Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2429015 C 2010/12/21

(11)(21) **2 429 015**

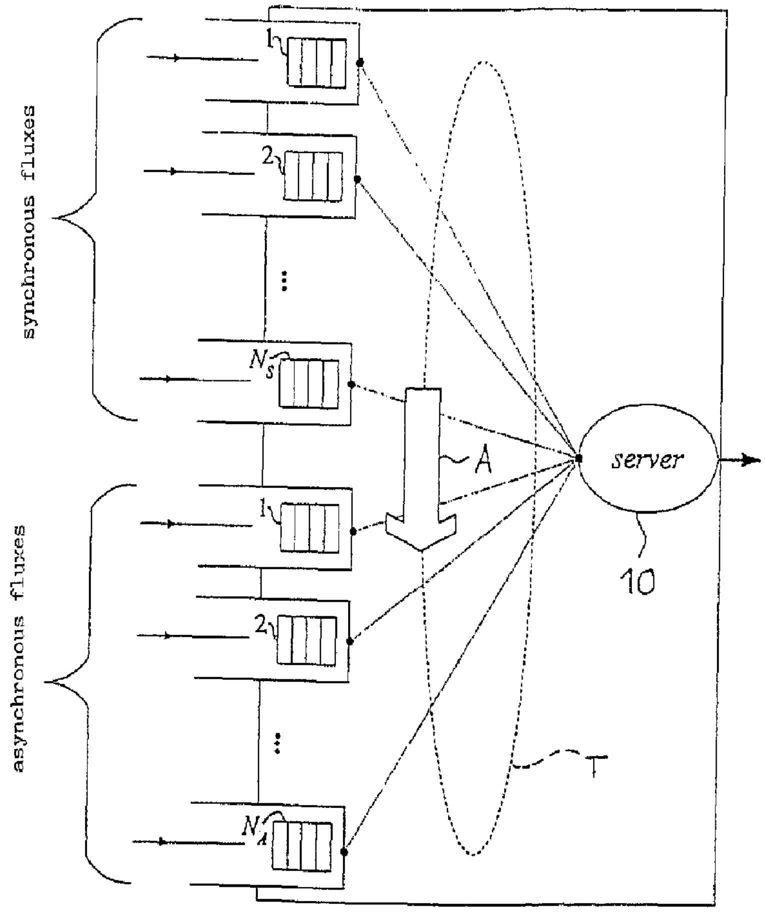
(12) **BREVET CANADIEN
CANADIAN PATENT**

(13) **C**

(86) Date de dépôt PCT/PCT Filing Date: 2001/10/19
(87) Date publication PCT/PCT Publication Date: 2002/05/02
(45) Date de délivrance/Issue Date: 2010/12/21
(85) Entrée phase nationale/National Entry: 2003/04/10
(86) N° demande PCT/PCT Application No.: IT 2001/000536
(87) N° publication PCT/PCT Publication No.: 2002/035777
(30) Priorité/Priority: 2000/10/23 (IT TO2000A001000)

(51) Cl.Int./Int.Cl. *H04L 12/56* (2006.01), *H04L 12/64* (2006.01), *H04L 29/06* (2006.01)
(72) Inventeurs/Inventors:
LENZINI, LUCIANO, IT;
MINGOZZI, ENZO, IT;
SCARRONE, ENRICO, IT;
STEA, GIOVANNI, IT
(73) Propriétaire/Owner:
TELECOM ITALIA S.P.A., IT
(74) Agent: RIDOUT & MAYBEE LLP

(54) Titre : PROCEDURE ET SYSTEME DE PROGRAMMATION D'UNE RESSOURCE PARTAGEE ENTRE DES FLUX DE PAQUETS D'INFORMATIONS MULTIPLES
(54) Title: PROCEDURE AND SYSTEM FOR SCHEDULING A SHARED RESOURCE BETWEEN MULTIPLE INFORMATION PACKET FLOWS



(57) Abrégé/Abstract:
Each synchronous flow (h=1, 2,, N_s) is associated to a respective synchronous capacity value (H_h) indicative of the maximum amount of time for which a synchronous flow can be served before relinquishing the token. Each asynchronous flow (l=1, 2, , N_A)

(57) Abrégé(suite)/Abstract(continued):

is, on the other hand, associated to a respective indicative value of the delay to be recovered so that the respective queue has the right to be served and to another value indicating the instant in which the server visited the respective queue in the pervious cycle. Each queue associated to a synchronous flow (h) is therefore served for a maximum amount of time that is equal to the aforesaid synchronous capacity value, while each queue associated to an asynchronous flow (i) is only served if the server's visit takes place with anticipation with respect to the expected instant. This anticipation is determined as the difference between the expected rotation time, needed by the server (10) to complete a visit cycle (T) of the queues associated to the aforesaid flows (h, i), and the time that has passed since the server's previous visit (10) and the delay accumulated. This difference, if positive, defines the maximum service time for the asynchronous queue. If the queue is empty when the server visits it, the server (10) moves on to the next queue even before the relative maximum service time has passed.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 May 2002 (02.05.2002)

PCT

(10) International Publication Number
WO 02/35777 A1

(51) International Patent Classification^7: H04L 12/56, 12/64, 29/06

(21) International Application Number: PCT/IT01/00536

(22) International Filing Date: 19 October 2001 (19.10.2001)

(25) Filing Language: Italian

(26) Publication Language: English

(30) Priority Data:
TO2000A001000 23 October 2000 (23.10.2000) IT

(71) Applicant *(for all designated States except US)*: **TELECOM ITALIA LAB S.P.A.** [IT/IT]; Via Reiss Romoli, 274, I-10148 Torino (IT).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **LENZINI, Luciano** [IT/IT]; c/o Telecom Italia Lab S.p.a., Via Reiss Romoli, 274, I-10148 Torino (IT). **MINGOZZI, Enzo** [IT/IT]; c/o Telecom Italia Lab S.p.a., Via Reiss Romoli, 274, I-10148 Torino (IT). **SCARRONE, Enrico** [IT/IT]; c/o Telecom Italia Lab S.p.a., Via Reiss Romoli, 274, I-10148 Torino (IT). **STEA, Giovanni** [IT/IT]; c/o Telecom Italia Lab S.p.a., Via Reiss Romoli, 274, I-10148 Torino (IT).

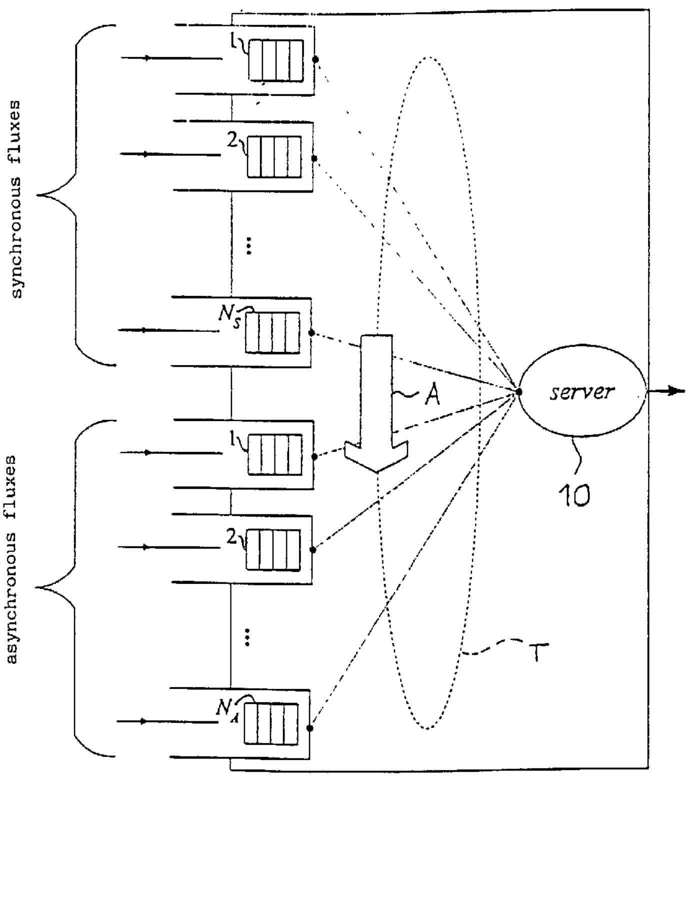
(74) Common Representative: **CASUCCIO, Carlo**; Telecom Italia Lab S.p.a, IPR Management, Via Reiss Romoli, 274, I-10148 Torino (IT).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF,

[Continued on next page]

(54) Title: PROCEDURE AND SYSTEM FOR SCHEDULING A SHARED RESOURCE BETWEEN MULTIPLE INFORMATION PACKET FLOWS



(57) Abstract: Each synchronous flow (h=1, 2, , N_s) is associated to a respective synchronous capacity value (H_h) indicative of the maximum amount of time for which a synchronous flow can be served before relinquishing the token. Each asynchronous flow (I=1, 2, , N_A) is, on the other hand, associated to a respective indicative value of the delay to be recovered so that the respective queue has the right to be served and to another value indicating the instant in which the server visited the respective queue in the pervious cycle. Each queue associated to a synchronous flow (h) is therefore served for a maximum amount of time that is equal to the aforesaid synchronous capacity value, while each queue associated to an asynchronous flow (i) is only served if the server's visit takes place with anticipation with respect to the expected instant. This anticipation is determined as the difference between the expected rotation time, needed by the server (10) to complete a visit cycle (T) of the queues associated to the aforesaid flows (h, i), and the time that has passed since the server's previous visit (10) and the delay accumulated. This difference, if positive, defines the maximum service time for the asynchronous queue. If the queue is empty when the server visits it, the server (10) moves on to the next queue even before the relative maximum service time has passed.

CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

"PROCEDURE AND SYSTEM FOR SCHEDULING A SHARED RESOURCE BETWEEN MULTIPLE INFORMATION PACKET FLOWS"

* * *

This invention refers to the packet communication systems, and in particular to the scheduling criteria of a shared resource, i.e. the criteria used to select the packet to which the resource is to be assigned each time this occurs.

The solution given in the invention has been developed both for radio resource scheduling (e.g.: MAC level scheduling), and for the scheduling of computational and transmissive resources in the network nodes (e.g.: flow scheduling with different service quality on Internet Protocol router (IP). The following description is based especially on the latter application example, and is given purely as an example and does not limit the scope of the invention.

For several years now, the widespread application and rapid evolution of the packet networks have given rise to the problem of integrating the traditional services offered by the old generation packet networks (electronic mail, web surfing, etc.) and the new services previously reserved for circuit switching networks (real time video, telephony, etc.) into the so-called integrated services networks. The integrated services networks must therefore be able to handle traffic flows with different characteristics and to offer each type of flow a suitable service quality, a set of performance indexes negotiated between user and service provider, which must be guaranteed within the terms agreed upon.

One of the key elements in providing the service quality requested is given by the scheduling implemented on the network nodes, i.e. by the criteria with which the packet to be transmitted is selected each time from those present on the node; this criteria must obviously match the following characteristics:

- flexibility, in the sense of capacity to provide different types of services;
- simplicity, a characteristic that makes it possible to use in environments that require high transmission speeds and the handling of numerous transmission flows; and
- efficiency in the use of the shared resource (e.g. the transmissive means).

This invention, having the characteristics referred to in the claims that follow, initially consists of a scheduling procedure that can satisfy the aforesaid requirements. Another aspect of the invention is that it also relates to the relative system.

In particular, the solution given in the invention is able to provide different types of service at a low computational cost, and can therefore be applied to computer networks that must guarantee its users quality of service, like the IP networks in intserv or diffserv techniques. The solution given in the invention also applies to the scheduling systems of radio resources such as MAC level scheduling algorithms (W-LAN systems, third-generation mobile-radio services).

In particular, the solution given in the invention guarantees the bit rate of the various flows, the maximum queueing delay and the maximum occupation of the buffers of each flow for synchronous traffic.

In its current preferred form of actuation, the solution given in the invention is capable of providing the following characteristics:

- flexibility: the solution given in the invention offers two different types of service, rate-guaranteed (suitable for synchronous flows) and best-effort (suitable for asynchronous flows), and is therefore able to function in service integration networks;

- isolation of flows: the special architecture makes it possible to isolate the transmission flows, i.e. it makes the service offered to a single-flow independent from the presence and behaviour of other flows;

- low computational complexity: the number of operations necessary to select the packet to be transmitted each time is independent from the number of transmission flows present, and therefore the system has one computational complexity 0(1); this property makes the system particularly suitable for environments in which the transmission speeds and the number of flows are high;

- adaptability: the solution given in the invention is able to handle a change in the operating parameters (e.g. the number of flows present) by redistributing its resources without having to resort to complex procedures; and

- analytic describability: a complete analytic description of the system's behaviour is provided; this makes it possible to relate the service quality measurements to the system parameters.

The following description of the invention is given as a non-limiting example, with reference to the annexed drawing, which includes a single block diagram figure that illustrates the operating criteria of a system working according to the invention.

A scheduling system as given in the invention is able to multiplex a single transmission channel into multiple transmission flows.

The system offers two different types of service: a rate-guaranteed service, suitable for transmission flows (henceforth, h synchronous flows with h=1, 2, …, N_S) that require a guaranteed minimum service rate, and a best-effort service, suitable for transmission flows (henceforth, i asynchronous flows, with i= 1, 2, …, N_A) that do not require any guarantee on the service rate. The system provides the latter, however, with a balanced sharing of the transmission capacity not used by the synchronous flows.

The traffic from each transmission flow input on the node is inserted in its queue (synchronous or asynchronous queues will be discussed later) from which it will be taken to be transmitted. The server 10 visits the queues in a fixed cyclic order (ideally illustrated in the figure of the drawings with trajectory T and arrow A), granting each queue a service time established according to precise timing constraints at each visit.

System operation as given in the invention includes initialisation followed by the cyclic queue visit procedures. These procedures will be discussed later.

Initialisation

First of all, it is necessary to give the system the information relating to the working conditions: how many synchronous flows there are (in general: N_A), what the transmission rate requested by each of these flows is, how many asynchronous flows there are, the expected rotation time (TTRT), i.e. how long a complete cycle during which the server visits all the queues once is to last.

On the basis of this information, the system parameters can be defined:

- each synchronous flow h, $h=1…N_S$, is associated, according to an appropriate allocation policy, to a variable H_h (synchronous capacity) that measures the maximum time for which the traffic of a synchronous flow can be transmitted before relinquishing the token. The possible allocation policies will be described below;

- each asynchronous flow i $i=1…N_A$ is associated to two variables, *lateness(i) and last_token_time(i);* the first variable stores the delay that must be made up for the asynchronous queue i to have the right to be served; the second variable stores the instant in which the server visited the asynchronous queue i in the previous cycle. These variables are initialised to zero.

The system clock is also started; supposing that the reading of the *current_time* variable gives the current time with the desired precision, the queue scanning will start.

Visit to a generic synchronous queue h, with h = 1...N_S

A synchronous queue can be served for a period of time equal to its maximum synchronous capacity H_h, determined during the initialisation stage. If the queue being served is empty, the server will move on to visit the next queue, even if the H_h time has not passed.

Visit to a generic asynchronous queue i, with i = 1...N_A

An asynchronous code can be served only if the server's visit occurs before the expected instant. To calculate whether the server's visit is in advance, subtract the time that has passed between the previous visit and the accumulated delay *lateness(i)* from the expected rotation time TTRT. If this difference is positive, it gives the period of time for which the asynchronous queue i has the right to be served, and in this case the *lateness* variable *(i)* is reset. If the difference is negative, the server is late, and therefore the queue i cannot be served; in this case, the delay is stored in the *lateness* variable *(i)*. The same applies to the asynchronous queues; if the queue being served is empty, the server will move on to visit the next one even if the previously calculated service time has not yet passed completely.

The pseudocode illustrated below analytically describes the behaviour of a system as given in the invention which proposes the scheduling of N_A asynchronous flows and N_S synchronous flows simultaneously (N_A and N_S must be non-negative integers). It should be supposed that each synchronous flow h, $h=1 \ldots N_S$ requires a service rate equal to f_h times the capacity of the output channel ($0 \le f_h \le 1$), and that the sum of the service rates requested by the synchronous flows does not exceed the capacity of the channel itself ($\sum_{h=1}^{N_S} f_h \le 1$).

Initialisation

```
fetch_parameters ($N_S$, $f_1 \ldots f_{N_S}$, $N_A$, TTRT);
select_parameters ($H_1 \ldots H_{N_S}$);
for (i=1 to $N_A$) {lateness(i) = 0; last_token_time(i)
= 0 ; }
current_time = 0 ;
```

```
Start_Cycle;
Visit to a generic synchronous queue h, with h = 1...$N_S$:
Transmit_for_a_Time ($H_h$);
Next_Visit;
Visit to a generic asynchronous queue i, with i = 1...$N_A$:
t = current_time;
temp = TTRT - lateness(i) - (t - last_token_time(i));
if (temp >0)
 { Transmit_for_a_Time (temp);
   lateness(i)=0 ; }
else
   lateness(i) = - temp;
last_token_time(i) = t;
Next_Visit;
```

The ability to guarantee that the synchronous flows receive a minimum service rate that is not less than that requested depends on whether the synchronous capacities H_h, $h=1 \ldots N_S$ have been selected correctly. In the system given in the invention, the H_h, $h=1 \ldots N_S$ are selected in proportion to the value of the expected rotation time TTRT:

$$H_h = TTRT \cdot C_h$$

The values of the proportionality constant C_h can be selected according to one of the following two schemes:

- local scheme: $C_h = f_h$
- global scheme: $C_h = \dfrac{N_A \cdot f_h}{N_A + 1 - \sum_{j=1}^{N_S} f_j}$

The applicability of the global scheme is naturally linked to the presence of at least one asynchronous flow.

If the H_h are calculated following one of the afore-mentioned schemes, each synchronous flow is served at a rate that is no less than r_h times the capacity of the channel, with r_h given by the following expression:

$$r_h = \frac{[N_A + 1] \cdot C_h}{N_A + \sum_{j=1}^{N_S} C_j} \geq f_h$$

and it can be guaranteed that, given any interval of time $[t_1, t_2)$ in which the generic synchronous queue h is never empty, the service time $W_h(t_1, t_2)$ received by the h queue in $[t_1, t_2)$, the following inequality will occur:

$$0 < r_h \cdot (t_2 - t_1) - W_h(t_1, t_2) \leq \Lambda_h < \infty, \quad \forall t_2 \geq t_1, h = 1 \ldots N_S \qquad (1)$$

with:

$$\Lambda_h = C_h \cdot TTRT \cdot (2 - r_h) < \min(2H_h, TTRT)$$

Relation (1) above establishes that the service provided by the system given in the invention to a synchronous flow h does not differ by more than Λ_h

from the service that the same flow would experience if it were the only owner of a private transmission channel with a capacity equal to r_h times that of the channel handled by the scheduler as given in the invention. Λ_h therefore represents the maximum service delay with respect to an ideal situation. Since Λ_h is proportional to TTRT, TTRT can be selected to limit the maximum service delay.

The global scheme guarantees a better use of the transmission capacity of the channel with respect to the local scheme, in that under the same operating conditions it allocates a lower capacity to the synchronous flows, leaving a larger section of the band free for asynchronous flow transmissions.

On the other hand, the use of a global scheme envisages that all the H_h parameters are recalculated each time the number of flows (synchronous or asynchronous) in the system changes; the use of a local scheme, however, means that the H_h can be established independently from the number of flows present in the system.

The guarantee on the minimum service rate makes it possible to provide guarantees on the maximum buffer occupation (backlog) and on the maximum queuing delay for synchronous traffic if appropriate mechanisms for conditioning input traffic are used.

Assuming a composite leaky bucket is used as a traffic conditioning mechanism, consisting of $n \geq 1$ leaky bucket in cascade, and granting that each leaky bucket is characterised by a pair of parameters (b_j, t_j), $j = 1 \ldots n$, where b_j is the dimension of the leaky bucket (expressed in units of time), and $1/t_j$ is the filling rate of the leaky bucket, it is possible to define the following quantities:

$$T_j = \frac{b_j - b_{j+1}}{t_j - t_{j+1}} t_j t_{j+1}$$

$$B_j = \frac{b_j t_j - b_{j+1} t_{j+1}}{t_j - t_{j+1}}$$

where $b_{n+1} = 0$ and $t_{n+1} = 0$ are introduced for the sake of easy notation. We can suppose (without losing general aspects) that the following inequalities have occurred: $t_j > t_{j+1}$, $b_j > b_{j+1}$, $T_j > T_{j+1}$ for $j = 1 \ldots n-1$.

Supposing that the generic synchronous flow k has guaranteed a rate equal to r_k, if the traffic sent by the synchronous flow k is limited by a composite leaky bucket with n stages described by the parameters (b_j, t_j), $j = 1 \ldots n$, the following guarantees can be formulated.

If $r_k \geq 1/t_1$, then both the backlog and the queuing delay have an upper limit; in addition, if the single leaky bucket is marked with index i, we have: $1/t_i \leq r_k < 1/t_{i+1}$, $i = 1 \ldots n$:

- the queuing delay is limited at the top by: $d_k = (\Lambda_k + B_i)/r_k - T_i$
- if $\Lambda_k/r_k \leq T_i$, the backlog is limited at the top by: $q_k = \Lambda_k + B_i - r_k \cdot T_i$

if $\Lambda_k/r_k > T_i$, the backlog is limited at the top by: $q_k = \dfrac{\Lambda_k}{t_h \cdot r_k} + b_h$, where h is the leaky bucket that checks the inequality $T_h \leq \Lambda_k/r_k < T_{h-1}$, $h = 1 \ldots i^1$.

$T_0 = \infty$ has been used in the above description for the sake of easy notation.

Obviously the details of how this is done can be altered with respect to what has been described, without however, leaving the context of this invention.

1.

Procedure for, in a node of a packet communication network, scheduling a service resource shared between multiple information packet flows, said flows generating respective associated queues wherein traffic from each said flows input to said node is inserted in a respective queue, and being served by the attribution of a token; this plurality of flows includes synchronous flows ($h = 1, 2, ..., N_S$) , which require a guaranteed minimum service rate, and asynchronous flows ($i = 1, 2, ..., N_A$) destined to exploit the service capacity of said resource left unused by the synchronous flows,

wherein the procedure includes the following operations:

- provides a server (10) that visits the respective queues associated to said flows (h, i) in successive cycles, determining a time value of expected rotation (TTRT), which identifies the time necessary for the server (10) to complete a visit cycle on the said respective queues,

- associates to each synchronous flow (h) a respective synchronous capacity value (H_h) indicative of the maximum amount of time for which a synchronous flow can be served before relinquishing the token,

- associates to each asynchronous flow (i) a first respective delay value (lateness(i)) that identifies the value that must be made up for the respective queue to have the right to be served, and a second value (*last_token_time*) that indicates the instant in which the server (10) visited the respective queue in the previous cycle, determining for said respective queue, the time that has passed since the previous visit of the server (10),

- serves each queue associated to a synchronous flow (h) for a maximum service time equal to said respective value of synchronous capacity (H_h), and

- serves each queue associated to an asynchronous flow (i) only if the server's visit (10) occurs before the expected instant, said advance being determined as the difference between said expected rotation time value (TTRT) and the time that has passed since the server's (10) previous visit and the accumulated delay; if positive, this difference defines the maximum service time for each said queue.

2. Procedure as per claim 1, characterised by the fact that it includes, if the queue is empty when the server visits it, the operation that makes the server (10) visit a subsequent queue even before the relative maximum service time has passed.

3. Procedure as per claim 1 or claim 2, characterised by the fact that, in the case in which

said difference is negative, each said queue associated to an asynchronous flow (i) is not served and the value of said difference is accumulated with said delay.

4. Procedure as per any of the claims 1 to 3, characterised by the fact that said first respective value (*lateness(i))* and said second respective value (*last_token_time*) are initialised to zero.

5. Procedure as per any one of the claims 1 to 4, characterised by the fact that said respective synchronous capacity value (H_h) is determined in proportion to said expected rotation time value (TTRT).

6. Procedure as per claim 5, characterised by the fact that said synchronous capacity value (H_h) is determined in proportion to said expected rotation time value (TTRT) according to a proportionality factor (C_h) selected in relation to the respective proportionality factor (f_h) between the service rate requested by the respective synchronous flow (h) and the service capacity of said shared resource.

7. Procedure as per claim 6, characterised by the fact that said proportionality factor (C_h) is selected equal to said respective proportionality factor.

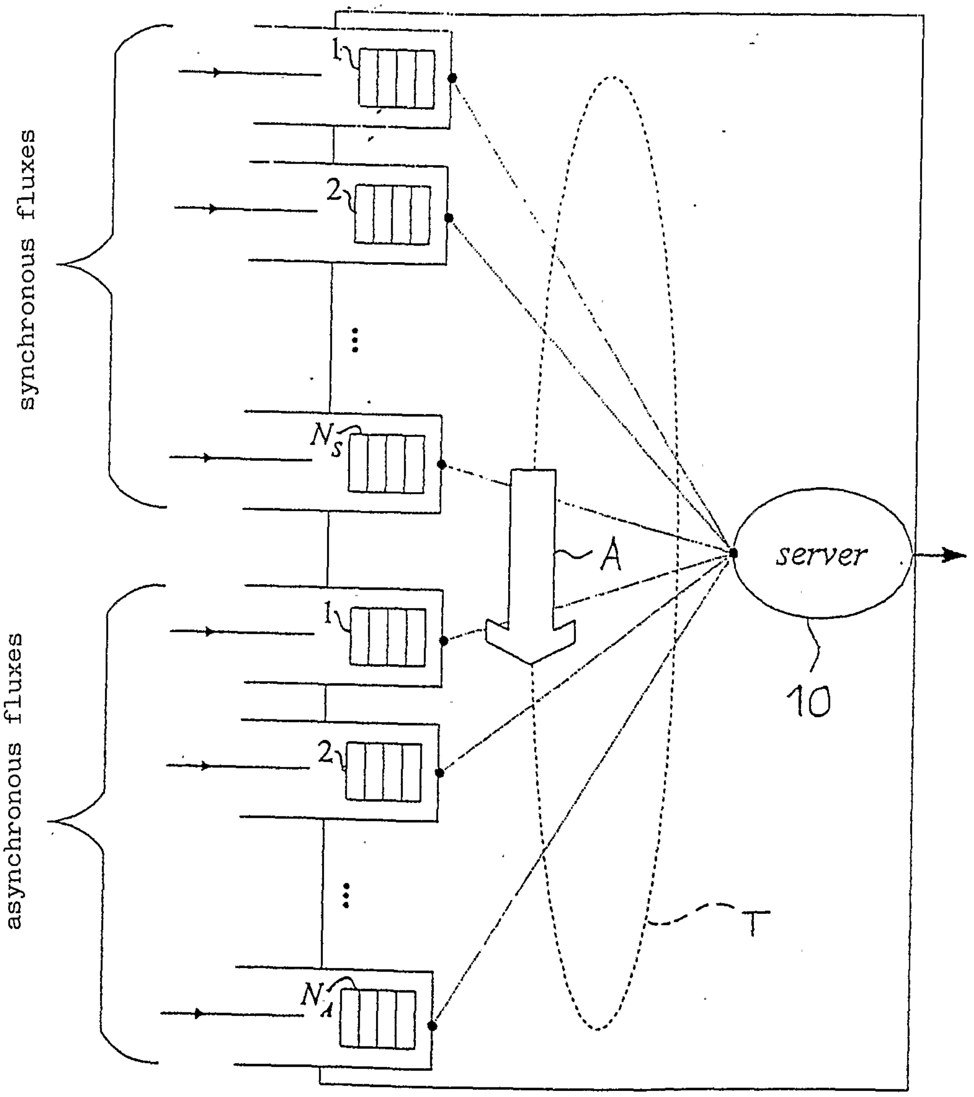
8. Procedure as per claim 6, characterised by the fact that said proportionality factor (C_h) is selected on the basis of the following formula $C_h = \frac{N_A \cdot f_h}{N_A + 1 - \sum_{j=1}^{N_S} f_j}$ where:

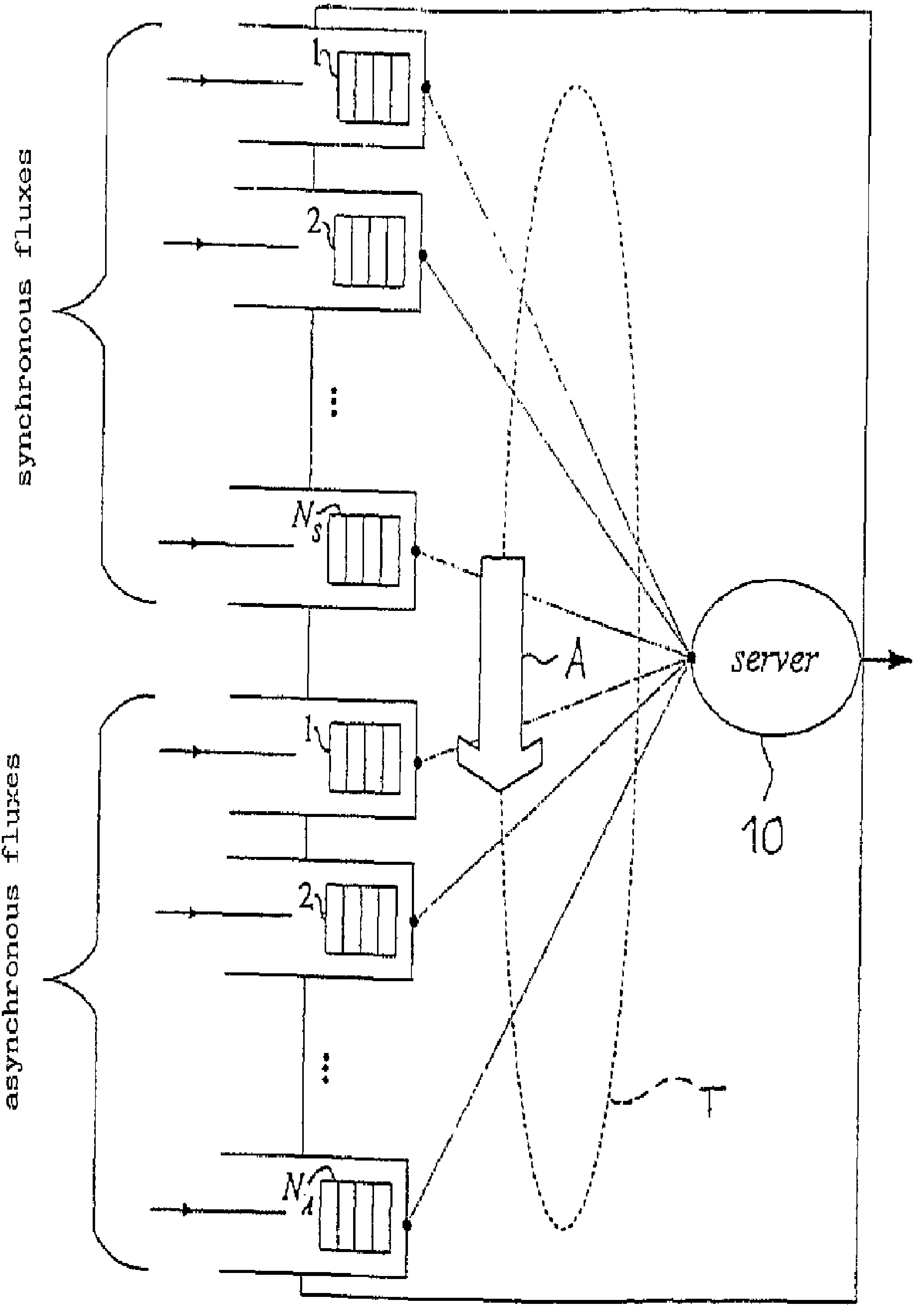
- f_j stands for said respective proportionality factor relating to the j-th synchronous flow, and

- N_A is the number of said asynchronous flows.

9. System for scheduling, in a node of a packet communication network, a service resource shared between multiple information packet flows, said flows generating respective associated queues wherein traffic from each said flows input to said node is inserted in a respective queue and being served by the attribution of a token; this plurality of flows includes synchronous flows ($h = 1, 2, ..., N_S$) , which require a guaranteed minimum service rate, and asynchronous flows ($i = 1, 2, ..., N_A$) destined to exploit the service capacity of said resource left unused by the synchronous flows,

wherein the systems includes a server (10) that is configured to visit the respective queues associated to said flows (h, i) in successive cycles and system means configured to perform the operations of the procedure of any of claims 1 to 8.


synchronous fluxes
1
2
N_s
asynchronous fluxes
1
2
N_A
A
T
server
10


synchronous fluxes
asynchronous fluxes
1
2
N_s
1
2
N_A
A
server
10
T