Scheduling Network Traffic

Thomas Bonald, James Roberts
France Telecom R&D
38-40 rue du general Leclerc
92794 Issy-les-Moulineaux, France
{thomas.bonald,james.roberts}@orange-ftgroup.com

ABSTRACT

We discuss the potential of packet scheduling as a means to control traffic and improve performance for both wired and wireless links. Using simple queuing models that take into account the random nature of traffic, we draw practical conclusions about the expected gains and limits of scheduling.

Keywords

Scheduling, bandwidth sharing, service differentiation.

1. INTRODUCTION

While the Internet has traditionally been developed in the "best effort" spirit, with most traffic control mechanisms implemented in end hosts, growing heterogeneity in the underlying network infrastructures and supported services calls for the introduction of intelligent control schemes inside the network. Such schemes mitigate the dependence of quality of service on the correct implementation of end-to-end protocols. They also provide scope for improved performance and allow the introduction of service differentiation. Packet scheduling plays a key role in realizing these objectives, together with higher-level control mechanisms like admission control, routing and load balancing.

In this paper, we discuss the scope of packet scheduling as a means to control traffic and improve performance. We do not aim to present a complete survey of the abundant literature that has appeared on this subject in recent years. Our objective is rather to highlight simple results derived from queuing theory that provide useful insights into the expected gains and limits of scheduling, taking proper account of the random nature of traffic. Scheduling policies are considered mainly in the setting of an ideal fluid model; we do not address implementation issues like the management of the associated buffer.

An essential role of scheduling is to realize bandwidth sharing between data flows for which there are no stringent requirements on performance at packet level. This issue is addressed in Sections 2 and 3 for isolated links and networks with several potential bottlenecks, respectively. A further important role of scheduling is to preserve the performance of delay sensitive flows when they share links with data flows. Section 4 identifies the conditions in which simple FIFO queuing is insufficient and discusses the performance of priority and fair sharing alternatives. Section 5 concludes the paper.

2. SHARING A BOTTLENECK LINK

Consider a link of capacity C bit/s shared by a random number of data flows. We assume flows arrive as a Poisson process of intensity λ flows/s and have independent random sizes of mean σ bits. The traffic intensity is $\lambda\sigma$ bit/s. Performance depends not only on the link load, defined as the ratio ρ of traffic intensity to link capacity, but also on the way flows share bandwidth. Each sharing policy, possibly enforced by means of packet scheduling, may be viewed as a specific service discipline for the underlying queuing system of load ρ. We review some key sharing schemes and discuss their impact on performance.

2.1 Fair sharing

We first consider fair sharing: the throughput of each flow is equal to C/n in the presence of n flows. The flow-level model then corresponds to the processor-sharing queue. It may represent a link shared by TCP flows with similar round-trip packet delays. Fair sharing may also be enforced by a packet scheduler like the deficit round-robin algorithm [32].

If $\rho < 1$, the number of flows N remains stable. In steady state, it has a geometric distribution of parameter ρ, independently of the flow size distribution [14]. This is true even for non-Poisson flow arrivals, provided flows are generated within sessions consisting of an alternating sequence of flows and idle periods [4]; it is then sufficient to assume that *sessions* arrive as a Poisson process, which is representative of real traffic [28]. Since the mean number of flows is given by

$$\mathrm{E}[N] = \frac{\rho}{1-\rho},$$

we deduce from Little's law the mean flow duration:

$$\tau = \frac{\mathrm{E}[N]}{\lambda} = \frac{\sigma}{C(1-\rho)}.$$

For convenience, we shall rather measure performance in terms of flow throughput, defined as the ratio γ of the mean flow size σ to the mean flow duration τ. We get:

$$\gamma = C(1-\rho). \quad (1)$$

Thus the flow throughput is equal to the link capacity when $\rho = 0$ and decreases linearly in the link load.

In many practical cases, flows do not have full access to the link capacity but are imposed a rate limit, due to the DSL access line or the disk where data are stored for instance. In the simple case of a common rate limit, the corresponding model is a generalised processor-sharing queue

for which similar results can easily be derived [4, 14]. Table 1 below shows the impact of a 20 Mbit/s rate limit on the flow throughput for a 1 Gbit/s link. We observe that the link is virtually transparent when $\rho < 0.9$. The reason is that the link is limiting in the presence of more than 50 flows only, which is a rare event for such load values.

link load	0.5	0.9	0.95	
without rate limit	500	100	50	(Mbit/s)
with rate limit	20	19	16	

Table 1: Impact of rate limit on flow throughput (1 Gbit/s link, 20 Mbit/s rate limit).

If $\rho > 1$, the number of flows grows continuously: the link is saturated. Unlike the stable case, performance is highly sensitive to the flow size distribution. The analysis of the transient regime shows that the number of flows grows in fact very slowly for the heavy-tailed flow size distributions observed in practice [20]. The reason is that most flows are short and thus go quickly through the link while the largest flows that contribute most to traffic accumulate very slowly. In the stationary regime, the number of flows stabilizes thanks to the phenomenon of impatience: some users abandon their transfer because of a too low throughput [11]. The resulting steady-state throughput is then naturally close to the minimum throughput users can tolerate.

Instead of letting the link enter congestion, it would be preferrable to reject some new flows to maintain the throughput of ongoing flows at a satisfactory level [5]. Assume for instance that the total number of flows is limited to m, which guarantees a minimum throughput of C/m. The corresponding blocking probability is then given by:

$$B = \frac{\rho^m}{1 + \rho + \ldots + \rho^m}. \qquad (2)$$

This admission policy is virtually transparent when $\rho < 0.9$, as illustrated by Table 2 for a 1 Gbit/s link with at most $m = 100$ flows, corresponding to a minimum throughput of 10 Mbit/s. The blocking probability is higher in the presence of rate limits, due to a less efficient link utilization. In the limiting case where the minimum throughput is equal to the rate limit, there is no rate adaptation and the blocking probability is simply given by the Erlang formula [17, 24].

link load	0.5	0.9	0.95
without rate limit	$4e{-}31$	$3e{-}6$	$3e{-}4$
with rate limit	$3e{-}21$	$2e{-}4$	$2e{-}3$

Table 2: Blocking probability (1 Gbit/s link, 20 Mbit/s rate limit, $m = 100$ flows).

2.2 Unfair sharing

Now assume bandwidth sharing is unfair. Flows may use different versions of TCP or have different round-trip times for instance. Unfair sharing may also be enforced by a packet scheduler like weighted deficit round-robin to favor some flows. We consider the simple example of two flow classes where flows share bandwidth in proportion to some fixed class-dependent weights w_1, w_2, with $w_1 > w_2$. Specifically, the throughput of a class-1 flow is equal to w_1/w_2 times that of a concurrent class-2 flow. The corresponding model is the discriminatory processor-sharing queue [18].

Consider the stable case $\rho < 1$. Although performance is sensitive to the flow size distribution, simulation results show that this sensitivity is slight for reasonable values of the weights, $w_1/w_2 < 10$ say. Performance results assuming the flow size distribution is exponential are therefore representative of typical behaviour. Since the steady-state distribution of an $M/M/1$ queue is independent of the service discipline, the flow throughput is the same as that obtained under fair sharing.

Now consider the flow throughputs of each *class*, denoted γ_1, γ_2. Their ratio:

$$\frac{\gamma_1}{\gamma_2} = \frac{w_1 + w_2(1-\rho)}{w_2 + w_1(1-\rho)}$$

grows from 1 to w_1/w_2 when ρ grows from 0 to 1 [18]. Thus the flow throughputs differ significantly at high load only, as illustrated by Table 3 for a 1 Gbit/s link with $w_1/w_2 = 2$. This is simply due to the fact that flows of both classes are rarely simultaneously active at low load.

The difference is even less significant when flows do not have a full link access but are limited by some external rate constraint. For the 20 Mbit/s rate limit considered in Table 1, the link is again virtually transparent as long as load is less than 0.9, meaning that the flow throughput of both classes is approximately the same for such load values.

link load	0.5	0.9	0.95	
class 1	560	140	72	(Mbit/s)
class 2	450	79	38	

Table 3: Flow throughput under unfair sharing (1 Gbit/s link, $w_1/w_2 = 2$, equal traffic distribution).

If $\rho > 1$, the number of flows of each class grows continuously. Thus the considered unfair sharing policy is unable to protect class-1 flows from saturation. Both the transient regime and the stationary regime, which depends on user behaviour in overload, are again highly sensitive to the flow size distribution [1].

We conclude that the impact of unfair sharing is slight for reasonable values of the weights, $w_1/w_2 < 10$ say: the only significant difference lies in the ratio of flow throughputs at load close to but less than 1. This is in constrast with the limiting case $w_1/w_2 = \infty$ where class-1 flows have priority over class-2 flows. Again, such priority sharing may arise due to different flow characteristics, like responsive TCP flows vs. unresponsive UDP flows, or to deliberate discriminatory scheduling. Class-1 flow throughput then depends on class-1 load only, denoted ρ_1. In particular, priority sharing is able to protect class-1 flows from saturation when $\rho > 1$ but $\rho_1 < 1$. The price to pay is significant performance degradation for class-2 flows, even at low load [8].

This highlights the disadvantage of differentiating services by means of scheduling: the performance of low priority classes is hardly predictable. Quality of service is better controlled by applying differentiated admission control with class-dependent admission thresholds [5]. Such a scheme allows one to protect class 1 from blocking whenever $\rho_1 < 0.9$, say, while ensuring negligible blocking for both classes as long as $\rho < 0.9$, cf. Table 2. Scheduling should rather be

used to enforce fairness, since the presence of unresponsive UDP flows without rate limit may strongly impact the performance of standard TCP flows.

2.3 Sharing a wireless link

Scheduling plays a key role in wireless systems. It has been a major driver of HDR technology for instance, which consists in scheduling packets in an opportunistic way to take advantage of the inherent "elasticity" of data traffic [6]. The current trend consists in combining frequency and time division multiplexing as in OFDMA systems, possibly on different antennas using MIMO techniques. In all cases, the scheduler determines the way radio resources are shared among users.

The system may again be viewed as a queuing system at flow level. Consider a simple example where the radio resource is time shared and flows belong to one of two classes. Each class is defined by the location of users with respect to the antenna, for example, and is characterized by a feasible rate corresponding to the throughput achieved by a flow when scheduled. We denote the feasible rates by C_1 and C_2. We still denote by σ the common mean flow size. Thus the mean service times of class-1 flows and class-2 flows at the scheduler are equal to σ/C_1 and σ/C_2, respectively. Denoting by f_1 and f_2 the proportions of class-1 and class-2 flow arrivals, with $f_1 + f_2 = 1$, the respective load contributions are given by:

$$\rho_1 = \lambda\sigma\frac{f_1}{C_1}, \quad \rho_2 = \lambda\sigma\frac{f_2}{C_2}.$$

We deduce the total link load:

$$\rho = \rho_1 + \rho_2 = \lambda\sigma\left(\frac{f_1}{C_1} + \frac{f_2}{C_2}\right). \tag{3}$$

The equivalent link capacity, defined as the maximum traffic intensity $\lambda\sigma$ such that the system is stable, is given by:

$$C = \left(\frac{f_1}{C_1} + \frac{f_2}{C_2}\right)^{-1}.$$

It is worth observing that the link capacity is independent of the scheduling policy. The latter only determines the service discipline of the associated queuing system. If all flows are scheduled the same fraction of time, the corresponding model is the processor-sharing queue. With the traffic model of §2.1, the mean number of flows of each class is given by

$$\mathrm{E}[n_1] = \frac{\rho_1}{1-\rho}, \quad \mathrm{E}[n_2] = \frac{\rho_2}{1-\rho},$$

independently of the flow size distribution. From Little's law, we deduce the mean flow duration of each class:

$$\tau_1 = \frac{\mathrm{E}[n_1]}{f_1\lambda} = \frac{\sigma}{C_1(1-\rho)}, \quad \tau_2 = \frac{\mathrm{E}[n_2]}{f_2\lambda} = \frac{\sigma}{C_2(1-\rho)},$$

and the corresponding flow throughputs:

$$\gamma_1 = C_1(1-\rho), \quad \gamma_2 = C_2(1-\rho).$$

Thus the flow throughput is equal to the feasible rate when $\rho = 0$ and decreases linearly in the link load.

On average, the flow throughput is given by:

$$\gamma = \frac{\sigma}{f_1\tau_1 + f_2\tau_2} = \left(\frac{f_1}{\gamma_1} + \frac{f_2}{\gamma_2}\right)^{-1} = C(1-\rho),$$

which coincides with the expression (1) found for wired links. These results are in fact valid for an arbitrary set of flow classes, that may be chosen to represent virtually any radio conditions [9, 12].

Now assume $C_1 < C_2$ for instance, meaning that class-1 flows experience worse radio conditions than class-2 flows. It is then tempting to schedule class-1 flows C_2/C_1 times more often than class-2 flows in order to equalize the throughputs. The corresponding model is the discriminatory processor-sharing queue with weight ratio $w_1/w_2 = C_2/C_1$. Performance is then sensitive to the flow size distribution, especially for high values of the weight ratio w_1/w_2. Using the results from [18], one may verify that for an exponential flow size distribution and similar flow arrival rates, $f_1 \approx f_2$, the improvement of class-1 flow throughput is slight while the degradation of class-2 flow throughput is typically high [9]. The reason is that class-1 flows contribute most to link load, cf. (3). On average, the flow throughput decreases, as illustrated by Table 4.

link load	0.5	0.9	0.95	
fair time shares	910	180	91	(kbit/s)
fair throughput	720	110	57	

Table 4: Impact of scheduling on flow throughput for a wireless link
(feasible rates $C_1 = 10$ Mbit/s, $C_2 = 1$ Mbit/s, $f_1 = f_2$).

We conclude that scheduling is unable to compensate for the spatial heterogeneity inherent to wireless systems, usually referred to as the near-far effect. Time and frequency resources should be equally shared among active flows independently of their radio conditions. This is the allocation that tends to be realized by the proportional fair scheduler of HDR and HSDPA systems [6, 27], see §3.3 below. Regarding IEEE 802.11 systems, the random access protocol turns out rather to equalize the throughputs [19]; some solutions have been proposed to correct this anomaly and share the medium access time in a fair way, see e.g. [33].

For similar reasons, use of schedulers like weighted deficit round-robin to provide service differentiation has very limited impact on performance compared to the effect of radio conditions. It is hardly possible to compensate for the low throughput of a "platinum" flow due to poor radio conditions by penalizing lower priority flows having good conditions. As for wired links, differentiation is significant mainly in overload and is then most effectively realized by strict priority scheduling. However, differentiated admission control again appears as a more effective overload control with the scheduler being left to realize equal radio resource sharing between the admitted flows.

2.4 Size-based scheduling

It is well known that performance may be significantly improved by favoring short flows, especially in the practically interesting case of heavy-tailed flow size distributions where such flows represent a small fraction of the overall traffic volume [3, 15, 29]. While the so-called shortest-remaining-processing-time service discipline is optimal, it can hardly be implemented in practice since it requires knowledge of the remaining size of each flow. Other disciplines like least-attained-service that give priority to those flow having the

lowest transferred volumes are more practical and yield similar performance gains.

The potential performance gains of size-based scheduling are illustrated by Table 5, which compares the flow throughputs obtained with the processor sharing discipline (blind scheduling) and the least-attained-service discipline (size-based scheduling) for an access link of 1 Mbit/s and a Pareto flow size distribution, namely $P(\text{size} > x) \propto 1/x^{\alpha}$ with $\alpha = 1.5$. The arrival process is Poisson. The results are derived from (1) and the formula from Kleinrock [24], respectively. The difference is huge, especially at high load. One may think that, while average performance is improved, big flows suffer with this scheduling policy. Surprisingly enough, this is not the case; for the considered flow size distribution, *all* flows benefit from size-based scheduling [13].

link load	0.5	0.9	0.95	
blind scheduling	500	100	50	(kbit/s)
size-based scheduling	780	580	550	

Table 5: Impact of size-based scheduling on flow throughput
(1 Mbit/s link, Pareto flow size distribution $\alpha = 1.5$).

Another advantage of size-based scheduling is related to its behaviour in overload. Unlike blind sharing, flow throughput remains satisfactory for short flows, that is for most flows in practice. Only the biggest flows, whose duration is long and typically not critical, suffer from congestion. Thus size-based scheduling may be seen as an interesting alternative to admission control.

3. SHARING IN A NETWORK

We now consider a network with several potential bottleneck links. We first review several notions of fair sharing, then highlight the potential capacity losses due to unfair sharing and focus on the specific case of wireless systems.

3.1 Fair sharing

Consider a set of L wired links shared by N flow classes. Let r_i be the route of class-i flows in the network, defined as a subset of the set of links $\{1, \ldots, L\}$. Thus we have $r_1 = \{1\}$, $r_2 = \{2\}$, $r_3 = \{1, 2\}$ for the linear network of Figure 1 and $r_1 = \{1, 4\}$, $r_2 = \{2, 4\}$, $r_3 = \{3, 4\}$ for the concentration tree of Figure 2.

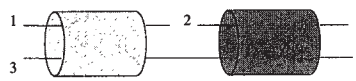

Figure 1: A linear network.

Let n_i be the number of class-i flows. Denoting by C_l the capacity of link l, the throughput φ_i of each class-i flow must satisfy the capacity constraints:

$$\sum_{i:l \in r_i} n_i \varphi_i \leq C_l. \tag{4}$$

It is unclear how network resources should ideally be allocated in this context. Max-min fair sharing consists in allocating bandwidth as equally as possible [7]; it may be realized by implementing a fair queueing algorithm at each link [26]. A larger overall throughput is achieved by proportional fair sharing, which is formally defined as the allocation that maximizes the sum of throughput logarithms,

$$\sum_{i=1}^{N} n_i \log(\varphi_i), \tag{5}$$

under the capacity constraints (4); this sharing is close to that realized by current congestion control algorithms of TCP under FIFO scheduling [21].

Assume class-i flows arrive as a Poisson process of intensity λ_i. Denote by σ_i their mean flow size. The load of link l is given by the ratio of traffic intensity to link capacity:

$$\rho_l = \frac{\sum_{i:l \in r_i} \lambda_i \sigma_i}{C_l}.$$

For both max-min fair and proportional fair sharing, the network is stable if and only if $\rho_l < 1$ for all links l [8]. In the stationary regime, performance may differ quite significantly, as illustrated by Table 6 for the linear network of Figure 1 with equal link capacities and equal loads. These results are derived by simulation for max-min fair sharing, using a common exponential flow size distribution, and analytically for proportional fair sharing [8]. We observe a significant performance improvement with proportional fair sharing.

link load	0.5	0.9	0.95	
max-min fair sharing	300	52	23	(Mbit/s)
proportional fair sharing	450	78	38	

Table 6: Impact of sharing on flow throughput in a linear network
(1 Gbit/s links, $\rho_1 = \rho_2 = \rho_3$).

Again, the difference is attenuated by the presence of rate limits, in which case links are constraining at high load only. For some network topologies like concentration trees, that may represent the successive multiplexing stages of a backhaul network, max-min fair sharing and proportional fair sharing coincide and thus yield exactly the same performance, with or without rate limits.

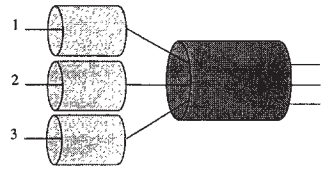

Figure 2: A concentration tree.

3.2 Unfair sharing

The above fairness notions extend to *weighted* max-min fair sharing and *weighted* proportional fair sharing. As for an isolated link, the weights may correspond to either different flow characteristics (e.g. round-trip delay, TCP version) or a discriminatory scheduling policy. Again, the impact on

flow throughput turns out to be insignificant for moderate weight ratios except at load close to but less than 1.

The only significant difference with an isolated link is related to the limiting case where one class, say class 1, has priority over another class, say class 2. Such priority sharing not only impacts class-2 flow throughput, as for an isolated link, but may lead class 2 to saturation at load less than 1. In the linear network of Figure 1 with equal link capacities, assuming for instance that classes 1 and 2 have priority over class 3, the network is stable if and only if $\rho_3 < (1-\rho_1)(1-\rho_2)$. For equal traffic distribution, i.e. $\rho_1 = \rho_2 = \rho_3$, this imposes a maximum link load of $3-\sqrt{5} \approx 0.76$. Such capacity losses may occur for any network topology under priority sharing [8].

Size-based scheduling, that introduces some form of priority for small flows over big flows, may also yield a suboptimal utilization of network resources when applied to several potential bottleneck links [34]. In both cases, the capacity loss is strongly attenuated by the presence of rate limits.

3.3 Sharing wireless links

The difference between max-min fair sharing and proportional fair sharing is exemplified in the case of wireless links. Consider the example of §2.3 with two flow classes. Since a class-i flow has throughput C_i when scheduled, the ratio φ_i/C_i of actual rate to peak rate corresponds to the fraction of time each class-i flow is scheduled. We deduce the throughput region:

$$n_1 \frac{\varphi_1}{C_1} + n_2 \frac{\varphi_2}{C_2} \leq 1. \quad (6)$$

Max-min fair sharing consists in equalizing the throughputs. Proportional fair sharing, on the other hand, equalizes the access time to the scheduler: maximizing (5) under the capacity contraint (6) yields:

$$\frac{\varphi_1}{C_1} = \frac{\varphi_2}{C_2}.$$

Thus Table 4 shows the difference between proportional fair sharing (fair time shares) and max-min fair sharing (fair throughput). Again, it is much more efficient to share the resources according to proportional fairness. This is even more critical in multihop wireless networks that combine the effects of route length (cf. §3.1) and radio conditions (cf. §2.3). The design of practical schemes that achieve proportional fair sharing in a distributed way, using packet scheduling and congestion control algorithms, is still an open issue.

4. SERVICE INTEGRATION

Finally, scheduling plays a key role in the integration of data traffic and delay-sensitive traffic like voice and video streaming. In the following, we refer to these two broad traffic classes as elastic and streaming, respectively. We first give conditions for which FIFO scheduling is sufficient, then discuss the pros and cons of non-FIFO scheduling policies like priority queuing and fair queuing. The case of wireless links is considered separately.

4.1 Scope of FIFO scheduling

For wired links, the need for non-FIFO scheduling clearly depends on link capacity: the transmission of a packet of maximum size, 1500 Bytes say, lasts 12 ms on 1 Mbit/s access links and only 12μs on 1 Gbit/s core or backhaul links. For low-speed links (less than 20 Mbit/s, say), bursts of packets of maximum size lead to unacceptable delays for streaming flows under FIFO scheduling. FIFO scheduling is also not acceptable for high-speed links if flows do not have rate limits. While most streaming flows have intrinsic rate limits, some elastic flows may well be constrained by the considered link only, thanks to optical access lines for instance, and saturate the associated buffer.

FIFO scheduling may be sufficient if all flows have rate limits significantly less than link capacity and share bandwidth in the conditions of so-called bufferless multiplexing: the sum of flow rates exceeds link capacity with negligible probability, less than 10^{-3} say. The number of flows then evolves like the number of customers in an $M/G/\infty$ queue. Specifically, assuming a common rate limit, it has a Poisson distribution of mean ρm, where ρ is the link load and m denotes the link capacity to rate limit ratio. Table 7 below gives the probability that overall traffic exceeds link capacity and thus saturates the associated buffer for various values of m and ρ. Thus if all flows have rate limits less than 1 Mbit/s, the probability of saturating a 1 Gbit/s link is negligible as long as $\rho < 0.9$ (case m=1000). This maximum link load falls to 0.7 and 0.4 for 10 Mbit/s (case $m = 100$) and 100 Mbit/s (case $m = 10$) rate limits, respectively.

link load	0.5	0.9	0.95
$m = 10$	$1e{-2}$	$3e{-1}$	$4e{-1}$
$m = 100$	$1e{-10}$	$1e{-1}$	$3e{-1}$
$m = 1000$	$1e{-22}$	$1e{-3}$	$1e{-1}$

Table 7: Buffer saturation probability (m = link capacity to rate limit ratio).

In the conditions of bufferless multiplexing described above, packet delays and losses due to traffic burstiness at packet timescales turn out to be of secondary importance. The "negligible jitter" conjecture described in [10] suggests that traffic is statistically better than Poisson in the sense that packet delays and losses are less than those obtained with a virtual Poisson stream of packets of maximum size, which can be derived from known results for the $M/D/1$ queue. Thus for 1 Gbit/s links, packet delays exceed 2 ms with probability less than 10^{-7} for an instantaneous load (ratio of traffic to link capacity) as high as 0.95. Packet loss rates are negligible for such load values provided the buffer size is larger than 1 MByte, which is typically the case.

To summarize, FIFO scheduling is sufficient on high-speed links for ensuring low packet delays and negligible loss rates. It must simply be verified that the conditions of bufferless multiplexing are satisfied, which depends both on the link load and the link capacity to rate limit ratio (cf. Table 7). For low-speed links, or to increase the load of high-speed links, it is necessary to apply specific scheduling policies.

4.2 Priority sharing

A natural scheduling policy consists in giving priority to streaming traffic over elastic traffic. While the packet delay and loss rate of streaming flows is minimized, elastic flows may suffer from such priority sharing.

The main source of performance degradation is the occurrence of periods of "local instability" where the traffic intensity of elastic flows temporarily exceeds the residual capacity left by streaming flows [16]. The number of elastic

flows in progress tends to increase in such periods, which corresponds to the transient overload situation described in §2.1. The resulting flow throughput depends on traffic characteristics like the duration of streaming flows and the size of elastic flows [16, 25]. In general, it is much lower than its value (1) derived in the absence of streaming traffic, where ρ denotes the total traffic load, except for low streaming traffic loads or low streaming flow rates [2].

Priority sharing may in fact be unable to guarantee low packet delays and loss rates to streaming traffic if the latter contributes most to traffic, as in the case of TV broadcast on DSL or optical backhaul networks. It may then be necessary to protect voice traffic by handling its packets with priority over those of less delay-sensitive streaming traffic for instance. Such solutions require three or more traffic classes, making its implementation and management much harder and less flexible.

4.3 Fair sharing

In an alternative integration scenario, fair sharing is imposed by means of a specific packet scheduler like the deficit round-robin algorithm. For a link capacity C, this guarantees a service rate of C/n to each flow in the presence of n flows, either elastic or streaming. Unlike priority sharing, there is no need for explicit service differentiation. Flows with rates lower than the fair rate C/n are naturally given priority and experience low packet delays and loss rates. Streaming flows with intrinsic rates higher than the fair rate must adapt, on the other hand, to experience lower packet delays and loss rates. The quality of the streaming audio or video then temporarily suffers, of course.

A significant observation is that adaptive rate streaming traffic preserves stability until the elastic traffic intensity alone exceeds capacity [22]. The reason is that streaming flows have intrinsic durations and thus volumes proportional to their rate: in the limiting case where the elastic traffic intensity alone is close to capacity, the fair rate is very low and streaming traffic vanishes.

In practice, one would seek to dimension a network such that the load of each link is less than 0.9, say. Streaming flows then typically attain their natural rate limit except on rare occasions when there are many flows in progress. In these conditions, it makes little difference to account for the variable volume of streaming flows and a conservative approach would be to suppose all flows are elastic. One may then apply the results of §2.1 to deduce the elastic flow throughput, as well as the distribution of the fair rate that determines the quality of streaming flows.

4.4 Case of wireless links

Recent wireless systems like HDR and HSDPA systems use fast scheduling with short timeslot durations ranging from 0.67 to 2 ms [6, 27]. Thus fair time sharing based on a simple round-robin scheduler should in principle be sufficient to guarantee low delays to streaming traffic. As in the case of wired links, streaming flows must simply adapt their rate in case of excessive delays or losses.

The difficulty comes from the fast fading variations of radio signals due to multipath propagation. Under blind scheduling policies like the round-robin scheduler, packets may be transmitted in very bad radio conditions and be unsuccessfully received by the mobile, causing retransmissions and delays. So-called opportunistic policies, that consist in scheduling packets with respect to the radio conditions of each mobile to avoid such fading holes, turn out to be much more efficient. They must be carefully designed, however, to limit the additional delay due to waiting for favourable radio conditions [30, 31].

For similar reasons, handling the packets of streaming traffic with priority may cause retransmissions and delays due to fast fading variations. Some form of opportunism is required to avoid unsuccessful transmissions and timeslot wastage. Such a trade-off may be achieved by the *weighted* proportional fair scheduler [23]. A careful choice of the elastic to streaming weight ratio may limit the packet delay of streaming flows while avoiding fading holes for both elastic and streaming flows.

5. CONCLUSION

Scheduling is a key traffic control mechanism. As shown in this overview, it should mainly be used to enforce fair bandwidth sharing of wired links and fair time sharing of wireless links. This ensures an efficient utilization of network resources, even in the presence of unresponsive flows, and guarantees low packet delays and loss rates to rate adaptive streaming flows. For wireless links, the scheduling policy should additionally be opportunistic to avoid transmission during fading holes.

The scope for scheduling as a means to introduce service differentiation is much more narrow. Sharing the bandwidth of wired links according to class-dependent weights is largely ineffective in realizing perceptible differences in the throughput performance of flows under normal link loads. Moreover, sharing weights have no impact on performance in the usual case where flow rates are limited by external constraints such as the user's access line. Relative performance differentiation does occur under overload but absolute performance is then typically unsatisfactory for all classes.

Priority scheduling, on the other hand, may be used to protect the high priority class from overload or to control packet delays and loss rates of streaming traffic. However, the performance of low priority classes is then hardly predictable. In the presence of multiple bottleneck links, priority sharing may even lead to significant capacity loss due to the inefficient utilization of network resources, cf. §3.2. This loss is strongly attenuated, however, when high priority flows are subject to rate limits that prevent them from completely monopolizing link capacity.

Size-based scheduling may provide significant throughput gains for elastic traffic going through a bottleneck link, especially at high load. It could be implemented on users' access lines, hot spots or home networks to improve the quality of interactive traffic like Web browsing. Streaming traffic may suffer from such a scheduling policy, however, and require some ad-hoc implementation.

A similar, interesting issue is related to the integration of elastic and streaming traffic on wireless links. Handling the packets of streaming traffic with strict priority without regard to radio conditions may cause retransmissions, resulting in packet delays and timeslot wastage. Usual opportunistic schedulers like the proportional fair scheduler, on the other hand, may add significant delays to hit the peaks of radio conditions and avoid fading holes. The design of schedulers that limit packet delays and loss rates of streaming traffic while being opportunistic is a challenging and largely open issue.

6. REFERENCES

[1] E. Altman, T. Jimenez, D. Kofman, DPS queues with stationary ergodic service times and the performance of TCP in overload, Proc. of INFOCOM 2004.

[2] N. Antunes, C. Fricker, F. Guillemin, P. Robert, Perturbation analysis of a variable M/M/1 queue: A probabilistic approach, Advances in Applied Probability 38-1 (2006) 263–283.

[3] K. Avrachenkov, U. Ayesta, P. Brown, E. Nyberg, Differentiation between short and long TCP flows: Predictability of the response times, Proc. of INFOCOM 2004.

[4] S. Ben Fredj, T. Bonald, A. Proutière, G. Régnié and J.W. Roberts, Statistical bandwidth sharing: A study of congestion at flow level, Proc. of SIGCOMM 2001.

[5] N. Benameur, S. Ben Fredj, S. Oueslati-Boulahia and J.W. Roberts, Quality of Service and flow level admission control in the Internet, Computer Networks 40 (2002) 57-71.

[6] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana and A. Viterbi, CDMA/HDR: A bandwidth-efficient high-speed wireless data service for nomadic users, IEEE Commun. Magazine, July 2000.

[7] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1987.

[8] T. Bonald, L. Massoulié, Impact of fairness on Internet performance, Proc. of SIGMETRICS/Performance 2001.

[9] T. Bonald, A. Proutière, Downlink data channels: User performance and cell dimensioning, Proc. of MOBICOM 2003.

[10] T. Bonald, A. Proutière, J.W. Roberts, Statistical performance guarantees for streaming flows using Expedited Forwarding, Proc. of INFOCOM 2001.

[11] T. Bonald, J. Roberts, Congestion at flow level and the impact of user behavior, Computer Networks 42 (2003) 521–536.

[12] S. Borst, User-level performance of channel-aware scheduling algorithms in wireless data networks, IEEE/ACM Trans. on. Networking 13-1 (2005) 636–647.

[13] P. Brown, Comparing FB and PS policies, Proc. of MAMA Workshop 2006.

[14] J. W. Cohen, The multiple phase service network with generalized processor sharing, Acta Informatica 12 (1979) 245–284.

[15] M. Crovella, B. Frangioso, M. Harchol-Balter, Connection scheduling in Web servers, Proc. of USENIX 1999.

[16] F. Delcoigne, A. Proutière, G. Régnié, Modelling integration of streaming and data traffic, Performance Evaluation 55 (2004) 185–209.

[17] A.K. Erlang, Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges, in: *The life and works of A.K. Erlang*, Eds: E. Brockmeyer, H.L. Halstrom, A. Jensen, 1948. First published in Danish, 1917.

[18] G. Fayolle, I. Mitrani, R. Iasnogorodski, Sharing a processor among many classes, Journal of the ACM 27 (1980) 519–532.

[19] M. Heusse, F. Rousseau, G. Berger-Sabbatel, A. Duda, Performance Anomaly of 802.11b, Proc. of INFOCOM 2003.

[20] A. Jean-Marie, P. Robert, On the transient behavior of the processor sharing queue, Queueing Systems 17 (1994) 129–136.

[21] F.P. Kelly, A. Maulloo and D. Tan, Rate control for communication networks: Shadow prices, proportional fairness and stability, Journal of the Operat. Res. Society 49 (1998).

[22] P. Key, L. Massoulié, A. Bain and F. Kelly, Fair Internet traffic integration: network flow models and analysis, Annals of Telecommunications 59 (2004) 1338–1352.

[23] K. Khawam, D. Kofman, E. Altman, The weighted proportional fair scheduler, Proc. of QShine 2006.

[24] L. Kleinrock, *Queueing Systems, Vol. 2*, John Wiley and Sons, 1976.

[25] R. Litjens, R. Boucherie, Elastic calls in an integrated services network: the greater the call size variability the better the QoS, Performance Evaluation 52 (2003) 193–220.

[26] L. Massoulié, J. Roberts, Bandwidth sharing: Objectives and algorithms, IEEE/ACM Trans. on Networking 10-3 (2002) 320–328.

[27] S. Parkvall, E. Dahlman, P. Frenger, P. Beming, M. Persson, The high speed packet data evolution of WCDMA, Proc. of the 12th IEEE PIMRC, 2001.

[28] V. Paxson, S. Floyd, Difficulties in Simulating the Internet, IEEE/ACM Trans. on Networking 9-4 (2001) 392–403.

[29] I. Rai, E.W. Biersack, G. Urvoy-Keller, Size-based scheduling to improve the performance of short TCP flows. IEEE Network Magazine 2005.

[30] M. Sharif, B. Hassibi, Delay analysis of throughput optimal scheduling in broadcast fading channels, Proc. of INFOCOM 2005

[31] S. Shakkottai, R. Srikant, A. Stoylar, Pathwise optimality of the exponential rule for wireless channels, Advances in Applied Probability 36-4 (2004) 1021–1045.

[32] M. Shreedhar, G. Varghese, Efficient fair queueing using deficit round-robin, IEEE/ACM Trans. on Networking 4-3 (1996) 375–385.

[33] G. Tan, J. Guttag, Time-based fairness improves performance in multi-rate WLANs. Proc. of USENIX 2004.

[34] I.M. Verloop, S.C. Borst, R. Núñez-Queija, Stability of size-based scheduling disciplines in resource-sharing networks, Performance Evaluation 62 (2005) 247–262.