ATS - Advance-Time Scheduling:
A Service Discipline for Quality of Service Provision

Authors:

Edmundo Monteiro, Gon alo Quadros, Fernando Boavida

Affiliation and Address:

Universidade de Coimbra
Departamento de Engenharia Inform tica
P lo II - Pinhal de Marrocos
3030 Coimbra
PORTUGAL
Tel: +351 39 7000000
Fax: +351 39 701266
e-mail: edmundo@dei.uc.pt

Abstract

Transmission scheduling mechanisms, or service disciplines, act on the switching subsystem of intermediate systems (e.g., routers, switches) and are responsible for the transmission scheduling of data units. Service disciplines have influence on the main quality of service parameters, namely on the parameters related to throughput and transit delay.

Conventional first-come-first-served (FCFS) service disciplines cannot, in general, provide quality of service guarantees. Some forms of performance guarantee can be achieved using other types of service disciplines, that warrant a minimum data rate to each packet flow and generally provide bandwidth protection between flows.

In addition to a brief review of service disciplines, this paper presents a new service discipline that implements link sharing as a function of the ratio between the agreed bandwidth share and the available bandwidth, as a way to provide the contracted quality of service. The three major advantages of the proposed service discipline are: the distribution of bandwidth in proportion to the negotiated quality of service throughput; its implementation simplicity; and its ability to schedule flows in advance – that is, even while the transmitter is busy servicing the current flow – which leads to a drastic efficiency improvement of the scheduling operations.

1. Quality of service, traffic control and service disciplines

A considerable effort has recently been put in the study of the support and provision of quality of service for integrated services computer networks [Braden 94, ATMF 94, Campbell 94, ISO 95]. It is expected that a significant part of the traffic in integrated services networks will present some kind of quality of service requirements that cannot be provided on a simple best-effort basis. This is the case, for instance, of real-time applications, that require stringent performance guarantees in terms of throughput, end-to-end delay, and packet losses.

Quality of service provision cannot be achieved without resource reservation which, in turn, must be supported by a set of functions and mechanisms in end systems as well as in intermediate systems. Traffic control functions and mechanisms are key elements in resource management and are used to create and maintain different qualities of service. The model presented in [Braden 94] identifies three main components of traffic control: admission control, classifier, and packet scheduler.

Admission control is responsible for the determination of wether a new packet stream (thus, a whole or part of a new service) can be granted the requested quality of service without degrading the agreed quality of service of existing services. The classifier is responsible for mapping each packet into some quality of service class. The packet scheduler manages the forwarding of the different packet streams, according to a given *service discipline* (that establishes the scheduling algorithm to be used). By reordering of the output queue, the packet scheduler controls link sharing according to the resource reservation policy enforced by the admission control functions.

Service disciplines are, thus, a key element in the provision of an agreed quality of service. When quality of service requirements exist, various types of service disciplines are relevant and can be used as a means to support link sharing: Although there is no unique scheme for the classification of service disciplines, some not-mutually-exclusive categories are generally found in the literature [Aras 94]: rate-based service disciplines, scheduler-based service disciplines, static/dynamic service disciplines, preemptive/non-preemptive service disciplines, etc.

Section 2 of the present paper gives a brief overview and characterization of a significant number of service disciplines. Section 3 introduces the proposed service discipline (Advance-Time Scheduling, ATS), compares it with the presented service disciplines and highlights some of its properties and advantages. Section 4 summarizes the presented work and identifies future and on-going work.

2. Brief overview of main service disciplines

Resource reservation and traffic control mechanisms must be complemented by transmission scheduling mechanisms in intermediate systems, as a way to guarantee the agreed end-to-end quality of service. As alternatives to the FCFS service discipline widely deployed in intermediate switches, several service disciplines can be used. In order to support the presentation and characterisation of the proposed service discipline, this section extends the comparison presented in [ZhangHui 91] and offers a brief overview of the following service disciplines: *delay earliest due date*, *jitter earliest due date*, *hierarchical round robin*, *fair queuing*, *virtual clock*, *dynamic time windows*, *stop-and-go queuing*, *counter based control*, *rate-controlled static-priority*, *deficit round-robin*, *leave-in-time*, and *packet generalised processor sharing*.

For the purpose of exposition and discussion, the concept of *flow* [ZhangLixia 90] is used: a flow is a stream of protocol data units which traverse the same route through the network, between end users, and require the same grade of service at each intermediate system along the path. This concept accomodates different types of communication: ATM virtual circuits, X.25 virtual calls, a sequence of datagrams identified by a flow label, etc.

Delay earliest due date (delay-EDD)

The delay-EDD service discipline was proposed by Domenico Ferrari and Dinesh Verma [Ferrari 90]; it is inspired in the earliest due date (EDD) service discipline of the simulation theory, which assigns a deadline to each client and serves the clients by ascending order of deadlines [Panwalkar 76]. In the case of the delay-EDD service discipline, the deadline for each packet is established by the sum of the expected time of arrival of the packet (according to the negotiated average and peak transmission rates) with the average service delay in the intermediate system.

Jitter earliest due date (jitter-EDD)

The jitter-EDD service discipline was proposed by Ferrari and Verma [Verma 91, Ferrari 93]; it is an extension of the delay-EDD service discipline, in order to guarantee a lower limit to the delay suffered by the packets traversing the intermediate system. Each packet is stamped with the difference between its deadline and the service instant; in the next intermediate system, a regulating mechanism holds the packet by the amount determined by its time stamp before the packet is passed to the scheduler. In this way, the delay experienced by the packet stays between two known limits, which is extremely important for jitter-sensitive services like voice or full-motion video.

Hierarchical round robin (HRR)

The HRR service discipline was proposed by Charles Kalmanek, Hemant Kanakia and Sirinivasan Keshav [Kalmanek 90]. It is based on the definition of several hierarchical service levels, each of which is serviced in round robin during a pre-determined number of slots. The higher the hierarchical level the greater the number of slots assigned to that level and, thus, the greater the bandwidth assigned to the flows belonging to the level. As all hierarchical levels are serviced during each cycle of the round-robin, the HRR service discipline guarantees limited delays, according to the position of flow in the service hierarchy.

Fair queuing (FQ)

The fair queuing service discipline was initially proposed by John Nagle [Nagle 85, Nagle 87], and has suffered subsequent modifications and refinements by various authors [Demers 89, Davin 90, Mckenney 90, Keshav 91a, Keshav 91c]. This service discipline is based on a conceptual bit-by-bit round robin (BR). The BR scheme is simulated by calculating the time when a packet would have left the intermediate system using the BR algorithm and inserting that packet into a queue sorted by departure time. In [ZhangHui 91] a variation of the FQ service discipline is proposed, called *weighted fair queuing* (WFQ). The WFQ discipline enables the assignment of different bandwidths to flows, by the introduction of a factor that determines the number of bits that should conceptually be serviced in each round-robin cycle.

Virtual clock (VC)

The VC service discipline is equivalent to the weighted fair queuing discipline, and it was proposed by Lixia Zhang [ZhangLixia 90] as the service discipline to be used in conjunction with the RSVP protocol [Braden 94b]. As in the FQ discipline, each packet is marked with a Òvirtual timeÓ corresponding to its final time slice if it were transmitted bit-by-bit using time division multiplexing. Then, the packets are transmitted by ascending order of virtual time. The virtual time is determined using the negotiated average end-to-end transmission rate, taking into account the needs of applications and the resource availability.

Dynamic time windows (DTW)

The DTW service discipline was proposed by Faber, Landweber and Mukherjee [Faber 92], and it consists in a generalisation of the virtual clock service discipline. As in the VC service discipline, DTW is based on the existence of an end-to-end service contract, that specifies the peak transmission rate in addition to the average transmission rate. The time window is used for the calculation and imposition of the average transmission rate. During the time window it is possible to accommodate traffic bursts as long as they do not exceed the agreed peak transmission rate and as long as the average transmission rate in the time window is respected. The duration of the time window is dynamically adjusted as a function of the traffic impulsiveness. The DTW discipline has the advantage of preserving traffic bursts, as opposed to the VC and FQ service disciplines.

Stop-and-go queuing (SG)

The stop-and-go service discipline was proposed by Jamaloddin Golestani, of Bell Communication Research [Golestani 90, Golestani 91a, Golestani 91b, Trajkovic 92]. The main goal of this service discipline is the preservation of the traffic smoothness during the network crossing and, thus, to avoid the formation of traffic pulses that is typical in some service disciplines. As in EDD-based and VC-based service disciplines, the end-to-end transmission rate is determined at the time of the flow establishment, taking into account the available network resources. In the case of the stop-and-go service discipline, time is divided in fixed duration time frames. All the packets that arrive in a given time frame are always transmitted in the next time frame, in order to avoid packet grouping and to preserve traffic smoothness. Also, transit delay variation is limited by the duration of the time frame. In order to allow the support of services with varying delay requirements, Golestani proposed a variation of the stop-and-go service discipline that uses time frames of distinct duration.

Counter based control (CBC)

This service discipline was proposed by Imrich Chlamtac and Tao Zhang [Chlamtac 93] for ATM networks, and can be looked at as a variation of the stop-and-go service discipline discussed above. Call admission control functions assign a class of service to each virtual circuit. Each class of service corresponds to a different time frame made up of an integral number of ATM cells. For each virtual circuit, only one cell per time frame can be transmitted. If subsequent cells are generated in that time frame they must wait for transmission in subsequent time frames. Service classes are serviced according to their priority. In a given service class, virtual circuits are serviced using the FCFS discipline. If the peak rate of the virtual circuits is observed this service discipline guarantees limited jitter and buffer space.

Rate-Controlled Static-Priority (RCSP) queuing

RCSP was proposed by Hui Zhang and Domenico Ferrari [ZhangHui 93] for connection-oriented packet-switching networks. This service discipline separates rate-control and delay-control functions by using two stage servers: the first stage is made up of *n* rate controllers (*n* being the number of connections); the second stage consists of a static priority scheduler with several priority queues. The rate controllers shape the input traffic of the corresponding connections according to the agreed QoS contract; this is achieved by assigning each packet an eligibility time and holding that packet till that time. The scheduler chooses packets in FCFS order from each static priority queue, starting with the first non-empty highest priority queue. The authors show that RCSP provides throughput, delay and delay-jitter guarantees, if used in conjunction with connection admission control functions. Although the base algorithm is non-work-conserving, a work conserving version is also proposed in [ZhangHui 93].

Deficit round-robin (DDR)

M. Shreedhar and George Varghese proposed a simplification of fair queuing called deficit round-robin (DDR) [Shreedhar 95, Shreedhar 96]. With DDR bandwidth is shared equally among active flows with a modified round-robin service witch takes into account the influence of different packet sizes. The main advantage of DDR is its implementation simplicity, compared to the original FQ, since DDR requires only complexity O(1) to process a packet. Like the original FQ, DDR canÕt provide different bandwidth shares to active flows and, therefore, can only be used in best effort networks, where users are all treated equally.

Leave-in-time (LiT)

The LiT service discipline was proposed by Norival Figueira and Joseph Pasquale [Figueira 95]. LiT can be regarded as a merge between virtual clock and jitter-EDD, exploiting the good properties of each. LiT has a dual operation mode: as in J-EDD, the discipline can operate in a non work conserving mode providing jitter upper bound for jitter sensitive traffic, and; as in VC, the discipline can operate in a work conserving mode for normal data traffic. Like PGPS (see below) the authors studied the operation of LiT together with leaky bucket in end systems and showed that upper bound delay, jitter and buffer space is guaranteed.

Packet generalised processor sharing (PGPS)

Abhay Parekh and Robert Gallager studied a new and deep approach to weighted fair queuing called packet generalised processor sharing (PGPS) [Parekh 93, Parekh 94]. In their work Parekh and Gallager showed that WFQ used in conjunction with leaky bucket in end systems as admission control mechanism, provides bounded delay and jitter in end-to-end communication.

Table 1, below, summarises the main characteristics of the presented service disciplines. In [ZhangHui 91] a comparative study of some of the presented service disciplines is made (VC, FQ, D-EDD, SG, HRR and J-EDD). In [Trajkovic 92] the Stop-and-Go service discipline is discussed and very briefly compared with the following service disciplines: VC, FQ, D-EDD, J-EDD and HRR. The study highlights the advantage of limited delay jitter, which is only achieved by the jitter-EDD and CBC service disciplines, that have higher implementation complexity. In [ZhangHui 93] the RCSP service discipline is compared with the HRR, SG and J-EDD service disciplines. A more recent study by aglan Aras et al. [Aras 94] compares several service disciplines: D-EDD, J-EDD, SRT (smallest response time), PCT (premptive cut through), and RCSP (rate-controlled static priority).

Table 1 - Summary of service discipline characteristics

	VC	FQ	DTW	D-EDD	SG	HRR	J-EDD	CBC	DRR	LiT	PGPS	RCSP
Fixed time ? (1)	no	no	no	no	yes	yes	yes	no	no	yes/ no (4)	no	yes/ no (6)
Min. throughput guarantee ?	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Maximum delay guarantee ?	no	no	no	yes	yes	yes	yes	yes	no	yes	yes	yes
Limited jitter guarantee ?	no	no	no	no	yes	no	yes	yes	no	yes/ no (4)	yes	yes
Limited buffers ?	no	no	no	no	yes	yes	yes	yes	no	yes	yes	yes
Protection ? (2)	yes	yes	yes	yes	no	yes	yes	yes	yes	yes	yes	yes
Different througput ? (3)	yes	no(5)	yes	yes	yes	yes	yes	yes	no	yes	yes	yes
Main Reference	[Zhang Lixia 90]	[Nagle 85]	[Faber 92]	[Ferrari 90]	[Goles-tani 90]	[Kalma-nek 90]	[Verma 91]	[Chlam-tac 93]	[Shree-dhar 95]	[Figuei-ra 95]	[Parekh 93]	[Zhang Hui 93]

Notes:

1 - does the transmission occur in a pre-determined and fixed instant of time, even though the transmitter is idle (non work conservation)?
2 - does the service discipline provide bandwidth/resource protection between flows?
3- can the service discipline provide different bandwidth share according to flow needs ?
4 - LiT as dual mode operation (work conserving and non work conserving)
5 - except for weigthed fair queuing (WFQ)
6 - “no” in the case of the work-conserving version of RCSP.

Although the characteristics presented in Table 1 are important for the general categorisation of service disciplines, this table hides some key features of service disciplines. For instance, it is important that service disciplines provide a minimum throughput guarantee, but it is also very important the way in which service disciplines distribute bandwidth in the presence of an overload; it is very important to guarantee protection among different flows, but it may also be important to enable a flow to temporarily over-consume bandwitdth if it is available. Moreover, it is important to reduce the algorithmic complexity of scheduling activities, but it is also very important to be able to take scheduling decisions as soon as possible in order to decrease or eliminate the delay introduced by the scheduling operations. The proposed service discipline – Advance-Time Scheduling, ATS – addresses these issues, and is presented in the next section.

3. Advance-Time Scheduling service discipline

The selection, by the scheduling mechanisms, of the next logical flow to be serviced is determined by protocol factors and quality of service factors. Protocol factors determine the service availability, based on the state of the protocol machine and/or on flow control mechanisms. Given all the logical flows that can be serviced, quality of service factors determine the flow to be serviced, at each scheduling instant. Flows should be serviced taking into account the negotiated values for the quality of service parameters, in order to reduce deviation from these values.

The ATS service discipline is derived from a previous proposal [Monteiro 93] which bases the scheduling decision on the *utilisation index*, *Iu(t)*, of each flow, defined as the quotient between the effective throughput and the negotiated (QoS) throughout, as in expression 1, below.

$$Iu(t_k)_{flow} = \frac{throughput_{effective}}{throughput_{QoS}} = \frac{\frac{1}{t_k} \cdot \sum_{i=1}^{k} packet\_length_i}{throughput_{QoS}} \qquad (1)$$

The utilisation index is calculated at the scheduling instants t_k (measured in relation to the time of establishment of flows), for each of the active flows. The flow to be serviced at instant t_k is scheduled by ascending order of utilisation index. An utilisation index less than 1 means that the flow throughput is below the negotiated throughput. An utilisation index greater than 1 implies a bandwidth over-consumption.

Expression 2 enables the updating of the utilisation index at instant t_{k+1} , based on the utilisation index at the previous scheduling instant, t_k . This expression defines the initial conditions, $Iu(t_0)$, and the way for updating the utilisation index for serviced and non-serviced flows. In this expression, l_k is the length, in bits, of the transmitted packet and d_{QoS} is the negotiated value for the flow throughput.

$$Iu(t_{k+1})_{flow} = \begin{cases} 1 & for \quad t_k = t_0 \\ \frac{t_k}{t_{k+1}} \cdot Iu(t_k) & if \quad flow\ inactive \\ \frac{t_k}{t_{k+1}} \cdot Iu(t_k) + \frac{l_k}{t_{k+1} \cdot d_{QoS}} & if \quad flow\ active \end{cases} \quad (2)$$

The main obstacle to the practical use of expression 2 is the accounting of the instants t_k and t_{k+1} for each of the active flows. In addition, in the expression for the updating of $Iu(t_{k+1})$ for the active flows, the first element (the historic portion) has a growing importance as time elapses, when compared to the second element of that expression (the current portion), which may lead to asymmetries in the presence of traffic of irregular nature. Consider, for instance, the case of a flow that stays inactive for a long period of time; for this flow, the utilisation index approaches zero, which enables the flow to almost monopolise the transmission link when it generates traffic (as a way to recover from the unused portion of the agreed bandwidth). Thus, it is evident that the historic portion should have a limited time range, in order to reduce the influence of old traffic characteristics and take scheduling decisions based on recent traffic evolution.

$$Iu(t_{k+1})_{flow} = \begin{cases} 1 & for \quad t_k = t_0 \\ \frac{t_h}{t_h + t_{tx}} \cdot Iu(t_k) & if \quad flow\ inactive \\ \frac{t_h}{t_h + t_{tx}} \cdot Iu(t_k) + \frac{l_k}{(t_h + t_{tx}) \cdot d_{QoS}} & if \quad flow\ active \end{cases} \quad (3)$$

In order to overcome these problems, one can use expression 3 for updating of the utilisation index at instant t_{k+1} , based on the utilisation index at the previous scheduling instant, t_k. In this expression, the historic portion is limited to a fixed value – t_h, *historic time* – and the next scheduling instant is determined by the duration of the transmissions ($t_{k+1} = t_h + t_{tx}$). This latter fact makes it unnecessary to keep record and to account for the lifetime of each of the active flows, because the next scheduling instant is exclusively determined by the transmission time of the present packet. This simplification is equivalent to consider that time stops in the absence of transmissions, which does not affect the scheduling decision as the utilisation indexes retain the value of the last scheduling instant.

The historic time, t_h, should be determined by a number of factors, which include the timescales of the communication system, the average number of active flows, and the traffic characteristics. As a general rule, it should be greater than the average packet transmission time by one or two orders of magnitude, which corresponds to an historic portion that spans a time of ten to one hundred packets.

Figure 1 illustrates the use of the proposed service discipline, for which the scheduling decisions are made on the basis of the utilisation index. This example shows the evolution of the utilisation indexes of four flows with dissimilar QoS throughputs (10, 20, 30 and 40 Mbps), each of which ÒrandomlyÓ transmits packets of 10, 20, 30 and 40 kbits of length. The link transmission rate is 100 Mbps. The graphic shows that the flow that is selected for transmission at each scheduling instant is the one that has the lower utilisation index at that instant of time (as results from the proposed scheduling algorithm). In the period of five mili-seconds represented in figure 1, flows 1, 2, 3 and 4 transmitted 50, 100, 160 and 190 kbits, respectively, which corresponds to throughputs of 10, 20 32 and 38 Mbps.

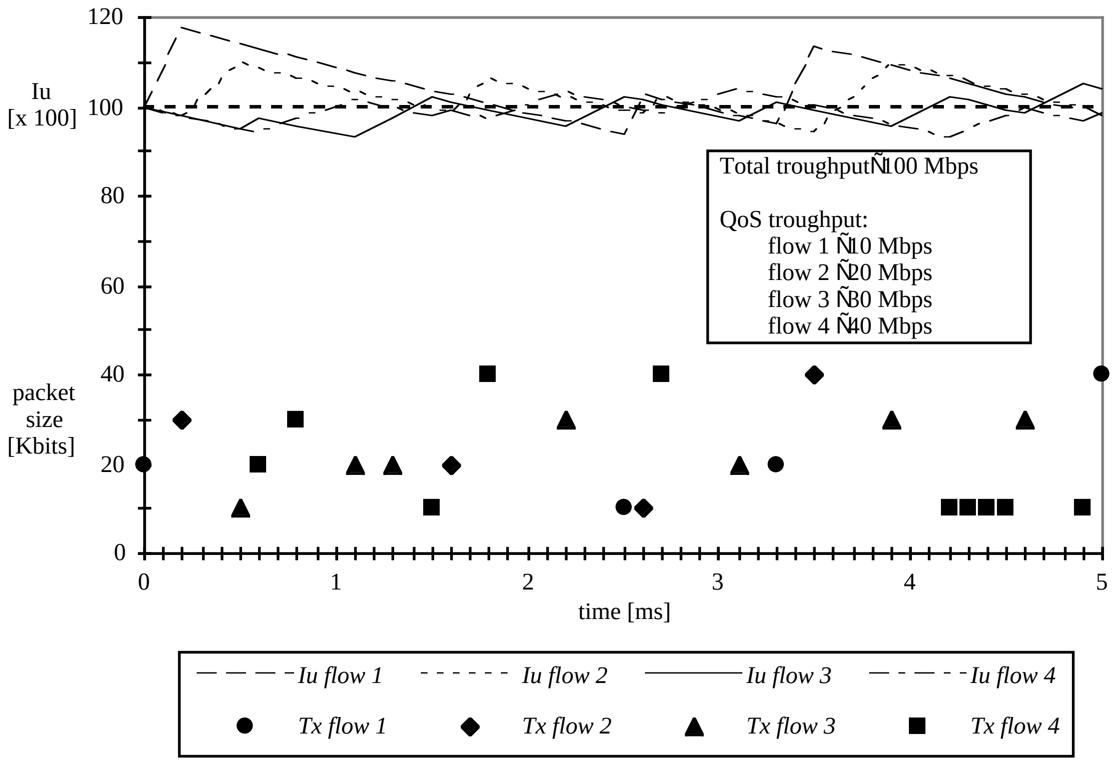

Figure 1 - Example of transmission scheduling based on the proposed service discipline

In the previous section we have briefly presented some recent service disciplines and we compared them with the help of Table 1. Table 2 extends the analysis to the service discipline proposed in this paper. It is important to note that the ATS service discipline is only concerned with transmission scheduling and it is not concerned with the provision of strict delay and buffer needs. These needs can only be accomodated in conjunction with call admission control and flow control functions, that are outside of the scope of the ATS service discipline.

Table 2 - Main characteristics of the proposed service discipline

Characteristic	ATS	Discussion
Fixed time ?	No	The service discipline doesnÕ schedule transmission for pre-determined instants of time; the transmitter is never stopped as long as there are packets for transmission.
Minimum throughput guarantee ?	Yes	If the control and call admission control functions do not over-reserve bandwidth, the discipline guarantees the agreed QoS throughput.
Maximum delay guarantee ?	Yes (1)	The delay is always less than or equal to the delay imposed by the intermediate system maximum queue length plus the delay in lower layers. The imposition of a maximum queue length requires the use of flow control mechaninsms.
Limited jitter guarantee ?	Yes (1)	The delay is always greater than a minimum value (the intermediate system forwarding time + delay in lower layers) and a maximum value (defined above).
Limited buffers ?	Yes (1)	If used in conjunction with flow control mechanisms, the service discipline guarantees limited and constant buffer needs.
Protection ?	Yes	If there is not an over-reservation of bandwidth the discipline guarantees throughput protection between flows, with the exception of small oscillations due to variations in packet lengths.
Different througput ?	Yes	The service discipline can provide different throughput to different flows, according to their needs.

Note:
1 - The guarantee of a maximum delay, limited jitter and limited buffers requires the use of flow control mechanisms, which are outside of the scope of the ATS service discipline.

In addition to the characteristics presented in table 2, the proposed discipline has the following properties:

bandwidth sharing - the available bandwidth is shared between the active flows in proportion to the negotiated QoS throughput; the bandwidth that is not reserved or that was not used by the flows is dynamically distributed among the active flows which can, then, profit from a throughput greater than the negotiated throughput without stealing bandwidth from other flows;

influence of the historic portion - the proposed discipline takes into account the past bandwidth use by the flows, in order to take scheduling decisions; this enables the imposition of a medium-term value for the QoS throughput without disabling short traffic bursts; of all the service disciplines presented in section 2, only the *dynamic time windows* (DTW) and the *counter based clock* (CBC) disciplines allow some form of scheduling memory;

advance-time scheduling - this property gives the name to the service discipline, as it is one of its key features: the computation of the utilisation index can be made in parallel with the transmission of packets, in order to eliminate further delays; thus, during the transmission of packet *k*, the utilisation indexes of the active flows can be updated for instant $t_{k+1} = t_h + t_{tx}$, and the scheduler can pre-select the flow with lower utilisation index among the flows that are in condition to be serviced; pre-scheduling, or advance-time scheduling, is an exclusive characteristic of the ATS service discipline, and directly derives from the form used to take the scheduling decision (see expression 3); Figure 2 compares advance-time scheduling with normal scheduling;

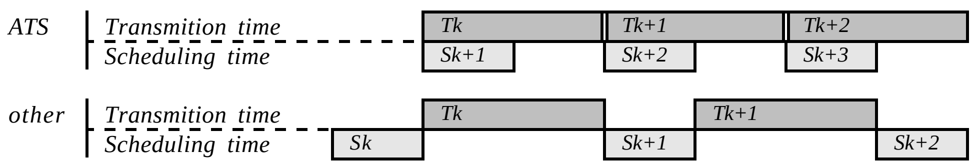

Figure 2 - Advanced-time scheduling

ease of implementation - the simplification introduced in expression 3 doesnÕ require complex computations in order to calculate the utilisation indexes; the selection of the flow with the lowest utilisation index is the most complex operation, with a worst case algorithmic complexity of *O(n)*, *n* being the number of active flows in each link; nevertheless, it is important to note that the complexity of the scheduling operation is not a critical factor in the case of the ATS service discipline, as the scheduling decisions are taken in paralel with packet transmission. Figure 3 presents the pseudocode for the proposed service discipline.

```
algorithm advanced_scheduling (var p_select: pointer to flow);
{activated when transmitter stops or, if no data pending, when a packet arrives}
{p_select points to the flow scheduled to be served}

local variables
        exists_packet: boolean;
        packet_size, tk_plus_one, t_tx, lower_iu: integer;
        p_next, p_lower: pointer to flow;

begin
        {serve the flow scheduled previously}
        "evaluate packet_size of packet to be transmitted";        {packets can be reassembled here}
        "starts packet transmission";

        {calculate the transmission finishing time}
        t_tx <- packet_size / total_throughput;
        tk_plus_one <- t_hist + t_tx;                   {time when  new transmission can take place}

        {update utilisation indexes and schedule next flow to be served}
        exists_packet <- false;
        p_next <- p_select;
        {update utilisation index of  the served flow}
        p_next.iu <- round (p_next.iu * (t_hist / tk_plus_one) +
                                        100 * packet_size / (tk_plus_one * p_next.qos_throughput));
        lower_iu <- p_next.iu;                          {first value for lower utilisation index}
        p_lower <- p_next;
        if      p_next.packet_counter > 0 then          {packets pending in the queue}
                exists_packet <- true;
        end if;
        p_next <- p_next.next_flow;                     {move to next flow }
        while p_next ≠ p_select  do                     {flows are connected together in a circular queue}
                p_next.iu <- round ((p_next.iu * (t_hist / tk_plus_one);    {update Iu}
                if      p_next.packet_counter > 0  and          {packets pending in the queue}
                        p_next.iu < lower_iu            {new  lower utilisation index}
                then
                        lower_iu <- p_next.iu;
                        p_lower <- p_next;              {new  flow selected}
                        exists_packet <- true;
                end if;
                p_next <- p_next.next_flow;             {move to next flow }
        end while;

        {return pointer to next flow to be served}
        if not  exists_packet then  p_select <- nil
        else    p_select <- p_lower   end if;
end
```

Figure 3 - Pseudocode for the advance scheduling service discipline

The algorithm starts by servicing the flow that was scheduled at the previous scheduling instant. After the transmission is started, and in paralel with it, the utilisation indexes are updated and the scheduling decision for the next transmission is made. In order to keep the processing needs to a minimum, the utilization index is represented as an integer percentage value, as can be seen in the expression for the updating of the utilisation index of the served flow, in figure 3. Figure 4 presents the data structures used by the algorithm of figure 3.

```
data structures

        buffer = record of      {buffers to store a packet, or part of it}
                packet_lenght: integer;                             {space used}
                packet_store: table [1 .. size_buffer] of bytes;    {place to store}
                last_buffer: boolean;                               {last buffer of a packet ?}
                next_buffer: pointer to buffer;                     {pointer to next buffer}
        end record;

        flow = record of        {flow description}
                flow_id: ?;                         {flow identification}
                iu: integer;                        {utilisation index}
                qos_throughput: integer;            {qos throughput}
                packet_counter: integer;            {counter of packets pending}
                buffer_counter: integer;            {counter of buffers used}
                point_insert: pointer to buffer;    {pointer to the end of packet queue}
                point_remove: pointer to buffer;    {pointer to the begin of packet queue}
                next_flow: pointer to flow;         {pointer to next flow}
                …                                   {other variables needed for the flow}
        end record;
end
```

Figure 4 - Data structures for the algorithm of figure 3

4. Conclusion

This paper presented a new service discipline, called Advance-Time Scheduling (ATS), that in addition to the provision of throughput and protection guarantees, has a low implementation complexity and enables scheduling decisions to be taken in advance, in parallel with the transmission of packets.

At present, the authors are carrying out additional work with the intent to further study the proposed service discipline characteristics, using both analytical and simulation tools. Another important issue to investigate is the extension of the service discipline decision mechanism in order for it to use a general metric of quality of service instead of being solely based on a relative throughput metric.

The authors believe that pre-scheduling, or advance-time scheduling, is a key feature of the proposed service discipline, that can reduce or even eliminate the delay introduced by scheduling operations. But, with no doubt, the main feature of the ATS service discipline is the fact that scheduling decisions are based on a relative measure of quality of service: the ratio between the effective throughput and the negotiated throughput for each flow. This feature opens the door to a new class of service disciplines, that can act on the basis of a more elaborate Ñ and, possibly, more generalised Ñ quality of service metric.

References

[Aras 94] - aglan M. Aras, James F. Kurose, Douglas S. Reeves, Henning Schulzrinne. ÒReal-Time Communication in Packet-Switched NetworksÓ, *Proceedings of the IEEE*, Vol. 82(1), pp. 122-139, January 1994.

[ATMF 94] - ÒATM User Network Interface Specification, v3.1Ó *ATM Forum*, September 1994.

[Braden 94] - Robert Braden, David Clark and Scott Shenker. *Integrated Services in the Internet Architecture: an Overview.* RFC 1633, IETF Network Working Group, June 1994.

[Braden 94b] - Robert Braden, Lixia Zhang. ÒRSVP: A Resource ReServation ProtocolÓ *ConneXions*. August 1994.

[Campbell 94] - Andrew Campbell, Geoff Coulson and David Hutchison. ÒQuality of Service FrameworkÓ *Computer Communications Review.* Volume 24(2), pp. 6-27, April 1994.

[Chlamtac 93] - Imrich Chlamtac and Tao Zhang. ÒA counter based congestion control (CBC) for ATM networksÓ *Computer Networks and ISDN Systems,* Volume 25, pp. 5-27, 1993.

[Davin 90] - James Davin and Andrew Heybey. ÒA Simulation Study of Fair Queueing and Policy EnforcementÓ *ACM Computer Communication Review,* Volume 20 (5), pp. 23-29, October 1990.

[Demers 89] - Alan Demers, Srinivasan Keshav and Scott Shenker. ÒAnalysis and Simulation of a Fair Queueing AlgorithmÓ *ACM Computer Communication Review,* Volume 19 (4), pp. 1-12, September 1989.

[Faber 92] - Theodore Faber, Lawrence Landweber and Amarnath Mukherjee. ÒDynamic Time Windows: Packet Admission Control with FeedbackÓ In *Proceedings of the SigCommÕ92 Conference,* ACM, pp. 124-135, Baltimore, USA, August 1992.

[Ferrari 90] - Domenico Ferrari and Dinesh Verma. ÒA scheme for Real-time Channel Establishment in Wide-area NetworksÓ *IEEE Journal on Selected Areas in Communications,* Volume 8 (2), pp. 368-379, February 1990.

[Ferrari 93] - Domenico Ferrari. ÒDistributed Delay Jitter Control in Packet-switching InternetworksÓ *Internetworking: Research and Experience,* Volume 4, pp. 1-20, 1993.

[Figueira 95] - Norival Figueira and Joseph Pasquale. ÒLeave-in-Time: A New Service Discipline for Real-Time Communications in Packet Switching NetworksÓ In *Proceedings of the SigCommÕ95 Conference,* ACM, pp. 207-218, Cambridge, MA, USA, August 1995.

[Golestani 90] - Jamaloddin Golestani. ÒA Stop-and-Go Queueing Framework for Congestion ControlÓ In *Proceedings of the SigCommÕ90 Conference,* ACM, pp. 8-18, Philadelphia, USA, September 1990.

[Golestani 91a] - Jamaloddin Golestani. ÒA Framing Strategy for Congestion ManagementÓ *IEEE Journal on Selected Areas in Communications,* Volume 9 (7), pp. 1064-1077, September 1991.

[Golestani 91b] - Jamaloddin Golestani. ÒCongestion-Free Communication in High-Speed Packet NetworksÓ *IEEE Transactions on Communications,* Volume 39 (12), pp. 1802-1812, December 1991.

[ISO 95] - ISO/IEC JTC1/SC21 N9680, Open Systems Interconnection, Data management and Open Distributed Processing. ÒInformation Technology - Quality of Service - FrameworkÓ Final CD, July 1995.

[Kalmanek 90] - Charles Kalmanek, Hemant Kanakia and Srinivasan Keshav. ÒRate Controlled Servers for Very High-speed NetworksÓ In *Proceedings of the GLOBECOMÕ90 Conference,* IEEE, San Diego, USA, December 1990.

[Keshav 91a] - Srinivasan Keshav. ÒOn the Efficient Implementation of Fair QueueingÓ *Internetworking: Research and Experience,* Volume 2, pp. 157-173, 1991.

[Keshav 91c] - Srinivasan Keshav. *Congestion Control in Computer Networks.* PhD thesis, Technical Report UCB:CSD-91-649, University of California, Berkeley, USA, December 1991.

[McKenney 90] - Paul McKenney. ÒStochastic Fairness QueueingÓ In *Proceedings of the INFOCOMÕ90 Conference,* IEEE, pp. 733-740, San Francisco, USA, June 1990.

[Monteiro 93] - Edmundo Monteiro, Fernando Boavida, Vasco de Freitas, "A Fairness Analysis of LAN/WAN Protocol Relays", *Proceedings of the 4th Joint European Networking Conference, European Research Networking in a Global Context* , May 10-13, 1993, Trondheim, Noruega, RARE/ACM/EARN,EUnet/IFIP/IAB/Internet Society/NORDUnet/UNINET, Amsterdam, pp. 250-260, 1993.

[Nagle 85] - John Nagle. *On Packet Switches With Infinite Storage.* RFC 970, Ford Aerospace and Communications Corporation, December 1985.

[Nagle 87] - John Nagle. ÒOn Packet Switches With Infinite StorageÓ *IEEE Transactions on Communications,* Volume 35, pp. 435-438, 1987.

[Panwalkar 76] - S. Panwalkar and Wafik Iskander. ÒA Survey of Scheduling RulesÓ *Operational Research*, Vol. 25 no. 1., pp. 45-61, April 1976.

[Parekh 93] - A. K. Parekh and R. G. Gallager. ÒA Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks - The Single Node CaseÓ *IEEE/ACM Transactions on Networking*, Vol. 1, 1993.

[Parekh 94] - Abhay K. Parekh and Robert G. Gallager. ÒA Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node CaseÓ *IEEE/ACM Transactions on Networking*, Vol. 2, no. 2, pp. 137-150, April 1994.

[Shreedhar 95] - M. Shreedhar and George Varghese. ÒEfficient Fair Queuing using Deficit Round RobinÓ *Computer Communication Review*, Volume 25, no. 4, pp 231-242, October 1995.

[Shreedhar 96] - M. Shreedhar and George Varghese. ÒEfficient Fair Queuing using Deficit Round RobinÓ *IEEE/ACM Transactions on Networking*, Vol. 4, no. 3, pp. 375-385, June 1996.

[Trajkovic 92] - Ljiljana Trajkovic and Jamaloddin Golestani. ÒCongestion Control for Multimedia ServicesÓ *IEEE Network Magazine*, Volume 6(5), pp. 20-25, September 1992.

[Verma 91] - Dinesh Verma, Hui Zhang and Domenico Ferrari. ÒGuaranteeing Delay Jitter Bounds in Packet Switching NetworksÓ In *Proceedings of the TricommÕ91 Conference*, Chapel Hill, USA, April 1991.

[ZhangHui 91] - Hui Zhang and Srinivasan Keshav. ÒComparison of Rate-Based Service DisciplinesÓ In *Proceedings of the SigCommÕ91 Conference,* ACM, pp. 113-121, Zurich, Switzerland, September 1991.

[ZhangHui 93] - Hui Zhang and Domenico Ferrari. ÒRate-Controlled Static-Priority QueueingÓ In *Proceedings of IEEE INFOCOMÕ93,* IEEE, 1993.

[ZhangLixia 90] Lixia Zhang. ÒVirtualClock: A New Traffic Control Algorithm for Packet Switching NetworksÓ In *Proceedings of the SigCommÕ90 Conference*, ACM, pp. 19-29, Philadelphia, USA, September 1990.