A STUDY OF A DYNAMIC SCHEDULING MECHANISM TO GUARANTEE QoS

Rachid Laalaoua - Tülin Atmaca *

Pre zabezpečenie QoS štúdiou mechanizmu dynamického riadenia frontov

1. Introduction

High speed networks are currently expected to carry a wide range of traffic types, including data, voice, broadcast and interactive video. Each of these traffic types requires a different service from the network. The design of networks that provide a good Quality of Service (QoS) to the large variety of expected users is an open and interesting research area.

Network performance is quite sensitive to the queue service discipline implemented at the output trunks of routers and switches. While most current implementations are of the FCFS type, recent works have shown that other disciplines in which the priorities are taken into account such as HoL, Preemptive discipline, WFQ (Weighted Fair Queuing) provide better performance [1][6].

Traditionally, the FCFS discipline is used at each output port of switches or routers. Also space priorities (threshold and push-out mechanisms) are considered to differ loss probabilities but not the response time between different Classes of Service [11]. With emerging Classes of Service, this discipline must change and different classes should be placed in separate queues with single server. With FCFS discipline, there is no particular treatment given to packets from flows that are of higher priority or that are more delay sensitive. In addition, small packets can be queued behind long packets, then FCFS queuing discipline results in a larger average delay per packet than if the shorter packets were transmitted before the longer packets. In general, flows of larger packets get better service.

Hence, priorities may be assigned to the basis of traffic type. For example, WFQ is based on a hypothetical fluid-flow system called the Generalized Processor Sharing (GPS) [12][13]. In GPS, if there are N non-empty queues, the server treats all of the N queues simultaneously at a rate proportional to their “reserved” weights. GPS is hypothetical because it can serve N queues simultaneously.

WFQ has received a lot of attention in the research community. Parekh’s work [16] shows that in the absence of link-sharing, the end-to-end delay bound provided by WFQ [15][16], which is the standard packet approximation algorithm of GPS, is very close to that provided by GPS. While WFQ maintains the bounded delay property of GPS, its fairness property is much weaker than GPS.

KEYWORDS: Queuing, QoS, Dynamic-WFQ, HoL, Markov Chain, Simulation, Class of Service.

* Rachid Laalaoua, PhD Student, Tülin Atmaca, Professor
Institut National des Télécommunications, 9 rue Charles Fourier 91011 Evry Cedex, France,
Tel.: +33-1-60764742, Fax: +33-1-60764780, Email: rachid.laalaoua@int-evry.fr, tulin.atmaca@int-evry.fr
As described in [14], WFQ can introduce substantial inaccuracy in GPS approximation. This inaccuracy significantly affects best-effort traffic management, real-time traffic management, and link-sharing algorithms.

WFQ, also known as the Packet-by-Packet Generalized Processor Sharing, is the most well-known packet approximation algorithm for GPS. In WFQ, when the server is ready to transmit the next packet at time $\tau$, it picks, among all the packets queued in the system at $\tau$, the first packet that would complete service in the corresponding GPS system if no additional packets arrive after $\tau$.

A queuing discipline is nothing more than a means for choosing which packet in the different queues (multi-queues) will be served next [10]. This decision may be based on one or all of the following criteria:
- different Classes of Service.
- queue occupancies.
- length of different packets.

In our work a three Class-of-Service (CoS) system is considered with various traffic types. A network node is modelled by three-queue system, each of which is dedicated to a set of traffic type. Most of the traffic, particularly best effort, remains bursty. Therefore, the aggregation of Markov Modulated sources used in this study attempts to generate realistic burstiness properties.

In this paper, we introduce a new scheduling algorithm that depends on the both first points.

Analytical models of priority queues, both preemptive and nonpreemptive are well developed [18][19], but are restricted to infinite queues. In the literature, the performance of a switch or router that schedules a mixture of real-time, non-real-time and best-effort traffic is studied. The performance measurement of interest for real-time traffic is average delay, and for non-real-time traffic is loss probability.

In this paper, we consider two models. The first is described in section 2 and is represented by three queues with a single server. Three service algorithms are presented: HoL, preemptive discipline and dynamic-WFQ. Let us note that the results are obtained both using simulations and an analytical method based on Markov chain approach, for three disciplines mentioned above. In this section, a second model is presented too to compare the performance obtained by three mechanisms mentioned above (HoL, WFQ and Dynamic-WFQ). However the traffic characteristics are changed as well as the number of sources. In section 3, a complete description is given for this model. So, we present only the simulations results for this model. Finally, some concluding remarks are given in section 4.

2 Priority discipline
2.1 Model Description

The system model is depicted in Figure 1. We consider three queues served by a single server. We assume that the arrival streams are independent Poisson processes with intensity $\lambda_i$ at queue $i$ ($i = 1, 2, 3$). The service time is exponentially distributed with parameter $\mu$.

This model will be studied with three service disciplines (HoL, Preemptive and Dynamic-WFQ).

![Fig. 1 Model to study](image)

2.2 Head-of-Line (HoL)

In this scheduling policy, priority is always given to Class 1 traffic (real-time). Class 2 traffic is served only if there are no queued Class 1 packets. The last Class is served when both queues (queue 1 and queue 2) are empty. A service is not preempted. Within a class, packets are served with FCFS discipline.

2.2.1 Markovian Model

Markov chains are known to be powerful modelling tool for a variety of practical situations. For HoL discipline with three queues, the behaviour of the system described above is represented by a Markov Chain.

The state vector of this Markov chain is given by $\mathbf{n} = (n_1, n_2, n_3, F)$ where $n_i$ is the number of customers at queue $i$, and Flag $F$ takes four values as follows:

$$F = \begin{cases} 
0 & \text{if } n_1 = n_2 = n_3 = 0 \\
1 & \text{if } n_1 > 0 \\
2 & \text{if } n_1 = 0 \text{ and } n_2 > 0 \\
3 & \text{if } n_1 = n_3 = 0 \text{ and } n_2 > 0 
\end{cases}$$

The decision of the server depends on the value of the Flag. So the following algorithm shows how the transitions take place:

1. If (queue1 is not empty)
2. take customer from queue 1
3. else
4. if (queue2 is not empty)
5. take customer from queue 2
6. else
7. if (queue3 is not empty)
8. take customer from queue 3
9. end

After generating $Q$ matrix, we compute the steady state probabilities using Arnoldi’s method [5].
we have:
\[
\begin{align*}
\pi_0 &= 0 \\
\sum \pi_i &= 1
\end{align*}
\]
\(
\pi_j = P(n = j) \) is the probability that the present state is \( j \) and the state vector of these probabilities is \( \pi = (\pi_1; \pi_2; \ldots \pi_N) \).

Let \( j = (n_1; n_2; n_3; F) \) be a state of Markov chain, and the probability that there is \( x \) customers in queue \( i \) is given as follows:

queue 1: \( P_1^1(x) = \sum \pi_j \) such that \( n_1 = x \).
queue 2: \( P_1^2(x) = \sum \pi_j \) such that \( n_2 = x \).
queue 3: \( P_1^3(x) = \sum \pi_j \) such that \( n_3 = x \).

Note that the probability that there is \( x \) customers in queue \( i \) is \( P_i^i(x) \).

2.2.2 Results

This section gives the simulation and analytical results concerning the queue length distribution (probability density function (pdf)) and the loss probability. These parameters depend on the buffer capacity and on the distribution of global traffic. The results are obtained for \( \lambda_1 = 0.27 \) packets/unit time \( \lambda_2 = 0.27 \) packets/unit time; \( \lambda_3 = 0.26 \) packets/unit time. Simulation model is written in C language programming.

Figure 2 shows that Markovian results are close to those obtained by simulation.

It is clear that HoL gives good performance for Class 1.

Figure 3 presents the loss probabilities vs the proportion of traffic types obtained by simulation and by analytical method for real-time and non-real-time traffics. It appears that the loss probability increases when the Class 1 load increases too.

The second curve in Figure 3 shows the loss probability of Class 2 as a function of traffic load (Class 1 and Class 2). We note that the Class 1 has an impact on the Class 2.

2.3 Preemptive discipline

Preemptive discipline gives absolute high priority to traffic of Class 1, because the traffic which has a high priority preempts the customer with low priority in service. Therefore, in HoL discipline, they are served after the customer having lower priority has finished his service. For preemptive discipline, three cases are usually identified:

- Preemptive resume: customer picks up from where he left off.
- Preemptive repeat without resampling: when customer reenters service after having been preempted, he starts with the same total service time that he has lost previously.
- Preemptive repeat with resampling: this case assumes that a new service time is chosen for our reentering customer.

In this study, Preemptive resume is considered. In context of integrated service in Frame Relay, some constructor uses this scheme in the switches. The head of sub-frames that mother frame is preempted, are calculated automatically.

2.3.1 Markovian Model

In this section, we study the model presented in Figure 1 under Preemptive discipline using Markov chain approach. The state vector is given by

\( \eta = (n_1; n_2; n_3; f_1; f_2; f_3) \) where \( n_i \) is the number of customers in queue \( i \), and the value of \( f \in [0, 1] \) select which customer will be served.

Transition rates are given in Figure 4. We compute steady state probabilities of this Markov chain using Arnoldi’s method.

2.3.2 Results

We note that the results obtained by solving of Markov chain are close to those obtained by simulation. These results are shown in Figure 5, and are obtained for the same parameters used previ-
ously. We note also that Class 1 has greater loss when the HoL
discipline are used because we can not preempt the service of
lower priority customers. Therefore, loss probability of Class 2
depends on the proportion of each traffic type. Then, when
proportion of Class 1 or Class 2 traffics increases, losses with
preemptive discipline will be smaller than those with HoL disci-
pline. Table 1 shows that the sejourn time increases when the pro-
portion of traffic corresponding increases.

The results obtained using the simulation and an analytical
method (HoL and Preemptive), have 1 % relative error.

<table>
<thead>
<tr>
<th>Class</th>
<th>Priorities</th>
<th>Traffic Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cl 1</td>
<td>Pr, HoL</td>
<td>1.37, 2.08</td>
</tr>
<tr>
<td>Cl 2</td>
<td>Pr, HoL</td>
<td>2.68, 3.25</td>
</tr>
<tr>
<td>Cl 3</td>
<td>Pr, HoL</td>
<td>8.18, 7.84</td>
</tr>
</tbody>
</table>

Fig. 4 Possible transitions

Fig. 5 Queue length distribution (Preemptive discipline)

Sejourn Time (second) as a function of traffic proportion

<table>
<thead>
<tr>
<th>Traffic Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.27</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>0.4</td>
</tr>
<tr>
<td>0.5</td>
</tr>
<tr>
<td>0.6</td>
</tr>
</tbody>
</table>

2.4 Dynamic-WFQ

From the discussion in the previous sections, it is seen clearly
that the adapting scheduling algorithm based on queue occupan-
cies and using weight for different Classes of Service can be beni-
fical to network performance. Here, we present Dynamic-WFQ
discipline in which we introduce the selection function which plays
an important role for the packet transmission scheduling.

This function computes a number of packets from different
queues that will be transmitted to virtual queue in the next cycle
(Figure 6).

A cycle can be determined as the total time spent to transmit
all packets in the virtual queue. The behaviour of this function
depends on two parameters (queue occupancies and priorities).

When these parameters increase, the function must increase too.
The selection function \( f(c_i) \) is given as follows:

\[
\begin{align*}
  f(c_i) &= w \cdot \alpha_i + (1 - w) \cdot pr_i, \\
  m_i &= \left[ \frac{QSize \cdot f(c_i)}{\sum_{i=1}^{3} f(c_i)} \right],
\end{align*}
\]

such that \( 0 \leq \alpha_i, pr_i \leq 1 \) and \( \sum_{i=1}^{3} pr_i = 1 \).

where \( m_i \) is a packet number of Class \( i \). QSize is the total packets
that will be transmitted next in the virtual queue and \( \alpha_i \) is queue i occupancy.

In this study, the parameters \( w; pr_i \) are taken in the following
values. \( w = 0.4; pr_1 = 0.55; pr_2 = 0.35; pr_3 = 0.1 \).

The packets already in the separate queues wait to be chosen
to transmit (by the selection function) later. This service schedul-
ing can be seen as cyclic service with priority discipline, but the
packet numbers to transmit from each queue can be changed in
each cycle. Packet numbers are calculated by the selection func-
tion.

2.4.1 Markovian Model

The state vector of Markov chain of Dynamic-WFQ is given as
by \((n_1, n_2, n_3, x_1, x_2, x_3)\) where \(n_i\) is the number of customers in
queue \(i\), and \(x_i\) is the number of class \(i\) customers in virtual queue.

The selection function starts when \( \sum_{i=1}^{3} x_i = 1 \). This means that
virtual queue contains a single customer. So the selection function
must compute the number of packets from each separate queues
and transit with \( \mu \) rate to the next state:

\[
(\text{Max}(n_1 - m_1, 0), \text{Max}(n_2 - m_2, 0), \text{Max}(n_3 - m_3, 0), \text{Min}(n_1, m_1), \text{Min}(n_2, m_2), \text{Min}(n_3, m_3))
\]

When all queues are empties, the first packet that arrives,
transits directly to virtual queue.
2.4.2 Results

Given the number of states of related Markov chain increasing as function of buffer capacities, the computation time of the steady state probabilities becomes considerably long. Therefore, we have reduced buffer sizes to $N_1 = N_2 = N_3 = 9$ in order to show that Markov results are close to simulation results. These results are shown in Figure 7.

In order to show the real performance of Dynamic-WFQ algorithm, we compare queue length distribution intended in Figure 8 of HoL and Dynamic-WFQ algorithms. We observe that the buffer occupancies for class 1 (higher priority) is longer in the case of Dynamic-WFQ scheduling than HoL scheduling. On the contrary, for both classes with lower priorities, Dynamic-WFQ provide better performance.

3 Performance Comparison of HoL, WFQ and Dynamic-WFQ

3.1 Model

In multimedia communication, multiple data streams need to be multiplexed on a single transmission channel. Multiplexing of data streams using such simple mechanism may have undesirable results for multimedia real-time traffic. In this section, we analyze and compare three mechanisms with priority.

The assumptions considered for the model under study aim to represent the context of a high speed Wide Area Network (WAN). This implies to take into account non negligible transmission times between the sources, the bottleneck and the destinations.

We consider that in such backbone, the connectivity is reduced, thus we have modelled three groups of numerous sources whose traffics go through three high speed links (622 Mbit/s). Each set of sources includes numerous sources of different types of traffic (voice, videoconference, video, data with QoS, data best effort). The traffic profiles are disrupted by background traffics which transported in the same links but have other destinations.

This configuration depicted in Figure 9, is intended to be a test for the comparison between the three service scheduling disciplines described above, in the presence of different traffic types and different number of sources.

3.2 Traffic Characterization

We have three sets of sources which represent different traffic types and background flows that represent $\frac{2}{3}$ of global traffic. The drawback of this method is that it requires a realistic network simulator, and considerable amount of computing time. However, we approximate the set of background flows by using a server with vacation in multiplexer for each set of sources. Since the parasitic proportion is $\frac{2}{3}$ of all traffic, the idle time of server is $\frac{0.8 \times 2}{3} + 0.2$, and working time is $\frac{0.8 \times 1}{3} (0.8$ is load). These periods have an exponential distribution. Figure 10 shows the probability density function of the number of packets in the multiplexer using real model with classical server, and other method with vacation server using different time. We note that 0.01 second, which represents the sum of idle time and working time, is the best approximation.
In order to understand the effect of different mechanisms, we study a scenario (Figure 9) using a simulator written in C language. As shown in Figure 9, the simulated network consists of three multiplexers with vacation server which are linked to a router. The router is modeled by three queues with a single server using a priority scheduling mechanisms. The following table shows the parameters used in the model.

### Table: Parameters used in the model

<table>
<thead>
<tr>
<th>TYPE</th>
<th>Model</th>
<th>number of sources per set</th>
<th>Average rate per source</th>
<th>% of global Set traffic</th>
<th>Packet size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice</td>
<td>IDP</td>
<td>324</td>
<td>32 kbit/s</td>
<td>5 %</td>
<td>160</td>
</tr>
<tr>
<td>Videoconference</td>
<td>MMDP</td>
<td>50</td>
<td>256 kbit/s</td>
<td>9 %</td>
<td>256</td>
</tr>
<tr>
<td>Video (MPEG2)</td>
<td>MMDP</td>
<td>6</td>
<td>5 Mbit/s</td>
<td>16 %</td>
<td>512</td>
</tr>
<tr>
<td>Data with QoS</td>
<td>ON/OFF</td>
<td>3</td>
<td>9 Mbit/s</td>
<td>16 %</td>
<td>512</td>
</tr>
<tr>
<td>Data Best effort</td>
<td>ON/OFF</td>
<td>12</td>
<td>8 Mbit/s</td>
<td>51 %</td>
<td>1536</td>
</tr>
</tbody>
</table>

#### 3.3 Results

This section gives the simulation results as the queue length distribution and voice end-to-end delay.

It appears that the end-to-end delay for voice traffic using Dynamic-WFQ mechanism is near to delay obtained by WFQ and HoL mechanisms. We remark that the peaks corresponding to 20 s in Figure 11, are due to bursting arrivals. In all case, the end-to-end delay is less than 11 ms. The parameters which are taken in this section are:

$$ pr_1 = 0.6, \quad pr_2 = 0.3, \quad pr_3 = 0.1 \quad \text{and} \quad w = 0.35. $$

On the other hand, Figures 12 and 13 show the comparison of queue 2 and 3 packet distribution using different scheduling mechanisms. In Figure 12 we note that the line graphs which represent the pdf of videoconference traffic using HoL, WFQ and
Dynamic-WFQ, have the similar behaviour. Figure 13 shows the pdf of class 2 and Best-effort traffics, using Dynamic-WFQ and the results are better than those using HoL or WFQ disciplines.

4. Conclusion

In our studies we observed that the priority queuing mechanisms cannot isolate the impact of load between traffic streams. The basic property of assigning the bandwidth first to the high priority traffics help to maintain short delay and delay variance for high priority queues.

With WFQ, the absolute weight are given for different traffic types. When congestion occurs for no real-time traffic, it remains for long time because the real-time traffic have a higher weight. The basic idea in Dynamic-WFQ is to change weight each cycle.

Then, the absolute weight is given to each class \( (w_i \cdot p_r_i) \), and the remain proportion of bandwidth is shared. This sharing depends on queue occupancies. The congestion level in nodes (for no real-time traffics) is reduced.

As described in [6], FCFS scheduling mechanism which is used in most of the transport protocols today is not suitable for supporting the multimedia data streams. The real-time requirements as audio and video traffic, cannot be supported well in FCFS. It is shown in [14] that the inaccuracy introduced by WFQ can (a) significantly increases the delay bound for real-time sessions under hierarchical link-sharing; (b) cause end-to-end feedback algorithms for best-effort traffic to oscillate.

As shown above, Dynamic-WFQ using a simple linear function is performant. In future work, we will focus on an analytical method to obtain the better parameters such a \( w \) and \( p_r_i \).

Reviewed by: I. Baroňak, M. Klimo

Literatúra - References