



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
16.10.2002 Bulletin 2002/42

(51) Int Cl.7: **H04L 12/413**, H04L 12/42,
H04L 12/64, H04M 7/00

(21) Application number: **01440101.2**

(22) Date of filing: **10.04.2001**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR**
Designated Extension States:
AL LT LV MK RO SI

(72) Inventors:
• **Attimont, Luc**
78100 Saint Germain en Laye (FR)
• **Bodin, Jannick**
92380 Garches (FR)

(71) Applicant: **ALCATEL**
75008 Paris (FR)

(74) Representative: **Brose, Gerhard, Dipl.-Ing. et al**
Alcatel
Intellectual Property Department, Stuttgart
70430 Stuttgart (DE)

(54) **A method to ensure the quality of preferred communication services such as voice versus regular data**

(57) The invention relates to a method to ensure the quality of preferred (communications) services, especially voice services, in a local data network (LN) with different types of services, where data congestion or data collision on a shared transmission medium (TM) is treated by a collision detection method and by application of re-transmission rules after detection of data collisions, wherein depending on the type of service (e.g.

voice or data), the corresponding data packets are subject to different re-transmission rules and that data packets of a preferred type of service (i.e. voice) are subject to re-transmission rules, that ensure a preferred treatment against data packets of other types of services (regular data that is not time critical), a local network, a station, a local network controller and a program module therefor.

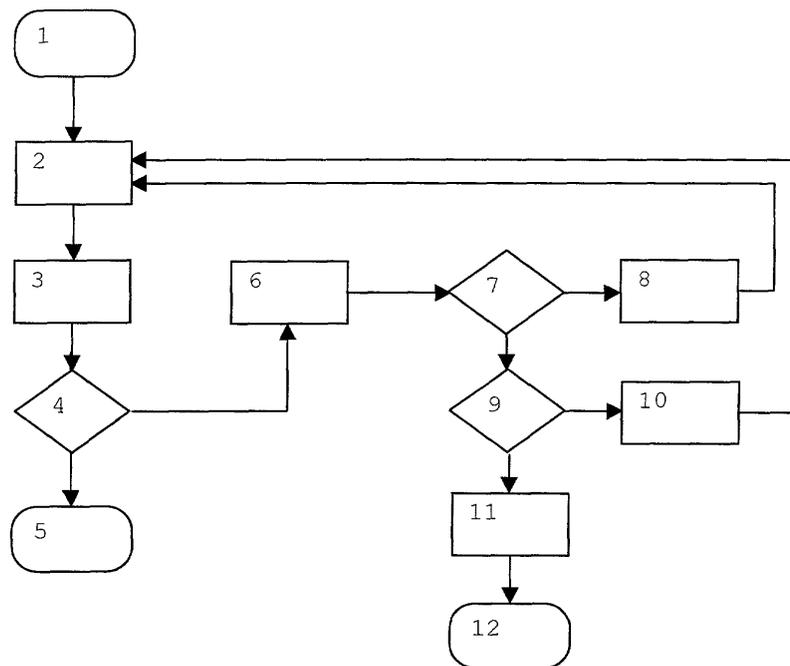


Figure 2

Description

[0001] The invention relates to a method to ensure the quality of preferred communication services according to the preamble of claim 1, as well as to a local network, a station, a local network controller and a program module therefor.

[0002] In Internet Telephony voice is transported over a network using data packets instead of circuit switched connections over voice only networks. Internet Telephony or IP Telephony refers to the transfer of voice over the Internet Protocol (VOIP) of the so-called TCP/IP (Transfer Control Protocol/Internet protocol) protocol suite.

[0003] In an advanced corporate enterprise, IP telephone terminals connect via a Local Area Network (LAN), often based on the so-called Ethernet, along with Personal Computers. These IP telephones interact with a VOIP-enabled private exchange (PBX) for call set-up and administration and access to the external (public switched) telephone network (PSTN) and/or an external packet network, especially the internet.

[0004] The transmission medium or bus of an Ethernet system is a shared medium, where transmission is not controlled by a central control unit, e.g. a bus master, but all terminals or stations connected follow certain rules, when transmitting data on the transmission medium. These rules are described in the document "IEEE Std 802.3, Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications". All stations are always listening the transmission medium for data traffic and they wait for sending data until said data traffic has ceased. If, at one time, two or more stations start sending data on the transmission medium, data collision occurs. Said stations, detecting the collision, stop sending and wait each for a specific random time until starting re-transmission. The detection of a collision may be carried out by supervision of the voltage of received signals. See following example: If the data signal sent on the bus shows two voltage levels, e.g. 0 Volt for a bit representing the value "0" and 5 Volt for a bit representing the value "1", a collision is detected in a receiving station as soon as said station sees a signal voltage, that exceeds the defined maximum voltage of the data signals, i.e. 5 Volt, caused by superposition of two or more data signals.

[0005] For a reliable and fast detection of collisions, any station detecting a collision may explicitly send out a specific information, a so-called jam signal. As soon as this jam signal is received by the sending stations, said stations stop sending. Without limiting the invention, it is assumed in the following, that an explicit jam signal is sent by the stations, when a collision is detected.

[0006] The more stations are connected to the transmission medium and the more these stations are actively sending data, the probability for data collision increases; it might happen, that one station has to start re-trans-

mission several times until data transmission can be successfully completed. For avoiding dead locks evoked by peak situations of bursty data traffic, the number of re-transmission trials can be limited. When exceeding this limit, the affected data packets are dropped.

[0007] In the following, only the Ethernet is considered for the two lowest layers of a local network, the data link level and the physical level of the so-called OSI reference model. Nevertheless, the invention can be carried out in any network with a shared (transmission) medium, where transmission on this medium is controlled by the stations themselves.

[0008] In existing local area networks based on Ethernet, generally all data is treated uniquely. The previously described mechanism of collision does not distinguish between different types of data. But more and more, additionally to "normal data", e.g. data to be visualised on a display, real time data traffic, especially voice (data) traffic for voice over IP applications, is to be treated by local area networks.

[0009] Real time communication applications exhibit strict latency requirements. A lot of real time communication application require, that time delay between sending and receiving data, must not exceed a defined time value. Voice over IP applications, e.g., require a maximum delay of about 0,2 seconds. Data exceeding a defined delay time is dropped, i.e. is not used for communication. To guarantee high quality real time communication services, the drop rate, i.e. the ratio between the number of dropped data packets and the number of transmitted data packets, has to be kept low.

[0010] For economic reasons, it is required, to both handle normal data traffic and real time data traffic in a unique local network. But today no method of different treatment of both types of data in local networks with a shared medium exists.

[0011] The object of the invention is to ensure the quality of real time or preferred communication services in a local network.

[0012] This object is achieved in accordance with the invention by a method according to the teaching of claim 1, a local network according to the teaching of claim 7, a station according to the teaching of claim 8, a local network controller according to the teaching of claim 9 and a program module according to the teaching of claim 10.

[0013] The basic principle of the invention is that the quality of preferred communication services in a local network with different services is ensured by distinguishing between different types of services. After detection of data congestion or data collision, depending on the type of service, the corresponding data packets are subject to different re-transmission rules to ensure a preferred treatment of data belonging to a preferred type of service.

[0014] Further developments of the invention can be gathered from the dependent claims and the following

description.

[0015] In the following the invention will be explained further making reference to the attached drawings in which:

Figure 1 shows a local network according to the invention connected to a further network,

Figure 2 shows an exemplary flow chart diagram to be executed in a station according to the invention and

Figure 3 shows an example for data encapsulation in the internet protocol layer and the Ethernet layer.

[0016] Figure 1 schematically shows a local (area) network LN and a further network IPN. This network might be any data packet network, e.g. the internet, an internet access network or an intranet. In the following, the further network IPN will be considered to be the internet IPN. The local network LN shows a first station T1, a second station T2, a gateway GW and a transmission medium or bus TM. The stations T1 and T2 and the Gateway GW are connected to the bus TM. The Gateway GW connects the local network LN to the Internet IPN. For both stations T1 and T2, the first station T1 exemplary shows a central processor CPU and a local network controller LC connected to each the central processor CPU and the bus TM. The local network controller LC exemplary shows, in a layered structure, a (data) decapsulation and encapsulation unit DED, an access controller MAC and a physical layer unit PHL. A memory unit MEM is shown in the decapsulation and encapsulation unit DED.

[0017] The functions of the local network controller LC is explained by the following transmission scenario:

[0018] Sending digital information from the first station T1 to the second station T2, the central processing unit CPU generates a so-called (IP-)datagram out of said digital information forming a data packet and adding a (packet) header comprising control information, e.g. a transmission control protocol (TCP) header. If necessary, said information is divided into several data packets generating one datagram out of each packet. The datagram(s) is/are transferred to the memory unit MEM of the decapsulation and encapsulation unit DED. The decapsulation and encapsulation unit DED, encapsulates each datagram, i.e. for each datagram it generates a frame with a destination address field, a source address field, a length information field, a data field and a frame check field and, if necessary, further additional fields. The source address field indicates the physical address of the sending station T1, the destination address field indicates the physical address of the receiving station T2, the length field indicates the frame length, the data field contains said datagram and the frame check field contains a cyclic redundancy code to check

the correctness of the received frame in the receiving station T2. The length field alternatively may be used as a type field containing a code or value bigger than the maximum possible frame length, that indicates a protocol carried by the Ethernet, e.g. a value 0800H (hexadecimal) may indicate the internet protocol. Said frame is handled over to the access controller MAC, that transforms the frame data into a serial bit stream. This serial bit stream is handed over to the physical layer unit PHL, that transforms the bit stream into an electromagnetic or physical data signal to be transmitted over the bus TM.

[0019] On the receiving station T2, now in reverse direction, the (physical) data signal is re-transformed into a serial bit stream and the serial bit stream is re-transformed into frames. For each frame, the frame check field is checked for correctness of the received frame and the destination address field is checked, whether the frame is intended for the receiving station T2. If the received frame is recognised to be correct and to be intended for said station, each datagram included in one frame is separated from the remaining frame and handed over to the central processing unit CPU.

[0020] If information is to be sent outside the local network LN, e.g. from the first station T1 to the internet, the data is sent to the gateway GW. This gateway GW terminates the local network protocol extracting the IP datagrams out of the corresponding frames and sending the IP datagrams to the internet INT, using the required protocol of the internet INT.

[0021] The bus TM is a shared medium used by all sending stations to transmit data to receiving stations. As described in the beginning, there is no central bus function managing the access to the bus TM. Collisions of data signals simultaneously sent to the bus TM by different stations must be managed by the stations themselves. It is one function of the physical layer unit PHL of each station (T1, T2) to always listen to the bus and to inform the access controller MAC, whether data signals are detected or not. When the bus is considered to be free, the access controller MAC of the sending station T1 starts to send a bit stream, that is converted into a data signal and sent to the bus TM by the physical layer unit PHL. As soon as a data signal of another station is detected, the physical layer unit PHL informs the access controller MAC, that immediately stops sending this data, and instead of said data, sends a jam signal. Subsequently, the access controller MAC wait for a certain time, the so-called backoff time, until retransmission of the affected frame is started. For each collision incident a different backoff time is determined randomly, thus stations having sent conflicting data signals, normally do not wait for the same backoff time to retransmit their data.

[0022] With an increasing number of active stations and increasing bus load, the probability for collisions increases. Additionally, the probability increases, that a station has to initiate retransmission of one datagram

several times in sequence. Normally, the backoff time of subsequent collision incidences is increased each time to react against increasing collision incidents. This method is well adapted for normal data to be transmitted. In those applications, a low packet loss ratio is much more important than a low delay time. If congestion on the bus occurs, an increasing delay time is acceptable. For real time data however, an increasing delay time causes problems. As described in the beginning, real time data, especially voice data, is very sensitive to delay. A local network (LN) with mixed normal data and real time data must provide a special mechanism to guarantee the quality of real time services.

[0023] In Figure 2 a flow chart diagram of an exemplary functional sequence for sending a data packet to be executed in a local network controller LC according to the invention by way of example is shown.

[0024] In Step 1, a data packet is prepared for sending on the bus TM.

[0025] In the following step 2, it is waited until the bus is quiet, i.e. no signals are detected.

[0026] In the following step 3, the prepared data is sent to the bus TM.

[0027] In the following step 4, a decision is taken. If a collision is detected, data transmission is stopped and step 6 executed next. If no collision is detected, transmission of the data packet is completed before a transfer is made to step 5.

[0028] In step 5, the transmission is successfully completed. A new data packet restarting with step 1 may be transmitted. In step 6, a jam signal is send out to inform all other sending stations.

[0029] In the following step 7, a further decision is taken. If the data packet is recognized as normal data packet a transfer is made to step 8. If the data packet is recognized as real time data packet, a transfer is made to step 9.

[0030] In step 8, a first backoff time is awaited and then a transfer is made back to step 2.

[0031] In step 9, a further decision is taken. If a defined maximum time or a defined number of subsequent retransmission attempts is exceeded, a transfer is made to step 11. Otherwise a transfer is made to step 10.

[0032] In step 10, a second backoff time is awaited and then a transfer is made back to step 2.

[0033] In step 11, the data packet is dropped.

[0034] In step 12, (unsuccessful) transmission of the data packet is completed. A new data packet restarting with step 1 may be transmitted.

[0035] The first backoff time of step 8 is recalculated each time. Known in the prior art, this time is calculated by a pseudo random algorithm. The mean backoff time starts to be relatively short for a first retransmission attempt of a data packet. Each subsequent retransmission attempt, the mean backoff time is increased. The increase of the mean time often follows an exponential curve. Not shown in the flow chart diagram, the data packet might be dropped after a certain number of (un-

successful) retransmission attempts. This method is widely used in Ethernet local networks.

[0036] The second backoff time of step 10 is also calculated randomly each time, but with shorter mean than the first backoff time. This is to prefer real time data packets. In order to achieve a more deterministic behavior (i.e. to be able to initiate at least one or several retransmission attempts within the maximum allowed time delay), the variance is much smaller than the variance of the random process for the first backoff time. Moreover there is no increase of the mean time or at least the increase is smaller than the increase of the first backoff time. Alternatively or additionally, the maximum value for the second backoff time is limited. With this method it is ensured or at least very likely, that after a collision of normal data with real time data, the station sending the real time data will initiate retransmission earlier than other stations sending normal data. If congestion on the bus is that high, that even real time data is blocked, e.g. if a lot of stations simultaneously try to send real time data, the respective data packets are dropped much earlier than other data packets. The real time packet is either dropped if said data packet could not be successfully transmitted in a defined maximum time (i.e. the maximum allowed delay time) or after a limited number of retransmission attempts. Said limited number might be calculated on the base of the mean backoff time or the minimum second backoff time. Whereas normal data is transmitted in a so-called best effort mode, minimizing the loss ratio, real time data is now transmitted in an expedited delivery mode, but the corresponding real time packets are early dropped.

[0037] In a first embodiment to distinguish normal data packets and real time data packets, the central processing unit CPU informs the local network controller LC about the type of service, when transferring a datagram.

[0038] Local network controllers (LN) or Ethernet controllers often have an intelligent Direct Memory Access Circuit (DMA). A direct memory access technique allows a bigger amount of datagrams to be transferred directly between the bus and the local network controller (LN), without any participation of the central processing unit CPU. An array or a chained circular buffer of command blocks to be handled subsequently provides a dialog between the central processing unit CPU and the local network controller LC. In each command block it is indicated in special fields, e.g. whether the command block is valid, whether the command block is ready for handling and whether the corresponding frame was successfully emitted. To inform the local network controller LC about the type of service, a field is added in the command block to specify for the corresponding frame the type of service. Additionally or alternatively, the maximum number of retransmission attempts or the maximum transmission attempt time for this data is added.

[0039] In an alternative embodiment, no additional information from the central processing unit CPU is nec-

essary. For explanation of this embodiment, Fig.3 shows an example for data encapsulation in the internet protocol layer and the Ethernet layer. An Ethernet Frame ETF consists of an Ethernet header ETH, an (IP-) datagram and an Ethernet Trailer ETT. The Ethernet header ETH contains different fields, like a preamble, address fields, a frame length field. The Ethernet trailer ETT consist of a control field described under Figure 1. The Ethernet frame ETF corresponds to the data link layer, the second level of the OSI reference model. The Datagram IPD refers to the network layer, the second level of the OSI reference model. The Datagram IPD consists of a part of data D and an IP-header IPH. The IP header consists of several data fields. One of these data fields is the so-called type-of-service field TOS. The other fields are not discussed here.

[0040] Applying the so-called DiffServ Protocol defined in the document "RFC 2474" of the Internet Engineering Task Force (IETF), the type-of-service field TOS indicates whether the corresponding datagram IPD is to be forwarded either in the mode "expedited delivery, dropable" or in the mode "best effort, low drop rate". The local network controller LC now autonomously chooses the requested mode, by analyzing the type-of-service field TOS of the datagram.

[0041] In a further embodiment, more than two different types of data are processed. The different types are either indicated by communication between the central processing unit CPU and the local network controller LC or by the type-of-service field TOS of the datagrams. In a system with e.g. three different priority levels, the highest priority might be assigned to voice data, that is most sensitive to delays. The second highest priority might be assigned to other real time data and the lowest priority might be assigned to normal data. Data with the highest priority is forwarded very expedite and is dropped very early. Data with the second highest priority is forwarded expedite and is early dropped. The normal data is forwarded with best effort and dropped late.

[0042] In a further embodiment, the local network controller LC re-orders frames to be sent, if two or more frames are stored in the emission queue and if frames belong to different types of service. The frames are re-ordered according to their priority, i.e. frames with the highest priority or the lowest maximum allowed delay time are preferred and sent first.

Claims

1. A method to ensure the quality of preferred (communications) services, especially voice services, in a local data network (LN) with different types of services, where data congestion or data collision on a shared transmission medium (TM) is treated by a collision detection method and by application of re-transmission rules after detection of data collisions, **characterised in, that** depending on the type of

service, the corresponding data packets are subject to different re-transmission rules and that data packets of a preferred type of service are subject to retransmission rules, that ensure a preferred treatment against data packets of other types of services.

2. A method according to claim 1, **characterised in, that** the different re-transmission rules each define a different backoff time or a different algorithm to calculate the backoff time, that a station (T1, T2) has to wait after a collision detection, before it re-transmits a data packet affected by the collision.
3. A method according to claim 1, **characterised in, that** the different re-transmission rules each define a different number of re-transmission attempts, before a data packet, that is several times consecutively affected by collisions, is to be dropped.
4. A method according to claim 1, **characterised in, that** the re-transmission rule for data packets of a preferred type of service defines a maximum time interval and that if a data packet can not be successfully transmitted in said time interval, said data packet is dropped.
5. A method according to claim 1, **characterised in, that** a local network controller (LC) of a station (T1, T2), responsible for the application of an appropriate retransmission rule, derives the appropriate rule for a datagram by data additionally received from the station (T1, T2).
6. A method according to claim 1, **characterised in, that** a local network controller (LC) of a station(T1, T2), responsible for the application of an appropriate retransmission rule, derives the appropriate rule for a datagram by analysing a defined information field (TOS) of said datagram.
7. A local network (LN) with stations (T1, T2) connected to a shared transmission medium (TM), with the following means:
 - means to exchange data packets between different stations (T1, T2),
 - means detect collisions of data signals on the transmission medium (TM),
 - means to distinguish between data packets of different types of services and
 - means to apply different re-transmission rules for data packets of different types of services.
8. A station (T1,T2) of a local network (LN), with an interface to a shared transmission medium (TM), with the following means:

- means to transform data packets to (physical) data signals and to send said data signals on the transmission medium (TM),
- means to detect collisions of data signals on the transmission medium (TM), 5
- means to distinguish between data packets of different types of services and
- means to apply a re-transmission rule appropriate for data packets of a preferred type of services. 10

9. A local network controller (LC) of a station (T1, T2) of a local network (LN), with an interface to a shared transmission medium (TM), with the following means: 15

- means to receive data packets from a central processing unit of the station (T1/ T2), to transform said data packets to data signals and to send said data signals on the transmission medium (TM), 20
- means to analyse the received data packets or corresponding additional information to distinguish between data of different types of services, 25
- means to detect collisions of data signals on the transmission medium (TM) and
- means to apply different re-transmission rules for data packets of different types of services. 30

10. A program module to be executed in a local network controller (LC), with means to control the following steps:

- receiving a data packets from a central processing unit (CPU) of a station (T1, T2) and transform said data packets to data signals, 35
- sending said data signals on the transmission medium (TM) when the transmission medium is quiet, 40
- stop sending, if a collision is detected and
- depending on the type of service of a data packet, an appropriate re-transmission rule of different re-transmission rules is applied. 45

50

55

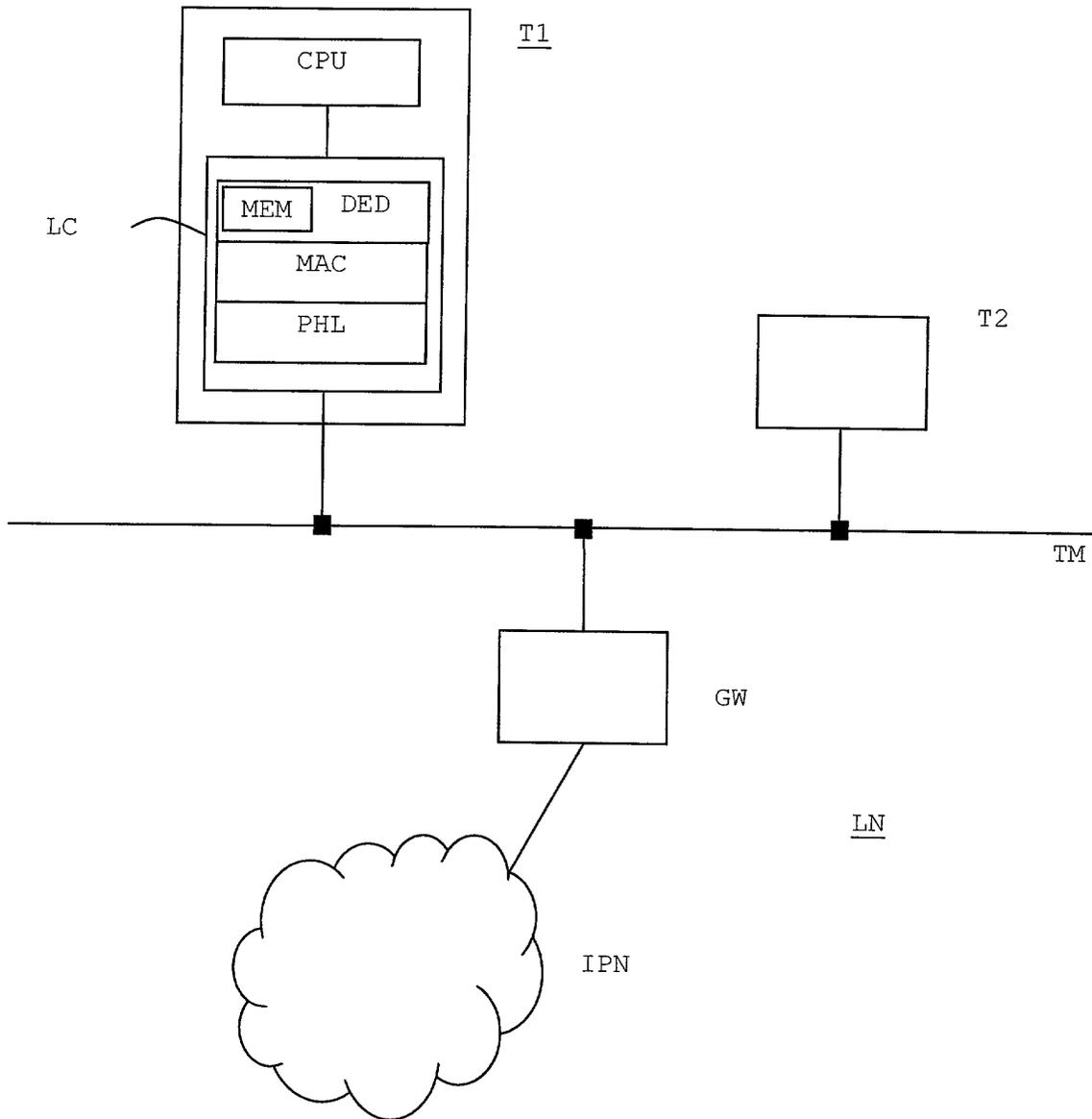


Figure 1

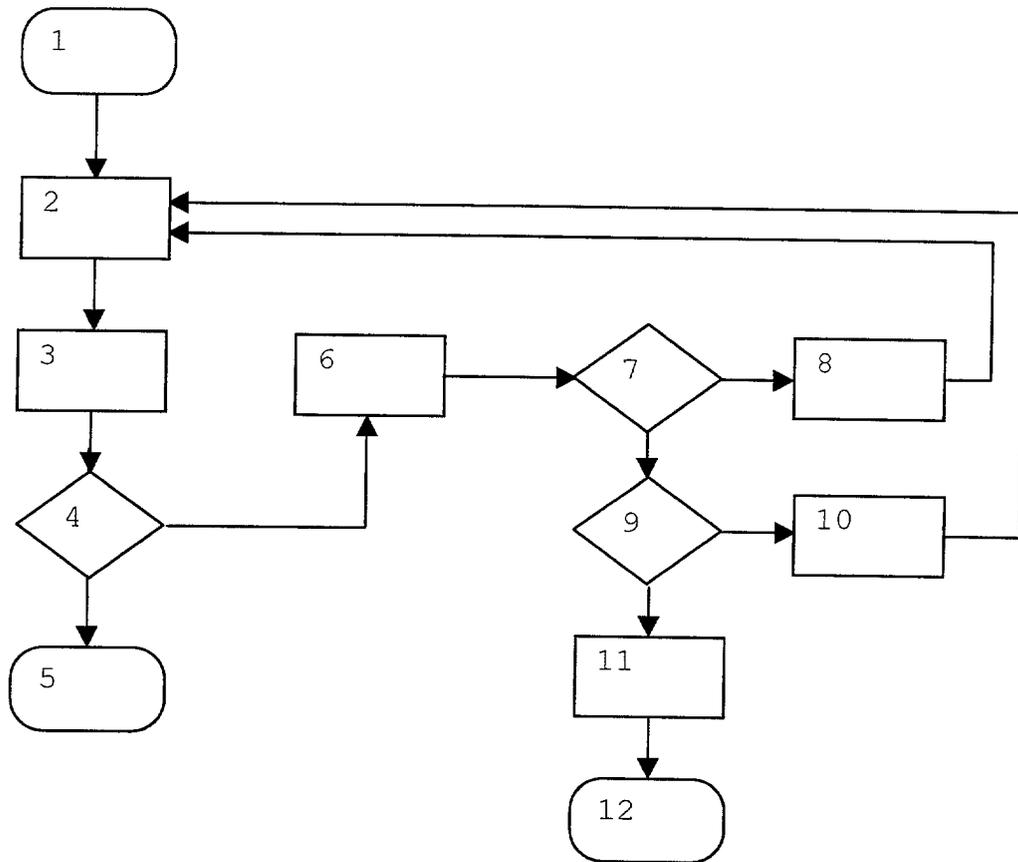


Figure 2

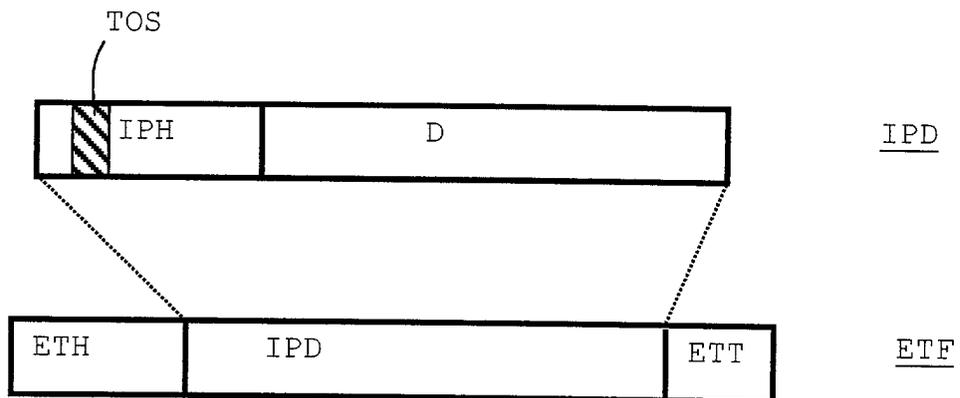


Figure 3



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 01 44 0101

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	EP 1 017 198 A (NORTEL NETWORKS CORP) 5 July 2000 (2000-07-05) * column 1, line 33 - column 6, line 33 * ----	1-4,7-10	H04L12/413 H04L12/42 H04L12/64 H04M7/00
X	ULUSOY O: "Network access protocol for hard real-time communication systems" COMPUTER COMMUNICATIONS, ELSEVIER SCIENCE PUBLISHERS BV, AMSTERDAM, NL, vol. 18, no. 12, 1 December 1995 (1995-12-01), pages 943-948, XP004032425 ISSN: 0140-3664 * the whole document * -----	1-10	
X	JIA F ET AL: "The superchannel scheme for integrated services on multiple access broadcast networks" COMPUTER NETWORKS AND ISDN SYSTEMS, NORTH HOLLAND PUBLISHING. AMSTERDAM, NL, vol. 27, no. 11, 1 October 1995 (1995-10-01), pages 1523-1543, XP004002502 ISSN: 0169-7552 * page 1523, left-hand column - page 1530, left-hand column, last paragraph * -----	1-10	
X	US 6 172 984 B1 (SHAFFER SHMUEL ET AL) 9 January 2001 (2001-01-09) * figures 3,4 * * column 1, line 15 - column 7, line 25 * -----	1-10	
The present search report has been drawn up for all claims			
Place of search MUNICH		Date of completion of the search 9 October 2001	Examiner Nash, M
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document</p>			

EPO FORM 1503 03/02 (P04/C01)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 01 44 0101

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

09-10-2001

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
EP 1017198	A	05-07-2000	EP	1017198 A2	05-07-2000
US 6172984	B1	09-01-2001	US	5960001 A	28-09-1999

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82