



(11)

EP 1 329 065 B1

(12)

## EUROPEAN PATENT SPECIFICATION

- |   |  |
|---|--|
| (45) Date of publication and mention<br>of the grant of the patent:<br><b>20.03.2013 Bulletin 2013/12</b> | (51) Int Cl.:<br><b>H04L 12/66 (2006.01)</b><br><b>H04L 29/06 (2006.01)</b>                  |
| (21) Application number: <b>01983777.2</b>  | (86) International application number:<br><b>PCT/IT2001/000536</b>                           |
| (22) Date of filing: <b>19.10.2001</b>  | (87) International publication number:<br><b>WO 2002/035777 (02.05.2002 Gazette 2002/18)</b> |

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

VERFAHREN UND VORRICHTUNG ZUR STEUERUNG VON MEHREREN INFORMATIONPAKETFLUSSEN AUF EINEM GEMEINSAMEN BETRIEBSMITTEL

PROCEDURE ET SYSTEME DE PROGRAMMATION D'UNE RESSOURCE PARTAGEE ENTRE DES FLUX DE PAQUETS D'INFORMATIONS MULTIPLES

<p>(84) Designated Contracting States: <b>AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR</b></p> <p>(30) Priority: <b>23.10.2000 IT TO20001000</b></p> <p>(43) Date of publication of application: <b>23.07.2003 Bulletin 2003/30</b></p> <p>(73) Proprietor: <b>Telecom Italia S.p.A.</b> <b>20123 Milano (IT)</b></p> <p>(72) Inventors: <ul style="list-style-type: none"> <li>• <b>LENZINI, Luciano</b> <b>TELECOM ITALIA S.p.A.</b> <b>I-10148 Torino (IT)</b></li> <li>• <b>MINGOZZI, Enzo,</b> <b>TELECOM ITALIA S.p.A.</b> <b>I-10148 Torino (IT)</b></li> <li>• <b>SCARRONE, Enrico</b> <b>TELECOM ITALIA S.p.A.</b> <b>I-10148 Torino (IT)</b></li> <li>• <b>STEA, Giovanni,</b> <b>TELECOM ITALIA S.p.A.</b> <b>I-10148 Torino (IT)</b></li> </ul> </p> <p>(74) Representative: <b>Bosotti, Luciano et al</b> <b>Buzzi, Notaro &amp; Antonielli d'Oulx</b> <b>Via Maria Vittoria, 18</b> <b>10123 Torino (IT)</b></p> <p>(56) References cited:</p>	<ul style="list-style-type: none"> <li>• <b>GENTER W L ET AL: "Delay analysis of the FDDI synchronous data class" MULTIPLE FACETS OF INTEGRATION. SAN FRANCISCO, JUNE 3 - 7, 1990, PROCEEDINGS OF THE ANNUAL JOINT CONFERENCE OF THE COMPUTER AND COMMUNICATIONS SOCIETIES (INFOCOM), WASHINGTON, IEEE COMP. SOC. PRESS, US, vol. 2 CONF. 9, 3 June 1990 (1990-06-03), pages 766-773, XP01019457 ISBN: 0-8186-2049-8</b></li> <li>• <b>SADIQU M N O ET AL: "Performance comparison of FDDI models" SOUTHEASTCON '97. ENGINEERING NEW NEW CENTURY., PROCEEDINGS. IEEE BLACKSBURG, VA, USA 12-14 APRIL 1997, NEWYORK, NY, USA, IEEE, US, 12 April 1997 (1997-04-12), pages 135-137, XP010230771 ISBN: 0-7803-3844-8</b></li> <li>• <b>MALCOLM N ET AL: "Guaranteeing synchronous messages with arbitrary deadline constraints in an FDDI network" LOCAL COMPUTER NETWORKS, 1993., PROCEEDINGS., 18TH CONFERENCE ON MINNEAPOLIS, MN, USA 19-22 SEPT. 1993, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC, 19 September 1993 (1993-09-19), pages 186-195, XP010224384 ISBN: 0-8186-4510-5</b></li> </ul>
---	--

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

- WELZEL T: "Performance analysis of token rings as high speed backbone networks" MULTIPLE FACETS OF INTEGRATION. SAN FRANCISCO, JUNE 3 - 7, 1990, PROCEEDINGS OF THE ANNUAL JOINT CONFERENCE OF THE COMPUTER AND COMMUNICATIONS SOCIETIES (INFOCOM), WASHINGTON, IEEE COMP. SOC. PRESS, US, vol. 2 CONF. 9, 3 June 1990 (1990-06-03), pages 23-29, XP010019366 ISBN: 0-8186-2049-8

**Description**

[0001] 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.

5 [0002] 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.

10 [0003] 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.

15 [0004] 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;
- 20 - 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).

25 [0005] GENTER W L ET AL: "Delay analysis of the FDDI synchronous data class" MULTIPLE FACETS OF INTEGRATION. SAN FRANCISCO, JUNE 3 - 7, 1990, PROCEEDINGS OF THE ANNUAL JOINT CONFERENCE OF THE COMPUTER AND COMMUNICATIONS SOCIETIES (INFOCOM), WASHINGTON, IEEE COMP. SOC. PRESS, US, vol. 2 CONF. 9, 3 June 1990 (1990-06-03); pages 766-773 discloses a model for describing performance of a network having a ring-like architecture applied to a (Fiber Distributed Data Interface).

30 [0006] 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.

35 [0007] 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).

[0008] 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.

40 [0009] 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  $O(1)$ ; this property makes the system particularly suitable for environments in which the transmission speeds and the number of flows are high;
- 45 - 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.

55 [0010] 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.

[0011] A scheduling system as given in the invention is able to multiplex a single transmission channel into multiple

transmission flows.

[0012] The system offers two different types of service: a rate-guaranteed service, suitable for transmission flows (henceforth,  $h$  synchronous flows with  $h=1, 2, \dots, N_S$ ) that require a guaranteed minimum service rate, and a best-effort service, suitable for transmission flows (henceforth,  $i$  asynchronous flows, with  $i=1, 2, \dots, N_A$ ) 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.

[0013] 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.

[0014] System operation as given in the invention includes initialisation followed by the cyclic queue visit procedures. These procedures will be discussed later.

#### Initialisation

[0015] 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.

[0016] On the basis of this information, the system parameters can be defined:

- each synchronous flow  $h$ ,  $h=1 \dots N_S$ , 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  $i$   $i=1 \dots N_A$  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.

[0017] 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 $h = 1 \dots N_S$

[0018] 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 $i = 1 \dots N_A$

[0019] 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.

[0020] 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  $h$ ,  $h=1 \dots N_S$  requires a service rate equal to  $f_h$  times the capacity of the output channel ( $0 \leq f_h \leq 1$ ), and that the sum of the service rates requested by the synchronous

flows does not exceed the capacity of the channel itself ( $\sum_{h=1}^{N_S} f_h \leq 1$ ).

Initialisation**[0021]**

```

5   fetch_parameters (NS, f1...fNs, A, TTRT) ;
    select_parameters (H1...HNs) ;
    for (i=1 to NA) {lateness(i) = 0; last_token_time(i)
    = 0 ; }
    current_time = 0 ;
    Start_Cycle;
10   Visit to a generic synchronous queue h, with h =
    1...NS;
    Transmit for a Time (Hh) ;
    Next_Visit ;
    Visit to a generic asynchronous queue i, with i =
15   1...NA;
    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;
25

```

**[0022]** 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  $H_h$ ,  $h=1...N_S$  have been selected correctly. In the system given in the invention, the  $H_h$ ,  $h=1...N_S$  are selected in proportion to the value of the expected rotation time TTRT:

$$30 \quad H_h = TTRT \cdot C_h$$

**[0023]** The values of the proportionality constant  $C_h$  can be selected according to one of the following two schemes:

- 35    - local scheme:

$$40 \quad C_h = f_h$$

- global scheme:

$$45 \quad C_h = \frac{N_A \cdot f_h}{N_A + 1 - \sum_{j=1}^{N_S} f_j}$$

**[0024]** The applicability of the global scheme is naturally linked to the presence of at least one asynchronous flow.

**[0025]** 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:

$$55 \quad r_h = \frac{[N_A + 1] \cdot C_h}{N_A + \sum_{i=1}^{N_S} C_i} \geq f_h$$

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:

$$5 \quad 0 < r_h \cdot (t_2 - t_1) - W_h(t_1, t_2) \leq \Lambda_h < \infty, \quad \forall t_2 \geq t_1, h = 1 \dots N_s \quad (1)$$

with:

$$10 \quad \Lambda_h = C_h \cdot TTRT \cdot (2 - r_h) < \min(2H_h, TTRT)$$

15 [0026] 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  $\Lambda_h$  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.  $\Lambda_h$  therefore represents the maximum service delay with respect to an ideal situation. Since  $\Lambda_h$  is proportional to TTRT, TTRT can be selected to limit the maximum service delay.

20 [0027] 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.

25 [0028] 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.

[0029] 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.

30 [0030] 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  $(b_j, t_j)$ ,  $j=1 \dots n$ , 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:

$$35 \quad T_i = \frac{b_j - b_{j+1}}{t_j - t_{j+1}} t_j t_{j+1}$$

$$40 \quad B_i = \frac{b_j t_j - b_{j+1} t_{j+1}}{t_j - t_{j+1}}$$

45 where  $b_{n+1} = 0$  and  $t_{n+1} = 0$  are introduced for the sake of easy notation. We want 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  $j = 1 \dots n-1$ .

[0031] 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  $(b_j, t_j)$ ,  $j=1 \dots n$ , the following guarantees can be formulated.

50 [0032] 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:  $1/t_i \leq r_k < 1/t_{i+1}$ ,  $i = 1 \dots n$ :

- the queuing delay is limited at the top by:

$$55 \quad d_k = (\Lambda_k + B_i) / r_k - T_i$$

- if  $\Lambda_k / r_k \leq T_i$ , the backlog is limited at the top

by:

$$q_k = \Lambda_k + B_i - r_k \cdot T_i$$

5

if  $\Lambda_k/r_k > T_i$ , the backlog is limited at the top by:  $q_k = \frac{\Lambda_k}{t_h \cdot r_k} + b_h$ ,

10

where h is the leaky bucket that checks the inequality  $T_h \leq \Lambda_k/r_k < T_{h+1}$ ,  $h=1\dots i^1$ .

- [0033]  $T_0=\infty$  has been used in the above description for the sake of easy notation.  
[0034] 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.

15

### Claims

- 20 1. Procedure 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 ( $h = 1, 2, \dots, N_S$ ), which require a guaranteed minimum service rate, and asynchronous flows ( $i = 1, 2, \dots, N_A$ ) destined to exploit the service capacity of said resource left unused by the synchronous flows,

25

characterised by the fact that it includes the following operations:

30 - 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),

35 - 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

40 - 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.

- 45 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.

- 50 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.

- 55 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.
- 5
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

10

$$\text{the following formula } C_h = \frac{N_A \cdot f_h}{N_A + 1 - \sum_{j=1}^{N_s} f_j} \text{ where:}$$

15

- $f_j$  stands for said respective proportionality factor relating to the  $j$ -th synchronous flow, and
- $N_A$  is the number of said asynchronous flows.

20

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 miserted in a respective queue and being served by the attribution of a token; this plurality of flows includes synchronous flows ( $h = 1, 2, \dots, N_S$ ), which require a guaranteed minimum service rate, and asynchronous flows ( $i = 1, 2, \dots, N_A$ ) destined to exploit the service capacity of said resource left unused by the synchronous flows,
- 25
- characterised by the fact that it 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.

30

### Patentansprüche

35

1. Verfahren zum Planen, in einem Knoten eines Paketkommunikationsnetzwerkes, einer Dienstleistungsressource, die zwischen mehreren Informationspaketflüssen geteilt wird, wobei die Flüsse entsprechend dazugehörige Warteschlangen erzeugen, wobei der Traffic von jedem der Flüsse, der in den Knoten eingegeben wird, in eine entsprechende Warteschlange eingefügt wird, und wobei den Flüssen ein Token zugeordnet wird, wobei diese mehreren Flüsse synchrone Flüsse ( $h = 1, 2, \dots, N_S$ ), die eine garantierte minimale Dienstleistungsr率e erfordern, und asynchrone Flüsse ( $i = 1, 2, \dots, N_A$ ) aufweisen, die bestimmt sind, um die Kapazität der durch die synchronen Flüsse ungenutzten Dienstleistungsressource auszunutzen,
- dadurch gekennzeichnet, dass es die folgenden weiteren Schritte umfasst:

40

- Bereitstellen eines Servers (10), der die entsprechend zugehörigen Warteschlangen der Flüsse ( $h, i$ ) in aufeinanderfolgenden Arbeitsgängen, die einen Zeitwert einer erwarteten Rotation (TTRT) bestimmen, inspiert, der die erforderliche Zeit identifiziert, die der Server (10) braucht, um den Inspektionsarbeitsgang für die entsprechende Warteschlange abzuschließen,

45

- zu jedem synchronen Fluss ( $h$ ) wird ein entsprechender synchroner Kapazitätswert ( $H_h$ ) verknüpft, der einen maximalen Zeitbetrag darstellt, für welchen ein synchroner Fluss verarbeitet werden kann, bevor der Token aufgegeben wird,

50

- zu jedem asynchronen Fluss ( $i$ ) wird ein erster entsprechender Verzögerungswert (lateness( $i$ )) verknüpft, der jenen Wert identifiziert, der für die entsprechende Warteschlange ausgeglichen werden muss, um das Recht zu besitzen, verarbeitet zu werden, und ein zweiter Wert (last\_token\_time) verknüpft, der den Moment andeutet, in welchem der Server (10) die entsprechende Warteschlange in einem vorherigen Arbeitsgang inspiert hat, um für die entsprechende Warteschlange die Zeit zu bestimmen, die seit einer vorherigen Inspektion durch den Server (10) verstrichen ist,

55

- Erbringen der Dienstleistung für jede zu einem synchronen Fluss ( $h$ ) zugehörigen Warteschlange für eine maximale Dienstleistungszeit, die gleich zu dem entsprechenden Wert der synchronen Kapazität ( $H_h$ ) ist, und
- Erbringen der Dienstleistung für jede zu einem asynchronen Fluss ( $i$ ) zugehörigen Warteschlange nur dann, wenn die Inspektion durch den Server (10) vor dem erwarteten Moment war, wobei das zeitliche Davorliegen als ein Differenz zwischen dem erwarteten Rotationszeitwert (TTRT) und der Zeit, die seit der vorherigen In-

spektion durch den Server (10) und der akkumulierten Verzögerung verstrichen ist, bestimmt wird, wobei, wenn sie positiv ist, definiert diese Differenz die maximale Dienstleistungszeit für jede der Warteschlangen.

2. Verfahren nach Anspruch 1, **dadurch gekennzeichnet, dass** es weiter Folgendes umfasst: wenn die Warteschlange bei der Inspektion durch den Server leer ist, inspiziert der Server (10) eine folgende Warteschlange sogar vor jenem Zeitpunkt, wenn die relative maximale Dienstleistungszeit abgelaufen ist.
3. Verfahren nach Anspruch 1 oder Anspruch 2, **dadurch gekennzeichnet, dass**, in dem Fall, wenn die Differenz negativ ist, jede zu einem asynchronen Fluss (i) dazugehörige Warteschlange nicht bedient wird und der Wert der Differenz mit der Verzögerung aufsummiert wird.
4. Verfahren nach einem der Ansprüche 1 bis 3, **dadurch gekennzeichnet, dass** der erste entsprechende Wert (lateness(i)) und der zweite entsprechende Wert (last\_token\_time) als Null initialisiert werden.
5. Verfahren nach einem der Ansprüche 1 bis 4, **dadurch gekennzeichnet, dass** der entsprechende synchrone Kapazitätswert ( $H_h$ ) im Verhältnis zu dem erwarteten Rotationszeitwert (TTRT) bestimmt wird.
6. Verfahren nach Anspruch 5, **dadurch gekennzeichnet, dass** der synchrone Kapazitätswert ( $H_h$ ) im Verhältnis zu dem erwarteten Rotationszeitwert (TTRT) entsprechend zu einem Proportionalitätsfaktor ( $C_h$ ) bestimmt wird und zwar in Bezug auf den entsprechenden Proportionalitätsfaktor ( $f_h$ ) zwischen der angefragten Dienstleistungsrate durch den entsprechenden synchronen Fluss (h) und der Kapazität der geteilten Dienstleistungsressourcen ausgewählt wird.
7. Verfahren nach Anspruch 6, **dadurch gekennzeichnet, dass** der Proportionalitätsfaktor ( $C_h$ ) gleich ist zu dem entsprechenden Proportionalitätsfaktor.
8. Verfahren nach Anspruch 6, **dadurch gekennzeichnet, dass** der Proportionalitätsfaktor ( $C_h$ ) auf der Basis der folgenden Formel ausgewählt wird:

$$C_h = \frac{N_A \cdot f_h}{N_A + 1 - \sum_{j=1}^{N_S} f_j},$$

35

wobei:

- $f_j$  der entsprechende Proportionalitätsfaktor, der sich auf den j-ten synchronen Fluss bezieht ist, und
- $N_A$  die Zahl der synchronen Flüsse ist.

9. System zum Planen, in einem Knoten eines Paketkommunikationsnetzwerkes, einer Dienstleistungsressource, die zwischen mehreren Informationspaketflüssen geteilt wird, wobei die Flüsse entsprechend zugehörige Warteschlangen erzeugen, wobei ein Traffic von jedem der Flüsse, der in dem Knoten eingegeben wird, in eine entsprechende Warteschlange eingefügt wird, und den Flüssen Token zugeordnet wird; wobei die mehreren Flüsse synchrone Flüsse ( $h = 1, 2, \dots, N_S$ ) umfassen, die eine garantierte minimale Dienstleistungsrate erfordern, und asynchrone Flüsse ( $i = 1, 2, \dots, N_A$ ) umfassen, die zum Ausnutzen der Dienstleistungskapazität der Ressource bestimmt sind, die ungenutzt durch die synchronen Flüsse verblieben ist,  
**dadurch gekennzeichnet, dass** es einen Server (10) umfasst, der ausgebildet ist, um die entsprechenden Warteschlangen, die zu den Flüssen (h, i) in aufeinanderfolgenden Arbeitsgängen zugehörig sind, in Augenschein zu nehmen, und Systemmittel umfasst, die ausgebildet sind, um das Verfahren nach einem der Ansprüche 1 bis 8 auszuführen.

## 55 Revendications

1. Procédure de programmation, dans un noeud de réseau de communication par paquets, d'une ressource de service partagée entre de multiples flux de paquets d'informations, lesdits flux générant des files d'attente associées res-

pectives, où le trafic de chacun desdits flux entrés dans ledit noeud est inséré dans une file d'attente respective, et étant desservis par l'attribution d'un jeton ; cette pluralité de flux inclut des flux synchrones ( $h = 1, 2, \dots, N_S$ ) , qui requièrent un taux de service minimum garanti, et des flux asynchrones ( $i = 1, 2, \dots, N_A$ ) destinés à exploiter la capacité de service de ladite ressource laissée inutilisée par les flux synchrones,

5 **caractérisée par le fait qu'elle inclut les opérations suivantes :**

- fournir un serveur (10) qui visite les files d'attente respectives associées auxdits flux ( $h, i$ ) en cycles successifs, en déterminant une valeur de temps de la rotation attendue (TTRT), qui identifie le temps nécessaire pour que le serveur (10) achève un cycle de visite sur lesdites files d'attente respectives,
- 10 - associer à chaque flux synchrone ( $h$ ) une valeur de capacité synchrone ( $H_h$ ) respective indicative de la quantité de temps maximum pour laquelle un flux synchrone peut être desservi avant de renoncer au jeton,
- associer à chaque flux asynchrone ( $i$ ) une première valeur de délai respective (*lateness (i)*) qui identifie la valeur qui doit être constituée pour que la file d'attente respective ait le droit d'être desservie, et une seconde valeur (*last token time*) qui indique l'instant auquel le serveur (10) a visité la file d'attente respective dans le cycle précédent, en déterminant pour ladite file d'attente respective, le temps qui s'est écoulé depuis la dernière visite du serveur (10),
- desservir chaque file d'attente associée à un flux synchrone ( $h$ ) pour un temps de service maximum égal à ladite valeur respective de capacité synchrone ( $H_h$ ) , et
- 20 - desservir chaque file d'attente associée à un flux asynchrone ( $i$ ) uniquement si la visite du serveur (10) se produit avant l'instant attendu, ladite avance étant déterminée comme la différence entre ladite valeur de temps de la rotation attendue (TTRT) et le temps qui s'est écoulé depuis la dernière visite du serveur (10) et le retard accumulé ; dans l'affirmative, cette différence définit le temps de service maximum pour chacune desdites files d'attente.

25 2. Procédure selon la revendication 1, **caractérisée par le fait qu'elle inclut**, si la file d'attente est vide lorsque le serveur la visite, l'opération qui amène le serveur (10) à visiter une file d'attente suivante même lorsque le temps de service maximum relatif s'est écoulé.

30 3. Procédure selon la revendication 1 ou la revendication 2, **caractérisée par le fait que**, dans le cas où ladite différence est négative, chacune desdites files d'attente associée à un flux asynchrone ( $i$ ) n'est pas desservie et la valeur de ladite différence est accumulée avec ledit délai.

35 4. Procédure selon l'une quelconque des revendications 1 à 3, **caractérisée par le fait que** ladite première valeur respective (*lateness(i)*) et ladite seconde valeur respective (*last token time*) sont initialisées à zéro.

5. Procédure selon l'une quelconque des revendications 1 à 4, **caractérisée par le fait que** ladite valeur de capacité synchrone ( $H_h$ ) respective est déterminée en proportion à ladite valeur de temps de la rotation attendue (TTRT).

40 6. Procédure selon la revendication 5, **caractérisée par le fait que** ladite valeur de capacité synchrone ( $H_h$ ) est déterminée en proportion à ladite valeur de temps de la rotation attendue (TTRT) selon un facteur de proportionnalité ( $C_h$ ) choisi en relation avec le facteur de proportionnalité ( $f_h$ ) respectif entre le taux de service demandé par le flux synchrone ( $h$ ) respectif et la qualité de service de ladite ressource partagée.

7. Procédure selon la revendication 6, **caractérisée par le fait que** ledit facteur de proportionnalité ( $C_h$ ) est choisi égal audit facteur de proportionnalité respectif.

45 8. Procédure selon la revendication 6, **caractérisée par le fait que** ledit facteur de proportionnalité ( $C_h$ ) est choisi en

50 se basant sur la formule suivante : 
$$C_h = \frac{N_A \cdot f_h}{N_A + 1 - \sum_{j=1}^{N_S} f_j} \quad \text{où :}$$

- $f_j$  représente ledit facteur de proportionnalité respectif relatif au j-ième flux synchrone, et
- $N_A$  est le nombre desdits flux asynchrones.

55 9. Système de programmation, dans un noeud de réseau de communication par paquets, d'une ressource de service partagée entre de multiples flux de paquets d'informations, lesdits flux générant des files d'attente associées respectives, où le trafic de chacun desdits flux entrés dans ledit noeud est inséré dans une file d'attente respective, et

**EP 1 329 065 B1**

étant desservis par l'attribution d'un jeton ; cette pluralité de flux inclut des flux synchrones ( $h = 1, 2, \dots, N_S$ ) , qui requièrent un taux de service minimum garanti, et des flux asynchrones ( $i = 1, 2, \dots, N_A$ ) destinés à exploiter la capacité de service de ladite ressource laissée inutilisée par les flux synchrones,  
5 **caractérisé par le fait qu'il** inclut un serveur (10) qui est configuré pour visiter les files d'attente respectives associées auxdits flux ( $h, i$ ) en cycles successifs, et le système est configuré pour réaliser les opérations de la procédure de l'une quelconque des revendications 1 à 8.

10

15

20

25

30

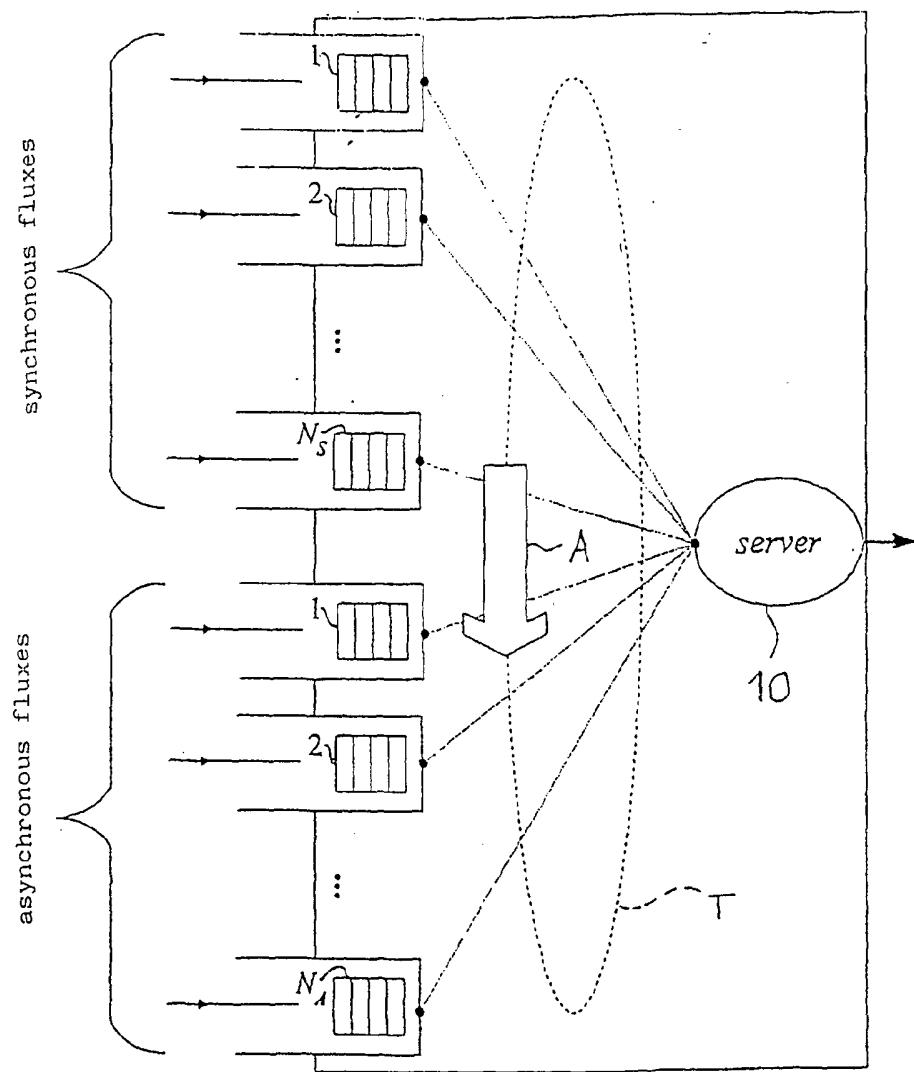
35

40

45

50

55



**REFERENCES CITED IN THE DESCRIPTION**

*This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.*

**Non-patent literature cited in the description**

- Delay analysis of the FDDI synchronous data class.  
**GENTER W L et al.** MULTIPLE FACETS OF INTEGRATION. SAN FRANCISCO. IEEE COMP. SOC. PRESS, 03 June 1990, vol. 2, 766-773 [0005]