(12) PATENT (11) Application No. AU 200229278 B2
(19) AUSTRALIAN PATENT OFFICE (10) Patent No. 784771

(54) Title
An access process

(51)[6] International Patent Classification(s)
H04L 29/02 (2006.01) 20060101AFI2006040
H04L 29/02 8BMEP

(21) Application No: 200229278 (22) Application Date: 2002 .03 .28

(30) Priority Data

(31) Number	(32) Date	(33) Country
PR4113	2001 .03 .30	AU

(43) Publication Date : 2002 .10 .03
(43) Publication Journal Date : 2002 .10 .03
(44) Accepted Journal Date : 2006 .06 .15

(71) Applicant(s)
The University of Melbourne

(72) Inventor(s)
Chuan Heng Foh; Moshe Zukerman

(74) Agent/Attorney
Davies Collison Cave,Level 15,1 Nicholson Street,MELBOURNE VIC 3000

(56) Related Art
US 2004/0153525
US 2003/0212794

ABSTRACT:

A process for accessing a channel of a communications network, the process executed by a node of the network, and including the steps of:

(a) generating request data;

(b) sending, to the channel, a request for access to the channel, the request including the request data and address data for the node;

(c) monitoring the channel for requests, including the request;

(d) selecting a request from the requests on the basis of request data of the requests; and

(e) accessing the channel if the address data of the selected request corresponds to the node.

P/00/011
Regulation 3.2

AUSTRALIA

Patents Act 1990

COMPLETE SPECIFICATION

STANDARD PATENT

(ORIGINAL)

Name of Applicant: **The University of Melbourne**, of Grattan Street, Parkville, Victoria 3052, Australia

Actual Inventors: Chuan Heng FOH
Moshe ZUKERMAN

Address for Service: **DAVIES COLLISON CAVE,** Patent Attorneys, of 1 Little Collins Street, Melbourne, Victoria 3000, Australia

Invention Title: **An access process**

Details of Associated Provisional Application: PR4113/01

The following statement is a full description of this invention, including the best method of performing it known to us:-

AN ACCESS PROCESS

FIELD OF THE INVENTION

The present invention relates to an access process for communications networks, and in particular to a process for accessing a channel of a communications network.

BACKGROUND

A multiple access protocol specifies how a broadcast channel of a communications network is allocated to competing users who wish to send data on that channel. In the framework of the open systems interconnection (OSI) reference model for networks, multiple access protocols are implemented in a sublayer known as the Medium Access Control (MAC) layer within the data link layer. Multiple access protocols are used in radio-frequency networks, such as cellular telecommunications networks, and also in local area networks (LANs). The challenge in developing a MAC layer protocol for networks is to establish a simple and efficient protocol, and particularly to ensure that its efficiency is not degraded to an unacceptable degree as the number of users accessing the broadcast channel increases. A MAC layer protocol ideally minimises collisions between nodes whose broadcasts overlap, whilst maximising the amount of data transmitted on the channel. One method of reducing the probability of collisions is for each node to check that the channel is idle before attempting to transmit data on it. Such a protocol is known as a carrier sense protocol. The efficiency of the protocol may be further improved if the transmission is aborted as soon the transmitting node senses a collision. Such a protocol is known as carrier sense multiple access (CSMA) with carrier detect (CD), or CSMA/CD. The IEEE 802.3 standard for Ethernet networks defines a version of this protocol.

Other protocols, such as IEEE 802.14, use an intelligent Central Controller (CC) to receive requests for bandwidth from a multiplicity of nodes. After receiving the requests, the CC transmits scheduling information to the nodes, which then transmit their data frames without collision. Other reservation protocols are based on a distributed control principle. Examples are the IEEE 802.5 token ring and the IEEE 802.6 Distributed Queue Dual Bus

(DQDB). These protocols achieve collision free transmission at the cost of node complexity.

The CSMA/CD protocol has been retained by the IEEE 802.3z working group as the MAC layer protocol for channel assignment in Gigabit Ethernet networks. However, collisions may not be detected by the transmitting node in a high-speed network if the frame transmission time is less than the propagation delay of the network. Due to the high data rate of these networks, to achieve backward compatibility and guarantee the proper operation of CSMA/CD, the IEEE 802.3z working group has introduced a *carrier extension* operation. If a data frame is too short for collision detection purposes, nodes append predefined carrier signals to the short data frame for a period of time that is long enough for collision detection. Another modification to the protocol is the increase of slot time by almost an order of magnitude, from 512-bit time in 10/100 Mb/s Ethernet to 4096-bit time for Gigabit Ethernet. Consequently, each collision in a Gigabit Ethernet network results in a loss of 10 times more data than in 10 or 100Mb/s Ethernet networks. Moreover, the utilisation efficiency of this protocol degrades as the number of users attempting to access the broadcast channel increases. Furthermore, the IEEE 802.3z protocol does not support service differentiation, whereby different priorities may be assigned to different kinds of services. Service differentiation is important to ensure that delay-sensitive services such as voice-over-IP and video-on-demand are not degraded as the network becomes more congested.

It is desired, therefore, to provide a process for accessing a channel of a communications network that alleviates the above difficulties, or provide at least a useful alternative to existing access processes.

SUMMARY OF THE INVENTION

The present invention provides a process for accessing a channel of a communications network, said process executed by a node of said network, and including the steps of:

(a) generating request data;

(b) sending, to said channel, a request for access to said channel, said request including said request data and address data for said node;

(c) monitoring said channel for requests, including said request;

(d) selecting a request from said requests on the basis of request data of said requests; and

(e) accessing said channel if the address data of the selected request corresponds to said node.

The present invention also provides a process for operating a communications network having a repeater and a plurality of nodes, including the steps of:

(a) sending a request signal for channel access from one of said nodes to said repeater;

(b) sending said request signal from said repeater to each of said plurality of nodes, including the one node; and

(c) receiving said request signal at said one node, and accessing said channel in response thereto.

The present invention also provides a process for accessing a channel of a communications network having a repeater, including the steps of:

(a) sending a request for access to said channel from a node of said network to said repeater;

(b) monitoring said channel for requests, including said request; and

(c) accessing said channel on the basis of said request.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the present invention are hereinafter described, by way of example only, with reference to the accompanying drawings, wherein:

Figure 1 is a schematic diagram of a preferred embodiment of a gigabit ethernet network;

Figure 2 is a schematic diagram of a finite state machine representing an access process executed by nodes of the network;

Figure 3 is a schematic diagram showing the structures of frames broadcast on the network;

Figure 4 is a schematic diagram showing the relative timing of events of the access process; and

Figure 5 is a graph showing the throughput of the access process and a prior art IEEE 802.3z process as a function of the number of nodes in the network.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A Gigabit Ethernet local area network, as shown in Figure 1, includes nodes 4 to 18 arranged in a star configuration and interconnected by fiber optic cables 2. The nodes 4 to 18 comprise data terminal equipment (DTE) such as personal computers, network servers, and printers, each with at least one network interface controller (NIC) or transceiver, and may further include interfaces to other networks. Each of the cable segments 2 shown in Figure 1 comprises a pair of optical fibers, one for each signal direction. The network also includes a passive hub or repeater 20 that receives signals broadcast from nodes 4 to 18 on incoming fibers of the cables 2, and repeats them to the outgoing fibers of the cables 2. For example, the passive repeater 20 may be a 10base-FP star, such as a CodeStar passive hub manufactured by Data Base Access Systems, Inc. Due to the presence of the repeater 20, the cables 2 constitute a single broadcast channel, because a signal sent by any one of the nodes 4 to 18 to the repeater 20 is retransmitted by the repeater 20 on all of the cables 2 and is received by all of the nodes 4 to 18, including the node that sent the signal. Access to the broadcast channel is determined by a request contention multiple access (RCMA) MAC layer process executed by each node 4 to 18 of the network, and described in detail below. In this context, access is for the purpose of transmitting user or system data and the transmission of request data to the channel in order to request access to the channel is not itself regarded as access. It will be appreciated by the skilled addressee that the access process can be executed by a network interface controller of each node, either alone, or in conjunction with one or more other processors of the node, and that the RCMA process may be executed as software code, or by dedicated hardware circuits such as application-specific integrated circuits (ASICs), or a combination of each.

The RCMA process or protocol makes use of the return signals repeated by the passive optical repeater 20 to allow a transmitting node to verify if its earlier transmission was successful. To improve the efficiency of the protocol, RCMA uses a very small request frame for each node contending for the channel access right to reserve the channel for longer data frame transmissions. The RCMA protocol is based on distributed control principle, and achieves efficient scheduling and fairness with low overhead, intelligence and complexity. It is more efficient than the IEEE 802.3z MAC protocol, and unlike the latter, the performance of RCMA remains stable as the number of nodes increases. Furthermore, RCMA can easily support service differentiation. In the description below, the word "node" is used to refer to a network view of a node. For example, when it is said that a "node" is "idle", this refers to a state of affairs wherein the NIC of the node is neither sending nor receiving data on the broadcast channel.

The nodes 4 to 18 of the network execute an RCMA process, as shown in the state diagram of Figure 2. Transitions between states S4 to S26 are indicated by state transition events E1 to E13 and give rise to timer events *S1 to *S3, as shown in Table 1, and described below

Table 1

State Transition Events	Timer Events
E1: ready E2: busy channel detected E3: RWTimer expired E4: transmission completed E5: RCTimer expired E6a: all requests checked, the channel is assigned to this station E6b: the channel is not assigned to this station E7: maximum request retry reached E8: an interframe gap period detected E9: ICTimer expired E10: data frame detected E11: NEXT frame detected E12a: NEXT frame checked, the channel is assigned to this station E12b: the channel is not assigned to this station E13: no data frame is waiting for transmission	*S1: reset and activate RWTimer *S2: reset and activate RCTimer *S3: reset and activate ICTimer

Initially, the network channel is idle, and a node 4 of the network is in an *idle state* S4. When the node 4 is ready for a data transmission, it is referred to as a *ready node*, and it performs a *request contention* operation. The node 4 first prepares a request frame 22, as shown in Figure 3. The request frame 22 includes preamble bits 21, a 1-byte start control frame delimiter (SCFD) 23, a 1-byte request number (RN), a 6-byte source address (SA) 26, and a 1-byte short frame check sequence (SFCS) 28 for error detection. The node 4 randomly generates a 6-bit *request number* and stores it in the request number (RN) field 24 of the request frame 22. Its MAC address is stored in the source address (SA) field 26 of the request frame 22.

Once the request frame 22 has been prepared, the node 4 enters E1 a *request waiting* state S6 and activates S1 a timer called the *request-waiting timer*, or RWTimer. The RWTimer value is set to $w \cdot T_s$, where w is a uniformly distributed random integer between zero and k-1, where k is a parameter of the protocol, and T_s is the minislot time, which is the time required to transmit the request frame 22 plus a short guard time. The guard time is included to ensure that the minislot time is not less than the transmit time, for example, in case a node's clock is running slightly faster than those of other nodes. After activating S1 the RWTimer, the node 4 monitors its incoming optical fiber. Detection E2 of a request or data frame generated by another node during the timer period causes the node 4 to abort its pending request and enter a *channel monitoring* state S7. This is because if the channel carries a request frame that was transmitted earlier by another node, that node is given access priority, based on a 'first come first served' principle.

However, if the incoming channel remains idle when the RWTimer expires E3, the node 4 enters a *request sending* state S8, and transmits its request frame on the broadcast channel. During transmission, the node 4 continues to monitor the incoming channel. If a carrier is detected on this channel, the request frame transmission is aborted immediately. The cables 2 of the network are long enough so that a request frame will not return back to the originating node whilst the latter is transmitting that request frame. However, request frame transmissions are subject to collisions. If two nodes transmit the request almost at the same time, so that the request frames meet at the passive repeater 20, these request frames are corrupted due to the overlapping signals. Otherwise, the request frames are considered successfully transmitted and can be read correctly.

The maximum signal propagation delay between the transmission and reception of a signal broadcast on the network cables 2, *e.g.*, the time it takes a signal sent from a node at the end of the longest cable segment 2 to reach the repeater 20, be retransmitted back along the outgoing fiber of each cable segment 2, and reach the same node, has a value of τ seconds. However, in the preferred embodiment, the cable segments 2 are of equal length. After the very last bit of the request frame has been transmitted E4, the node 4 enters a *request collecting* state S10, and activates S2 another timer called the *request-collection timer*, or RCTimer. This timer is set to 2τ plus a short guard time. During this time interval, the node 4 monitors its incoming channel and collects any request frame, including its own request frame, transmitted earlier. Any incorrect or incomplete request frames are discarded. When RCTimer expires E5, the node 4 can be sure that all transmitted request frames have arrived and no further request frames are still propagating in the network. The node 4 then enters a *request checking* state S12, and compares all the received requests. The request with the largest request number is the *winning request*. The node that originally sent the winning request is given the channel access right, and will henceforth be called the *winner*.

All nodes contending for the channel access right, including the winner, can identify the winner by comparing the request numbers, and identifying the MAC address of the winning node from the source address field 26 of the request frame with the largest request number. It is possible that two or more nodes may choose the same request number. In this case, the node with the larger numerical MAC address is the winner. This does not affect the fairness significantly because this event is very rare.

The winning node now enters E6a a *data frame sending* state S14, and sends its data frame on the outgoing fiber. The data frame structure used on the network is the IEEE 802.3 frame 30, as shown in Figure 3. Unlike CSMA/CD, no collision detection is required during data frame 30 transmission. Any other nodes who have pending requests enter E6b a *request pending* state S18, and when they detect E2 that the channel is busy, enter a *channel monitoring* state S7. Similarly, nodes with no data to send, and are in the *idle state* S4, also detect E2 that the channel is busy, and also enter the *channel monitoring* state S7. When nodes in this state S7 detect E10 the winning node's data frame 30, they enter a *data frame receiving* state S22.

After the data frame has been sent and an *interframe gap* (IFG) period has passed, the winning node enters E4 a *NEXT frame sending* state S16, and the requesting nodes enter E8 a *channeling monitoring* state S24. The winning node constructs and transmits a control frame 32 referred to as a NEXT frame, as shown in Figure 3, containing a list 34 of request numbers and corresponding source addresses from the other access requests collected by the node during its *request collecting* state S10, sorted by request number. The list 34 is delimited at either end by a NEXT frame indicator byte (NFI) 25. After the NEXT frame 32 is transmitted E4, or if no NEXT frame 32 was transmitted because no other requests were collected, the winning node enters the *idle state* S4. If a NEXT frame 32 was sent, the remaining nodes with pending requests enter a *non-contention channel assignment* operation.

When the pending nodes detect E11 the winning node's NEXT frame 32, they enter a *NEXT frame checking* state S26, in which each node compares its MAC address with the first MAC address 36 in the NEXT frame 32. The node whose MAC address matches the source address 36 in the NEXT frame 32 enters E12a the *data frame sending* state S14 and becomes the winning node, and the remaining nodes enter E12b the *request pending* state S18. The winning node sends its data frame, removes its record from the NEXT frame 32, transmits the modified NEXT frame 32, and enters the *idle state* S4.

When the last node in the list 34 in the NEXT frame 32 completes its data frame transmission, no NEXT frame is sent. After an IFG period has passed E8, nodes in the *channel monitoring* state S24 enter the request contention operation by returning to the *request waiting* state S6. Nodes without data to send return E13 to the *idle state* S4, and the entire process continues as described above.

If all the transmitted request frames 22 collide, no data frame 30 transmission will occur. After discovering that there is no data frame 30 transmission, all ready nodes immediately repeat the request contention operation to compete for the channel access right. Specifically, nodes whose pending request was aborted, and are in the *channel monitoring* state S7, will detect E8 an IFG, and enter the *request pending* state S18. Nodes who sent a request frame 22 and are in the *request checking* state S12 will also enter E6b the *request pending* state S18.

An important aspect of RCMA is its implicit channel assignment property. If the assigned node fails to initiate its transmit, a deadlock situation could potentially occur. To avoid this problem, when the channel is assumed to be assigned to a winning node, each node in the *request pending* state S18 activates a timer called *idle-channel timer*, or ICTimer. The duration of this timer is greater than the duration of RCTimer. The ICTimer is reset if the incoming channel is sensed busy, and the nodes enter the *channel monitoring* state S7. Otherwise, if the incoming channel remains idle after the ICTimer expires E9, the winning node forfeits its transmission right, and the nodes return to the *request waiting* state S6. However, if the number of transmissions of a request frame from a node exceeds a maximum attempt number, the channel access fails and the node returns E7 to the *idle state* S4. Each ready node then repeats the request contention operation to compete for the channel access right. Nodes without data to send return E13 to the *idle state* S4.

The timing of events in the RCMA process is illustrated in Figure 4. After a data frame is sent by a winning node, a period of the propagation delay τ + the IFG passes before every node has detected the end of the data frame transmission. If another IFG period passes without a NEXT frame being detected, *i.e.*, a total time of $\tau + 2 *$ IFG after the data frame is transmitted, the nodes detect 51 the end of all transmission for this cycle 48, and return to the *request waiting* state S6.

In an alternative embodiment, the RCMA protocol described above is extended to support service differentiation by dividing possible request numbers into two or more groups, *e.g.*, (i) a higher group; and (ii) a lower group. A higher priority service may choose its request number from the higher group to ensure that it will have the channel transmit right before a lower priority service, provided that its request frame is successfully transmitted. In another alternative embodiment, the performance of RCMA is improved by allowing the transmission of several data frames with one request, similar to the *frame bursting* operation in the IEEE 802.3z standard.

In one variant of Gigabit Ethernet, buffered distributors replace repeaters to achieve better performance, as described in J. Kadambi, I. Crayford and M. Kalkunte, "Gigabit Ethernet: Migrating to High-Bandwidth LANs," *Prentice Hall*, 1998. In a further alternative embodiment, the passive optical repeater 20 of the RCMA network is replaced by a buffered distributor without replacing RCMA transceivers in the network. As in Gigabit Ethernet, the buffered distributor acts as a central controller, collecting request frames and scheduling data frame transmissions according to the collected requests. However, the overhead of channel assignment in RCMA is lower than that of standard Gigabit Ethernet due to the non-contention channel assignment operation of RCMA.

For very high data rate LANs such as 10Gb/s LANs, where the time required to transmit an IEEE 802.3 frame 30 becomes very short relative to the propagation delay, reserving the channel for such a short transmission may not be efficient. In yet another alternative embodiment, a small data frame is encapsulated within an RCMA request frame. If the request frame transmission does not collide with other request frames, then the data frame transmission is also collision-free. To support this feature, nodes exclude requests that contain data frames when comparing collected requests, because no further data transmission is required for those requests.

The performance of RCMA can be simulated and compared to the simulated performance of IEEE 802.3z. For example, consider a network of m nodes where the distance between any two nodes is the same, and each node is 'saturated', that is, it always has data to transmit. Consequently, the event E1 shown in Figure 2 occurs as soon as any node enters the *idle state* S4, and the node will therefore enter the *request waiting* state S6.

Consider a realistic data frame size distribution, wherein 35% of the data frames 30 carry 46 bytes of useful information and the remaining 65% carry 1500 bytes of useful information, corresponding to the minimum and maximum sizes of IEEE 802.3 frames 30. The RCMA channel is subject to a cyclic series of event periods, as shown in Figure 4. Each cycle 48 has an *I*-period 40, an *R*-period 42, a *C*-period 44, and a *D*-period 46, representing idle, request transmission, request collection and data frame transmission periods respectively. The *I*-period 40 begins 51 as soon as the previous *D*-period 46 ends. In the *I*-period 40, all ready nodes, including the node that transmitted a data frame, enter the *request waiting* state S6. A node in this state S6 transmits its request when its RWTimer expires. As soon as the first transmitted request frame appears on the channel, the *I*-period 40 ends and the *R*-period 42 begins. During the *R*-period 42, each node is not aware of request frames transmitted by other nodes, hence whenever any node's RWTimer expires, it transmits its request frame 22. When the first bit of the first request frame 22 reaches 52 all nodes, the *R*-period 42 ends, and the C-period 44 begins.

When the C-period 44 begins, no further request frames 22 are transmitted. During this period, each node that sent a request resets and starts its RCTimer and collects requests on the channel. When a node's RCTimer expires, it checks its collected requests to determine the winner. Any node that is not the winning node resets and starts 53 (*S3) its ICTimer. The winning node initiates 54 its data frame transmission. At some point 56 during this transmission, the ICTimers of the other nodes expire; these nodes detect the busy channel and do not attempt to access it. The C-period 44 ends when the winning node begins 54 its data frame transmission. Due to the non-contention channel assignment operation during the D-period 46, several data frames can be transmitted by the winning node. The D-period 46 ends when there is no NEXT frame transmission after a data frame transmission, as described above.

Let B be the data rate of the network. Given m saturated nodes, let the random variables I, R, C, D be the duration of the I-period 40, the R-period 42, the C-period 44 and the D-period 46, respectively. Let the random variable U be the duration of the actual data transmission, excluding all IEEE 802.3 frame overheads during a cycle 48, and the random variable H be the duration of the overhead transmission such that $D=H+U$. Then the RCMA saturation throughput for m saturated nodes, S_{RCMA}, can be expressed by:

$$S_{RCMA} = \frac{E[U]}{E[I+R+C+H+U]} \quad , \quad (1)$$

where E[ ] represents the average value of the parameter within the square parentheses. As described above, RWTimer=$w \cdot T_s$, where w is a uniformly distributed random integer between zero and k-1, and T_s is the minislot time duration. The I-period 40 ends when at least one request frame 22 appears on the channel. Hence the probability that the I-period 40 lasts for x minislots is the probability that any of the m nodes choose to transmit their request frames given that no request frame transmission appears in previous minislots. The probability distribution function (PDF) of I is thus:

$$P\{I = x \cdot T_s\} = \begin{cases} q_x & , x = 0 \\ q_x\left(1 - \sum_{i=0}^{x-1} P\{I = i \cdot T_s\}\right) & , x = 1,2,\ldots,k-1 \\ 0 & , x = k \end{cases} \quad , (2)$$

where $q_x = 1-\left(1-\frac{1}{k-x}\right)^m$ is the probability that any of the m nodes chooses to transmit its request frame after x idle minislots.

For the R-period 42, since the distance between any two nodes is fixed, the duration for a signal to propagate from any node to all nodes is constant. Thus:

$$R = \tau \quad (3)$$

When the R-period 42 ends, no further request frames are transmitted. Since the R-period 42 is constant, then the number of minislots, r, within the R-period 42 is also a constant, and is given by:

$$r = \tau / T_s \qquad (4)$$

The duration of the C-period 44 depends on the position of the winner within the r minislots. If $k<r$, then the winner will appear within the k minislots instead. Recall that the winner is determined from the request numbers of all the successful requests. Since the request number is randomly and uniformly chosen between zero and k-1 by each node, hence if there is at least one successful request during the R-period 42, the probability that the winner will appear at the xth minislot in the R-period 42, making the C-period 44 to be ($\tau+x\cdot T_s$) unit of time, is similar, except when the winner appears at the first minislot in the R-period 42. This is because if the winner does not appear at the first minislot in the R-period 42, the first minislot is not idle. Thus the possibility that no nodes pick the first minislot, causing the first minislot to be idle, is excluded if the winner does not appear at the first minislot in the R-period 42. With this condition, the PDF of the C-period 44 is approximately:

$$P\{C = x\cdot T_s + \tau\} = \begin{cases} \alpha & , x = 1 \\ \alpha - \left(\frac{n-1}{n}\right)^m & , x = 2,3,\ldots,n \\ \text{where } n = \min(r,k) & \end{cases} \qquad ,(5)$$

where α can be obtained by summing up the probability for x=1,2,...,min(r,k) and equating the sum to unity.

Given m saturated nodes, r minislots and the k parameter, the PDF of the number of request frames successfully detected by all nodes during the R-period 42, N, can be derived recursively to be:

$$P\{N = x\} = \frac{N_b(x,r,k,m)}{N_a(k,m)}, x = 0,1,\ldots,r \qquad (6)$$

where

$$\begin{aligned} N_b(x,r,k,m) = &\binom{m}{0} N_b(x,r,k-1,m) \\ &+ \binom{m}{1} N_c(x-1,r-1,k-1,m-1) \\ &+ \sum_{n=2}^{m} \binom{m}{n} N_c(x,r-1,k-1,m-n), \end{aligned}$$

$$\begin{aligned} N_c(x,r,k,m) = &\binom{m}{0} N_c(x,r-1,k-1,m) \\ &+ \binom{m}{1} N_c(x-1,r-1,k-1,m-1) \\ &+ \sum_{n=2}^{m} \binom{m}{n} N_c(x,r-1,k-1,m-n), \end{aligned}$$

$N_a(k,m) = k^m$, with $\binom{m}{n} = \frac{m!}{n!(m-n!)}$ and the following initial conditions:

$N_b(x=-1,r,k,m) = 0;$
$N_b(x=0,r,k \neq 1,m=0) = 1;$
$N_b(x=0,r,k=1,m=1) = 0; N_b(x=0,r,k=1,m \neq 1) = 1;$
$N_b(x=1,r,k \neq 1,m=0) = 0;$
$N_b(x=1,r,k=1,m=1) = 1; N_b(x=1,r,k=1,m \neq 1) = 0;$
$N_b(x \geq 2,r,k \neq 1,m=0) = 0; N_b(x \geq 2,r,k=1,m) = 0;$

and

$N_c(x=-1,r,k,m) = 0;$
$N_c(x=0,r=0,k,m) = N_a(k,m);$
$N_c(x=0,r,k \neq 1,m=0) = 1;$
$N_c(x=0,r,k=1,m=1) = 0; N_c(x=0,r,k=1,m \neq 1) = 1;$
$N_c(x=1,r=0,k,m) = 0;$
$N_c(x=1,r,k \neq 1,m=0) = 0;$
$N_c(x=1,r,k=1,m=1) = 1; N_c(x=1,r,k=1,m \neq 1) = 0;$
$N_c(x \geq 2,r=0,k,m) = 0;$
$N_c(x \geq 2,r,k \neq 1,m=0) = 0; N_c(x \geq 2,r,k=1,m) = 0.$

where $N_c(x,r,k,m)$ is the total number of possible permutations, that x out of r minislots will carry successful requests, given m and k. $N_b(x,r,k,m)$ is similar to $N_c(x,r,k,m)$ but $N_b(x,r,k,m)$ is the number of possible permutations under the assumption that no idle slot appears in any of the previous minislots. $N_a(k,m)$ is the total number of possible permutations given m and k.

The duration of the D-period 46 depends on the number of successful requests that appear in the R-period given in Equation 6. If there were no successful requests, all ready nodes will enter the *request pending* state S18 due to events E6b and E8. Not all nodes discover the failure of channel assignment at the same time, but the difference between the time each node enters the *request pending* state S18 is insignificant. Therefore, all nodes may be considered to return to the *request pending* state S18 at the same time after the C-period 44 ends. In the case where there is no winner, if ICTimer lasts for 2τ, then it will take a duration of 2τ before this cycle 48 ends. With this assumption, the relationship between the number of successful requests, N, obtained in Equation 6, and the duration of the D-period 46 is:

$$D = \begin{cases} 2\tau & ,N = 0 \\ E[T_{FRAME}] + \tau + 2 \cdot T_{IFG} & ,N = 1 \\ \sum_{i=1}^{N-1} \left(E[T_{FRAME}] + T_{IFG} + T_{NEXT}(i) + \tau \right) & \\ \quad + E[T_{FRAME}] + \tau + 2 \cdot T_{IFG} & ,N = 2,3,\ldots,r \end{cases} \quad ,(7)$$

with $T_{NEXT}(i) = 8 \cdot (10 + 7i)/B$, and where T_{FRAME}, T_{NEXT} and T_{IFG} are the transmission time of the IEEE 802.3 frame, the NEXT frame, and the IFG duration, respectively. Knowing the distribution of a data frame, the mean of D can be computed.

The time duration of useful information transmitted during a cycle 48, U, also depends on N in Equation 6, and may be expressed as :

$$U = N \cdot E[T_u],\ N = 0,1,\ldots,r \quad ,(8)$$

where T_u is the transmission time of the useful information. Having obtained the PDF of I, R, C, D, and U, their mean values can be computed, as well as the saturation throughput of RCMA given in Equation 1.

The saturation throughput of Ethernet has been determined in C. Foh and M. Zukerman, "Performance Comparison of CSMA/RI and CSMA/CD with BEB," to appear in *Proceedings of IEEE ICC 2001* ("Foh"). Some modifications of the calculations in this paper are made below to include the carrier extension operation of Gigabit Ethernet.

From Foh, the saturation throughput of IEEE 802.3z, S_{CSMA}, is:

$$S_{CSMA} = \frac{E[U_{CSMA}]}{E[I_{CSMA} + C_{CSMA} + H_{CSMA} + U_{CSMA}]} , (9)$$

where the random variables I_{CSMA}, C_{CSMA}, H_{CSMA}, U_{CSMA} are the idle, contention, overhead transmission, and useful information transmission periods respectively in CSMA/CD. Let D_{CSMA} be the frame transmission period including the overhead, thus $D_{CSMA}=H_{CSMA}+U_{CSMA}$. By Foh:

$$\begin{aligned} E[I_{CSMA}] &= 0 \\ E[C_{CSMA}] &= (L_m - 1)\cdot T_{SCSMA}. \end{aligned} , (10)$$

where L_m is the mean number of slots required to resolve a collision caused by m nodes and to obtain a successful transmission in CSMA/CD, and T_{SCSMA} is the slot time.

The mean values of D_{CSMA} and U_{CSMA} are:

$$\begin{aligned} E[D_{CSMA}] &= E[T_{FRAME}] + E[T_{CARRIER}] + T_{IFG} + \tau \\ E[U_{CSMA}] &= E[T_u] \end{aligned} , (11)$$

where T_{IFG}, $T_{CARRIER}$, T_{FRAME} and T_u are the duration of the IFG, the duration of carrier extension, the IEEE 802.3 frame transmission time, and the useful information transmission time, respectively. By substituting Equations 10 and 11 into Equation 9, the saturation throughput of the IEEE 802.3z Gigabit Ethernet MAC protocol can be obtained.

The saturation throughputs of RCMA and IEEE 802.3z at 1Gb/s can be compared, using parameters from (i) IEEE 802.3/ISO 8802-3, "Information processing systems - Local area networks - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, 2nd edition," September 21, 1990, and (ii) J. Kadambi, I. Crayford and M. Kalkunte, "Gigabit Ethernet: Migrating to High-Bandwidth LANs," *Prentice Hall*, 1998, as shown in Table 2, for numerical computations and simulations.

Table 2

Parameters	Values
Data rate, B	1Gb/s
node numbers, m	1,2,...,50
Propagation delay, τ	2μsec
IEEE 802.3 frame overhead including preamble and SFD	0.208μsec (26 bytes)
Useful transmission duration for a short IEEE 802.3 frame	0.386μsec (46 bytes)
Useful transmission duration for a long IEEE 802.3 frame	12μsec (1.5kbytes)
IFG time duration, T_{IFG}	0.049μsec
Slot time in IEEE 802.3z, T_{SCSMA}	4.096μsec
minislot time duration in RCMA, T_s	0.128μsec (16 bytes)
minislot numbers in RCMA, r	15
Parameter k for RCMA	20

In addition, 35% of the data frames 30 are taken to be short frames, with the remaining 65% being long frames. The mean data frame transmission time and the useful information transmission time for RCMA and IEEE 802.3z are shown in Table 3. In IEEE 802.3z, if the data frame transmission duration (excluding preamble bits and start frame delimiter (SFD)) is less than a slot time, the transmission will be extended with carriers until the duration of a slot time is reached. E[$T_{CARRIER}$] represents the average time wasted due to carrier extension for each data frame transmission with the assumed data frame distribution.

Table 3

Mean frame transmission time, E[T_{FRAME}]	8.1368 μsec
Mean useful information transmission time, E[T_u]	7.9288 μsec
Mean duration of carrier extension in IEEE 802.3z, E[$T_{CARRIER}$]	1.2544 μsec

The saturation throughput 60 of RCMA and the saturation throughput 62 of IEEE 802.3z are shown in Figure 5. Analytical and simulation results are shown with solid lines and symbols respectively. The saturation throughput 62 of IEEE 802.3z drops significantly when the number of saturated nodes increased from one to five, and continues to drop as the number of saturated nodes increases. The throughput 62 even drops below 10% when there are over 32 saturated nodes sharing the 1Gb/s bandwidth. That is, each saturated node only receives around 3.125Mb/s bandwidth on average under these conditions.

In contrast, the saturation throughput 60 of RCMA is stable, and offers over 65% efficiency for up to 50 nodes, except when there are fewer than three saturated nodes. This is because when the number of saturated nodes is low, the channel assignment overhead for each data frame transmission is slightly higher due to the need for requests prior to data frame transmission. However, as the number of saturated nodes increases, the non-contention channel assignment operation of RCMA becomes effective, more data frame transmissions can be assigned during a request contention period, and therefore the channel assignment overhead for each data frame transmission becomes relatively small. In the case of 32 saturated nodes, RCMA achieves around 70% throughput, which is equivalent to a 21.875 Mb/s bandwidth for each node on average, seven times higher than that in the IEEE 802.3z protocol.

The IEEE 802.3z protocol performs better than RCMA only when there is exactly one saturated node in the network. In this case, no collisions occur in the IEEE 802.3z protocol. As the number of saturated nodes increases, collisions become more likely, and the throughput of the IEEE 802.3z MAC protocol drops significantly.

Many modifications will be apparent to those skilled in the art without departing from the scope of the present invention as herein described with reference to the accompanying drawings.

Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

The reference to any prior art in this specification is not, and should not be taken as, an acknowledgment or any form of suggestion that that prior art forms part of the common general knowledge in Australia.

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A process for accessing a channel of a communications network, said process executed by a node of said network, and including the steps of:
   (a) generating request data;
   (b) sending, to said channel, a request for access to said channel, said request including said request data and address data for said node;
   (c) monitoring said channel for requests, including said request;
   (d) selecting a request from said requests on the basis of request data of said requests; and
   (e) accessing said channel if the address data of the selected request corresponds to said node.

2. A process as claimed in claim 1, wherein said step of generating includes generating random request data.

3. A process as claimed in claim 2, wherein said random request data includes a random request number.

4. A process as claimed in claim 1, including, before step (b), the step (f) of monitoring said channel for request data sent by other nodes, and executing step (b) only if request data is not detected.

5. A process as claimed in claim 4, wherein the monitoring step (f) is performed over a randomly determined period of time, said period being an integer multiple of a minislot time, wherein said minislot time is substantially equal to the time required to send a request.

6. A process as claimed in claim 1, wherein step (b) includes simultaneously monitoring said channel for request data broadcast by other nodes, and aborting said step if said request data is detected.

7. A process as claimed in claim 1, wherein if step (c) fails to detect any requests, said process returns to step (b).

8. A process as claimed in claim 1, wherein said monitoring step (c) is performed over a monitoring period of time at least twice as long as the maximum propagation delay of said network.

9. A process as claimed in claim 3, wherein said selecting step (d) includes selecting a request with the greatest request number.

10. A process as claimed in claim 9, wherein if two or more requests share the same greatest request number, a request is chosen from said two or more requests on the basis of address data of said two or more requests.

11. A process as claimed in claim 1, wherein said address data includes a MAC address of said node.

12. A process as claimed in claim 3, including, after step (e), a further step (g) of sending said requests except for said selected request.

13. A process as claimed in claim 12, wherein the sent requests are sorted by request number and are sent in a control frame.

14. A process as claimed in claim 8, including, after step (e), the further steps of:
(h) monitoring said channel for data sent by another node of said network; and
(i) returning to step (c) if no data is detected.

15. A process as claimed in claim 1, wherein said monitoring step (h) is performed over a period of time exceeding said monitoring period.

16. A process as claimed in claim 1, including, after step (e), the further steps of:

(h) monitoring said channel for data sent by another node of said network;

(j) incrementing a request counter; and

(i) returning to step (c) if no data was detected and if the value of said request counter is less than a predetermined maximum request count value.

17. A process as claimed in claim 1, including the further steps of:

(j) monitoring said channel for a control frame containing at least one request;

(k) returning to step (c) if no control frame is detected;

(l) selecting a request from said control frame;

(m) returning to step (e)

18. A process as claimed in claim 17, wherein said monitoring step (j) is performed for a period substantially equal to an inter-frame gap of said network.

19. A process as claimed in claim 17, wherein said selecting step (l) selects the first request of said control frame.

20. A process as claimed in claim 1, wherein said generating step (a) includes generating request data on the basis of at least one of a priority of said node and a priority of user data to be sent during said access step (e).

21. A process as claimed in claim 20, wherein said request data includes a randomly generated number within a range of values corresponding to said priority.

22. A process for operating a communications network having a repeater and a plurality of nodes, including the steps of:

(a) sending a request signal for channel access from one of said nodes to said repeater;

(b) sending said request signal from said repeater to each of said plurality of nodes, including the one node; and

(c) receiving said request signal at said one node, and accessing said channel in response thereto.

23. A process as claimed in claim 22, wherein said process may also include the step (d) of checking the integrity of said signal.

24. A process as claimed in claim 23, wherein the length of the request signal is short relative to the length of a data transmission signal.

25. A process for accessing a channel of a communications network having a repeater, including the steps of:

(a) sending a request for access to said channel from a node of said network to said repeater;

(b) monitoring said channel for requests, including said request; and

(c) accessing said channel on the basis of said request.

26. A process as claimed in claim 25, including the additional step of sending said request from said repeater to each of said plurality of nodes, including said one node.

27. A process as claimed in claim 1, including, after step (c), the step (d) of checking the integrity of said request.

28. A network node having components for executing the steps of any one of claims 1 to 27.

29. A network transceiver having components for executing the steps of any one of claims 1 to 27.

30. A communications network, including a plurality of nodes having components for executing the steps of any one of claims 1 to 27.

31. A computer readable storage medium, having stored thereon program code for executing the steps of any one of claims 1 to 27.

32. A process for accessing a channel of a communications network, substantially as herein described with reference to the accompanying drawings.

33. A process for operating a communications network, substantially as herein described with reference to the accompanying drawings.

34. A network node, substantially as herein described with reference to the accompanying drawings.

35. A transceiver, substantially as herein described with reference to the accompanying drawings.

36. A communications network, substantially as herein described with reference to the accompanying drawings.

DATED this 5th day of April 2006

THE UNIVERSITY OF MELBOURNE
By its Patent Attorneys
DAVIES COLLISON CAVE

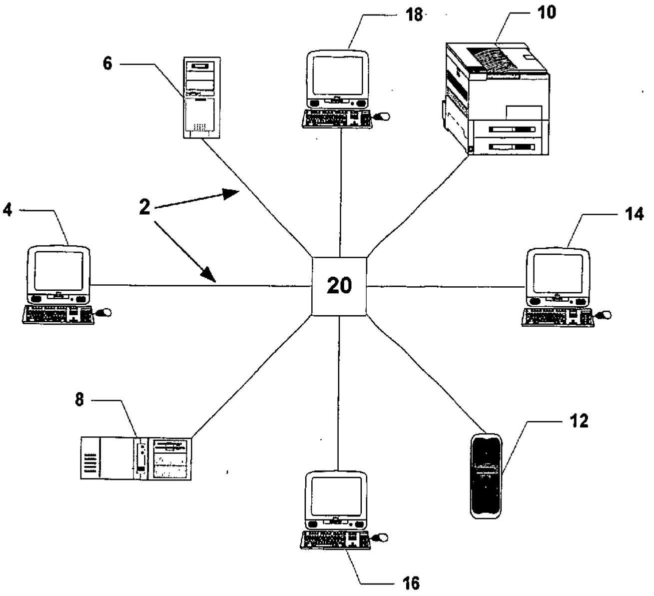


Figure 1

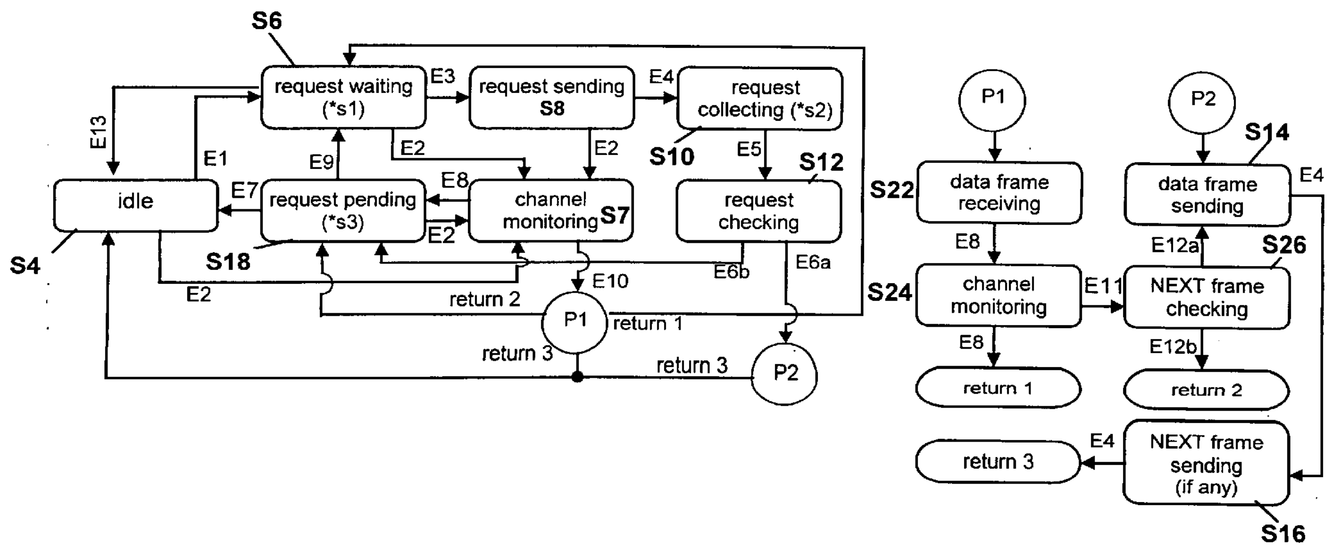

S6
request waiting (*s1)
E3
request sending S8
E4
request collecting (*s2)
S10
E5
S12
request checking
E6a
E6b
E13
E1
idle
S4
E7
E9
request pending (*s3)
S18
E2
E8
channel monitoring S7
E2
E2
E10
return 2
P1
return 1
return 3
return 3
P2
P1
S22
data frame receiving
E8
S24
channel monitoring
E8
return 1
E11
P2
S14
data frame sending
E4
E12a
S26
NEXT frame checking
E12b
return 2
E4
return 3
NEXT frame sending (if any)
S16

Figure 2

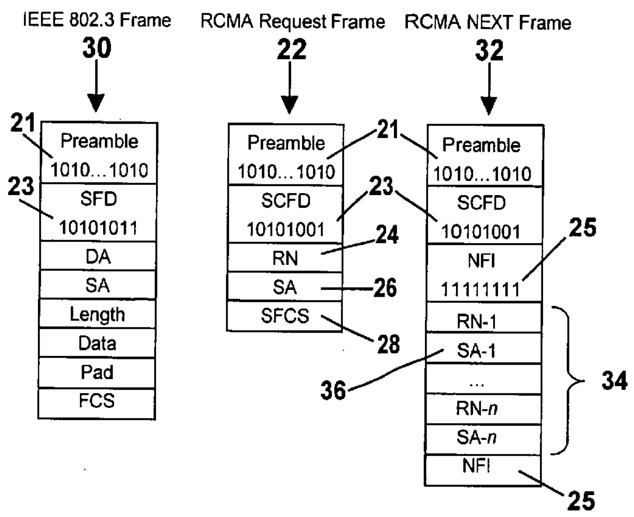

IEEE 802.3 Frame
RCMA Request Frame
RCMA NEXT Frame
30
22
32
21
23
Preamble
1010...1010
SFD
10101011
DA
SA
Length
Data
Pad
FCS
Preamble
1010...1010
SCFD
10101001
RN
SA
SFCS
24
26
28
36
Preamble
1010...1010
SCFD
10101001
NFI
11111111
RN-1
SA-1
...
RN-n
SA-n
NFI
25
34

Figure 3

28 3 02 29278

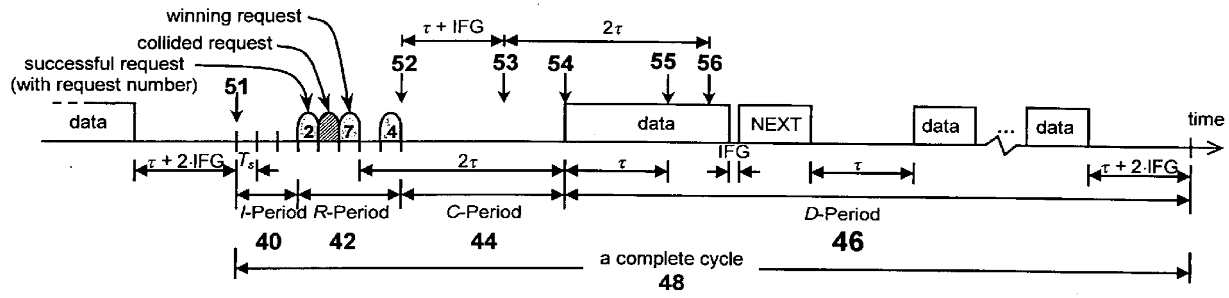
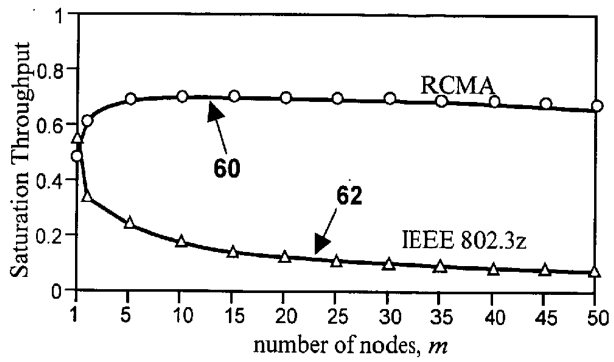


Figure 4



Saturation Throughput
1
0.8
0.6
0.4
0.2
0
RCMA
60
62
IEEE 802.3z
1
5
10
15
20
25
30
35
40
45
50
number of nodes, m

Figure 5