

[1] Nirav Parikh, "Decoupling Delay-Bandwidth Guarantee in the Fair Queueing Algorithms", M. Tech. Dissertation, Department of Electrical Engineering, Indian Institute of Technology, Bombay, Jan 1999.

[2] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *IEEE/ACM Transactions on Networking*, Vol 1, No 3, pp. 344-357, June 1993.

[3] P. Goyal, H. Vin, H. Chen, "Start-time fair queuing: A scheduling algorithm for integrated services packet switching networks," in *IEEE/ACM Transactions. on Networking*, Vol 1, no 3, pp. 344-357 October 1997.

[4] I. Stoica, H. Zhang, T.S Eugene, "A Hierarchical Fair Service Curve Algorithm for Link Sharing, Real Time and Priority Services," *Proceedings of SIGCOMM 97*

[5] S. Keshav, *An Engineering Approach to Computer Networking.* Addison Wesley, 1997.

[6] Milind Kopikare, "Hierarchical Link Sharing for Packet Networks", B.Tech Project Report, Department of Electrical Engineering, Indian Institute of Technology, Bombay, April 1999.