

Replacing the Ethernet access mechanism with the real-time access mechanism of Twentenet

Aiko Pras describes how a prioritized CSMA/CD access mechanism can be used on top of existing Ethernet hardware

The way in which a Local Area Network access mechanism (Medium Access Control protocol) designed for a specific type of physical service can be used on top of another type of physical service is discussed using a particular example. In the example, an Ethernet physical layer is used to provide service to the Twentenet real-time access mechanism. Relevant Ethernet and Twentenet concepts are explained, the approach taken is introduced, and problems encountered, along with the actual synthesis of both networks, are described.

Keywords: Local Area Networks, Ethernet, Twentenet, access mechanism, Medium Access Control, physical layer

ETHERNET

The first version of Ethernet was developed by the Digital Equipment Corporation (DEC), and the Intel and Xerox Corporations. A modified version of Ethernet was standardized by the Institute of Electrical and Electronics Engineers (IEEE) as the IEEE 802.3 standard. This standard was used as input to International Organization for Standardization (ISO) standardization efforts, resulting in the ISO 8802/3 International Standard (IS)^{1, 2}. This paper is based on the ISO 8802/3 IS, but the name Ethernet is still used throughout the text.

Faculty of Informatics, University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands

Figure 1 shows the relation between the Ethernet (10BASE5 version) implementation model² and the ISO Open Systems Interconnection (OSI) Reference Model³. Ethernet's physical and data link layers are discussed below.

Physical layer

The Ethernet physical layer performs coding for the following reasons:

- to ensure a sufficient number of transitions in the transmitted signal to enable receiving stations to extract clock information out of the received signal;
- and to provide a well known average DC level. This DC level is used by the carrier sense and collision detection functions.

Coding

The conversion of data bits into medium signals (and vice versa) is governed by the Manchester Biphase coding technique (an example is shown in Figure 2). Using this technique, each data bit is split into two parts with opposite polarity. Manchester Biphase coding therefore introduces a sufficient number of signal transitions to enable receiving stations to extract clock information.

The Low part of the signal is mapped upon -2.05 V, and the High part upon 0 V. Since transitions will always occur in the middle, the average Direct Current (DC) level is -1.025 V.

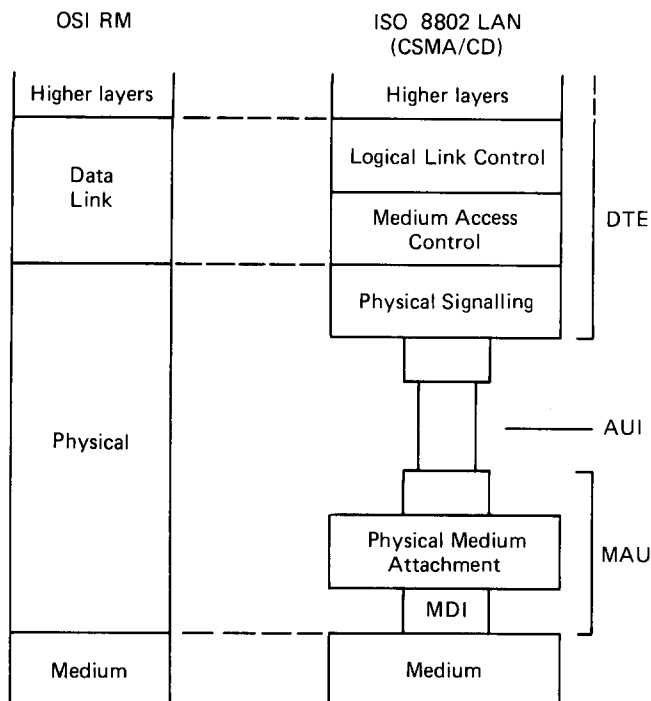


Figure 1. OSI Reference Model and Ethernet implementation model. AUI: Attachment Unit Interface; DTE: Data Terminal Equipment; MAU: Medium Attachment Unit; MDI: Medium Attachment Interface

Whenever a station has successfully completed the transmission of a data signal, an idle signal is transmitted. This idle signal is a continuous High signal (lasting at least 2 bit times), and is thus a Manchester Biphase violation. Note that this continuous High signal is mapped upon 0 V, and thus resembles inactivity.

Carrier sense

If no station transmits Manchester Biphase encoded signals over the medium, the DC level on the medium will be 0 V. As explained above, transmission of one Manchester Biphase encoded signal yields an average DC level of -1.025 V. Carrier sense may therefore be implemented by simply comparing DC levels.

Collision detection

If two (or more) stations start transmission at the same time, a collision occurs. Since the transmitting stations compare the data on the medium against the original data, the transmitting stations will eventually detect the collision.

As a result of a collision, an average DC level will be present on the medium; this DC level will usually be below the thresholds as defined for carrier sense. Non-transmitting stations may therefore implement collision detection by means of (again) level comparison.

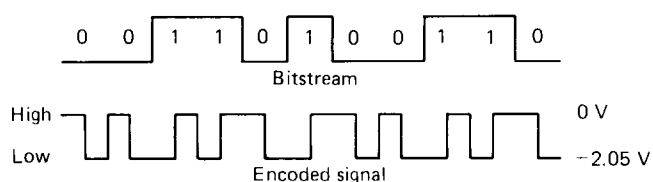


Figure 2. Manchester Biphase coding example

Non-transmitting stations may not always be able to detect collisions caused by only two transmitting stations. Depending on component tolerances, and on the location of the receiving and of both colliding stations, the average DC level on the medium may not drop below the threshold value at the location of the receiver (an example is shown in Figure 3). The Ethernet standard, therefore, does not require non-transmitting stations to detect all collisions.

Jabber function

Ethernet physical layer entities should implement the so-called 'jabber' function. This function defines that the transmission of long output messages should be interrupted whenever a specific time duration has been exceeded. This time duration should not be less than 20 ms or more than 150 ms, meaning that a message's length cannot exceed 200 Kbits.

Interface signals

A specification of the service, provided by the Ethernet physical layer to the data link layer, is given in ISO 8802/3². This specification defines, at a high level of abstraction, four primitives which are exchanged between the Physical Signalling (PLS) and the Medium Access Control (MAC) sublayer. These primitives are:

PLS Data Request: a data link entity starts the transmission of a data unit to all other stations connected to the medium by executing the PLS Data Request primitive. The physical service thus provides a broadcast service. Data Request primitives may be implemented by using three interface signals: the transmit clock; the data bits; and the data-valid signal.

PLS Data Indication: used to pass data that has been received by a physical entity from the physical layer to the data link layer. Data Indication primitives may be implemented by using two interface signals: the receive clock; and the data bits.

PLS Carrier Indication: executed whenever a system's physical entity detects valid data or signal quality errors (e.g. collisions) on the medium. This primitive does not distinguish between valid and invalid data — it merely signals to the user that 'something is happening' on the medium.

PLS Signal Indication: signals the occurrence of errors on the medium. An example of an error is a collision. The ISO 8802/3 standard specifies that each transmitting station should be capable of detecting the occurrence of a

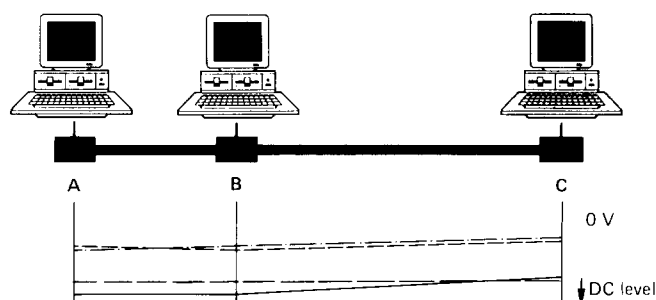


Figure 3. DC level on an Ethernet cable. — · · · · : DC level caused by A; - - - - - : DC level caused by B; — : sum of DC levels; — : collision detection threshold

collision. However, as already explained, non-transmitting stations do not need to detect all collisions.

Note that the PLS Data Indication primitive does not need a specific data-valid signal, since that information is provided by both the PLS Carrier Indication (Carrier = on) and the PLS Signal Indication (No Error).

Next to the above interface signals, a 'loop test' signal is available for test purposes. Whenever this signal is activated by the data link layer, the physical layer creates a local loop by connecting the station's output to the station's input. In this mode, no further data will be transmitted over the medium. This mode is useful for test purposes.

The Ethernet standard prescribes that a special test (called the 'heart beat test') should be performed by the physical entity after the successful transmission of a frame. The purpose of this test is to check the correct operation of the collision detection circuitry. The test is started by the generation of a 'signal quality error' by the Medium Attachment Unit (MAU). As a result, the Carrier Indication primitive signals to the data link layer the existence of a carrier (although the medium is idle), and the Signal Indication primitive indicates an error. Since this test is completed before the interframe gap period (lasting 9.6 μ s) is over, the test does not interfere with new transmission attempts made by the user.

Data link layer

The lower half of the Ethernet data link layer (the MAC layer) is primarily responsible for delimiting (frame-)synchronization and the access mechanism.

Delimiting and synchronization

If new data arrives at a station's MAU, the station may need time to detect the presence of this data and to adjust its receive clock. To avoid the loss of user data, a transmitting station will therefore always transmit a synchronization signal in advance of any user data. This synchronization signal is called a *preamble*.

Although it can be argued that maintenance of the receive clock is a physical protocol function, and thus initialization of this clock should be transparent to the physical layer user, the designers of Ethernet decided to make preamble generation a data link function.

Access mechanism

Ethernet has a so-called 1-persistent Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access mechanism. CSMA means that, before a station is allowed to start data transmission, the medium should be sensed. If a carrier is detected, transmission attempts should be deferred until the carrier disappears. After the medium is found to be idle, transmission attempts will start immediately after the interframe gap period of 9.6 μ s is over (1-persistent).

If two stations start to transmit at the same time, a collision will occur. This collision should, after some time, be detected by both transmitting stations, and as a consequence these stations should stop their transmissions. The time needed to detect this collision depends on a number of aspects such as the cable length between both transmitters, and the quality of the

components used. Without special precautions, the station that first detects the collision may stop transmitting before the other transmitter becomes aware of the collision. Since the collision's cause has now disappeared, the other station can no longer detect the interference, and it may erroneously tell its user that the transmission was successful.

To avoid such errors, the Ethernet CSMA/CD system requires that, after the detection of a collision, the transmitter will not stop until a jam signal, lasting at least 32 bits, has been transmitted. All implementations of Ethernet transmitters are required to detect the collision before this jam signal disappears.

After a collision is detected, the collision resolution mechanism prohibits new transmission attempts until a random period of time is over. After this period the medium should be sensed again, and only if no carrier is detected may the actual retransmission begin. This retransmission may again lead to a collision, in which case the entire procedure explained above is repeated. Figure 4 shows the Ethernet access mechanism.

Because of the random period of time each transmitter should wait before it can begin a retransmission attempt, it is impossible to predict which station is going to win the collision resolution phase. In an extreme case, all stations may begin their retransmission at the same time, in which

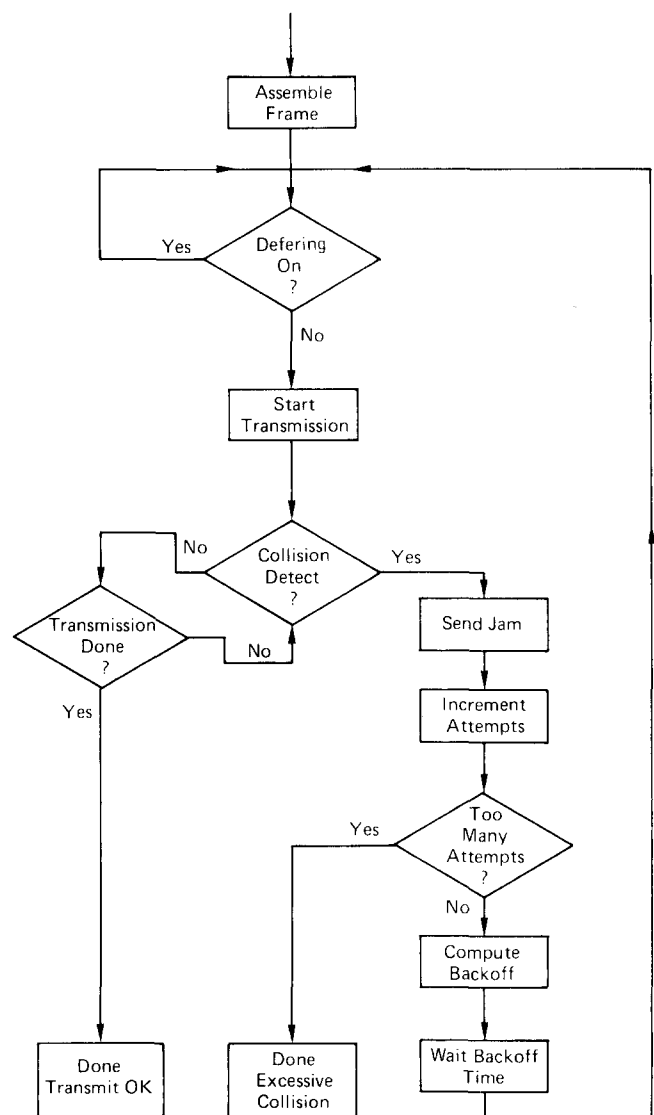


Figure 4. Ethernet access mechanism

case a new collision will result. This is why Ethernet's access mechanism shows a non-deterministic behaviour — such behaviour may not be acceptable in process control environments.

TWENTENET

The development of the Twentenet Local Area Network (LAN) began at the University of Twente in about 1980^{4, 5}. A number of prototypes for the VME and PC bus have been implemented to date.

A major design criterion was the requirement that the network should possess real-time characteristics, and should therefore be suited for industrial applications (for an overview of Twentenet's real-time capabilities see Reference 6). To guarantee the high reliability needed for industrial applications, no central active components were allowed, thus a bus topology was chosen. This topology ensures that, even if individual network resources fail, other stations can still communicate with each other. The maximum length of the bus is 2 km (no repeaters are needed), and 255 stations may be connected to the same bus.

The access mechanism is based on a CSMA/CD mechanism. In comparison with token access mechanisms, CSMA/CD mechanisms require a more sophisticated physical layer (since this layer should be capable of detecting collisions). This disadvantage however, is outweighed by the simplicity of the access mechanism itself, since token recovery procedures are not needed.

Physical layer

The Twentenet physical layer performs coding for the following reasons:

- to allow receiving stations to adjust their receive clocks (bit-synchronization);
- to avoid an average DC level on the medium; and
- to allow transmission of a special signal, called the beep. This signal is used to beat the alarm whenever a collision is detected, and ensures that *all* stations (not only those that are transmitting) detect the collision.

Coding

Twentenet uses the Partial Response 4 (PR-4) with precoding technique: the bitstream, generated by the data link layer, is converted by the physical layer into a three-level block signal with levels +V, 0 and -V volts. The coding rules are as follows:

- the original bitstream is divided in two imaginary bitstreams, one consisting of the *even-position* bits, and the other of the *odd-position* bits. Each bitstream is treated independently with respect to the next conversion rules;
- A 0 becomes: 0 volt,
A 1 becomes: +V volt if the last 1 in the same bitstream was converted to -V volt, -V volt if the last 1 in the same bitstream was converted to +V volt (Figure 5 gives an example).

To allow a receiver to synchronize his clock on the incoming data, the sender should guarantee the existence of a sufficient number of transitions in the transmitted

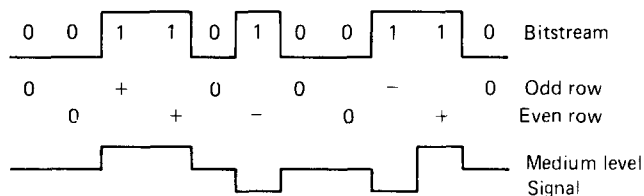


Figure 5. Partial Response 4 (PR-4) coding example

signal. The transmitting station will therefore precode the outgoing data in the following way: whenever five consecutive 0s have to be transmitted, the first and third 0 will be changed into a 1. To notify the receiver that these bits have been changed, the sender will transmit these 1s with a code violation. This code violation means that the 1 will be mapped upon the same level (instead of inverse) as the previous 1 in the same bitstream.

PR-4 coding is characterized by the fact that the average DC level is 0 V, and that the power dips exist at $n \times 1/2 f$ Hz ($n = 0, 1, 2, \dots$; f = the number of transmitted bits). Twentenet was designed for a 16 Mbit/s data rate, thus power dips exist at 0, 8, 16 Mhz, etc. These power dips are used to carry special information, such as the preamble (explained below) and the beep. The beep is a 16 Mhz block signal, with the high being mapped upon +V volt and the low on -V volt.

Delimiting and synchronization

To allow the receiving stations to synchronize their receive clocks on the data transmitted by the sender, a preamble is first transmitted. This preamble is a 8 Mhz block signal, with the high being mapped upon +V volt, and the low on -V volt. The length of the preamble may not exceed a maximum value (32 bit times).

The preamble is followed by a start of frame delimiter. Just as the preamble, this delimiter is generated by the physical layer, and is a violation of the PR-4 code. The start of frame delimiter is followed by the user data (data provided by the data link layer), and eventually by a postamble, which is again a PR-4 code violation generated by the physical layer.

Carrier sense

The Twentenet carrier sense function can be implemented by using a rectifier and a level detector.

Collision detection

In Twentenet, a number of checks are performed to allow the detection of a collision. The first check, performed by each transmitting station, is a comparison of the medium's signal level with the expected level. The second check, performed by all stations, is made on the length of the received preamble. If the received preamble lasts too long, it must have been generated by multiple transmitters, and thus represents a collision.

After the start of frame has been detected, the incoming signal is further checked for the absence of the 8 Mhz component (the spectrum of a PR-4 encoded Twentenet signal should show a dip at 8 Mhz). If a 8 Mhz component appears, it must have been generated by the preamble of another transmitter, and thus represents a collision. Not only the absence of the 8 Mhz component, but also the absence of the 16 Mhz component (the

beep) is checked. A beep signal will only be transmitted by a physical layer, whenever its data link layer requests it. A data link entity will request the transmission of a beep after it has been informed by its physical layer that a collision has been detected. After a beep request has been made, the data link entity will refrain from further transmission requests for a predetermined amount of time (to avoid an everlasting beep concert). A physical entity will interpret the reception of a 16 Mhz component as a collision that has been detected by another station.

Finally, each receiver checks the incoming signal on violations in the precoded PR-4 signal. If such a violation is detected, a collision is assumed to be responsible for it.

Interface signals

The information which is exchanged between a Twentenet physical and data link entity can be divided into three categories: information related to the transmission of data; information related to the reception of data; and information that ensures the correct operation of the access mechanism.

Whenever a data link entity wishes to transmit data, it first activates the 'request to send' interface signal. After detection of this signal, the underlying physical entity begins to transmit the preamble and the start of frame delimiter. When transmission of the start of frame delimiter has been completed, the physical entity signals to the data link entity 'clear to send'. This interface signal instructs the data link entity to start the actual data transmission (using the 'data' interface signals). As shown in Figure 6, 'send timing' (i.e. the clock) is the fourth interface signal needed.

For the reception of data, three interface signals are needed (see Figure 7): 'receive timing' (i.e. the clock); 'data'; and 'receive ready'. The 'receive ready' signal indicates that the received information is directed to the MAC layer ('receive ready' will not become active upon reception of a preamble).

To allow the operation and testing of the access mechanism, five interface signals are needed (see Figure 8): 'carrier sense' and 'collision detect' need no further explanation; 'beep out' is generated by the access mechanism as a reaction to the 'collision detect' — this signal tells the underlying physical entity to join the beep concert by generating a beep. These three signals are required for the normal operation of a Twentenet station.

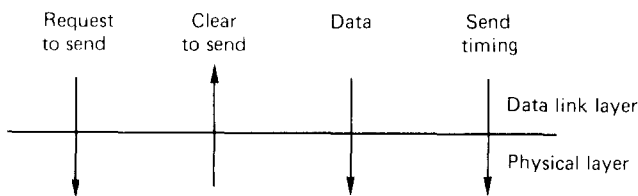


Figure 6. Interface signals used for data transmission

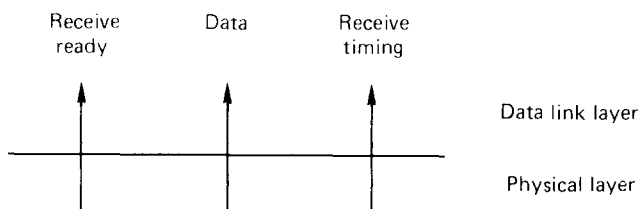


Figure 7. Interface signals used for data reception

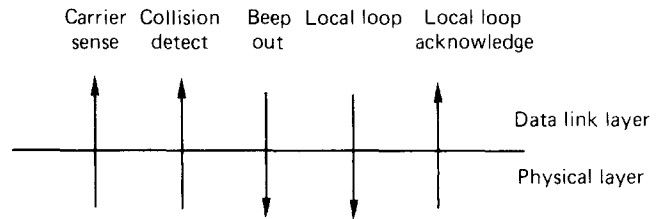


Figure 8. Interface signals used by the MAC sublayer

The two remaining signals, 'local loop' and 'local loop acknowledge', are provided for test purposes. After the physical layer receives the 'local loop' signal, a relay is switched to disconnect the station from the cable and feed back all PR-4 encoded signals from the station's output to the station's input. Since the switching of this relay may take some time (typically hundreds of ms), the physical layer will eventually respond with a 'local loop acknowledge'.

Data link layer

Access mechanism

Twentenet is based on a CSMA/CD access mechanism. In addition, it uses a priority scheme for messages to solve collisions. As opposed to the statistical back-off mechanism used in Ethernet, Twentenet uses a *deterministic* collision resolution mechanism.

Three modes of operation are defined for the stations: *Free (F-)* mode; *Priority (P-)* mode; and *Address arbitration (A-)* mode. If no messages are to be sent, the network is in the idle state, and all stations reside in the F-mode. When a station wants to transmit, it senses the bus and, if the bus is free, it starts sending. If the bus is not free, the station waits until it becomes free and then starts sending (1-persistent). If no collision occurs, the Twentenet access mechanism behaves in the same way as the Ethernet access mechanism.

If a station detects a collision, a beep signal of 32 bits will be transmitted. This beep signal serves as a synchronization signal, and ensures that all other stations become aware of the collision (it should be noted that the difference with Ethernet is that in Ethernet, receiving stations do not need to detect collisions).

When a station has completed transmission of the beep signal, it refrains from subsequent transmission attempts and ignores all further information received for a predefined period of time. This period of silence is needed so that a station does not detect the same collision again and transmit a second beep signal. On completion of this period of silence (slightly more than the round-trip delay), the station enters the priority mode.

The *Priority (P)* mode (see Figure 9) is a slotted access

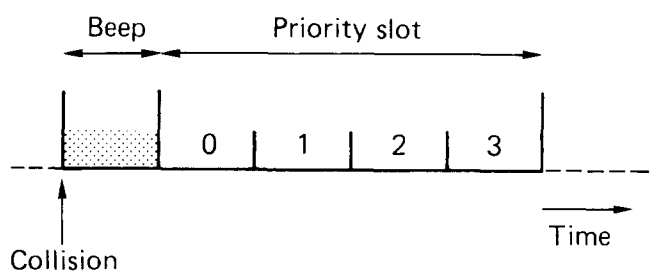


Figure 9. Twentenet priority mode

mode with a duration of four time slots (a time slot is equal to the period of time explained above). Each time slot corresponds to a possible message priority (priority 0 to 3 inclusive with 0 as the highest priority). In the first slot, priority-0 messages may be sent, in the second slot priority-1 messages may be sent, etc. If a message is sent successfully (no new collision occurs) the F-mode is re-entered.

If a collision is detected in the priority mode, the *Address Arbitration (A)* mode is entered (see Figure 10). Only the colliding stations may be active in the A-mode, with access being based on the (unique) station address. The A-mode consists of four submodes (A_0 – A_3), each of which is dedicated to arbitration on the basis of a 2-bit address portion. These address portions can be viewed as a second level of priority. A Twentenet station address is 8 bits wide, so that the A-mode necessarily ends up with a single station which is granted access.

Four time slots per A_i mode ($i = 0..3$) are needed to deal with the four possible combinations of a 2-bit address portion. Contending stations are only allowed to transmit in 'their' time slot, i.e. the time slot that corresponds with their 2-bit address portion. If only one station responds in the current time slot, the other stations will sense this transmission, and the collision is resolved. If several stations simultaneously try to occupy a time slot, the arbitration is still unresolved, and the next A_i mode is entered. This process continues until eventually one station succeeds in acquiring the bus. The F-mode is then re-entered.

Note that collisions in the F-mode occur only occasionally, dependent on network load. In addition, most collisions are solved in the P-mode. The worst case situation, going through F-mode, P-mode and A-mode in sequence to acquire the bus, is assumed to occur only rarely (a good overview of Twentenet's performance is given in Reference 6).

The mechanism explained above is, in fact, a simplified version of the Twentenet access mechanism. The complete Twentenet access mechanism behaves as explained above, except when higher priority messages are created during the collision resolution phase. In such cases, the collision resolution process is interrupted to allow the immediate transmission of these higher priority messages.

Variable throughput mechanism

One of Twentenet's initial requirements was that even low-cost stations should be able to communicate with each other. This means that those implementations that cannot offer data to their network controller at the

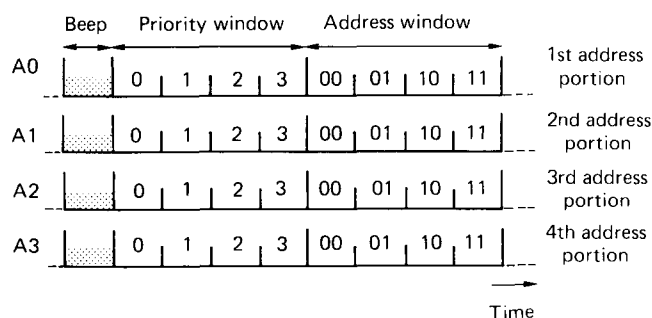


Figure 10. Twentenet address arbitration mode

maximum speed (16 Mbit/s), will be allowed to offer data at a lower speed ($16/n$ Mbit/s; $n = \{2, 4, 8, \dots, 128\}$).

The mechanism that allows for this effective throughput reduction works as follows: user data is organized in 16 bit words; attached to each word is an extra bit, and this control bit tells whether the associated word carries user data or dummy information. If the user cannot provide a new data word in time, a dummy word will be generated by the data link layer. Throughput reduction at the data link service level is thus possible, but the cost is an increase in the size of the data unit to be transmitted. Since the original data size may be up to 128 Kbytes, in the worst case situation, transmission of a single data unit may take up to eight seconds. Fortunately, a station that needs to transmit a higher priority message can interrupt such transmissions by using the 'beep out' signal.

CHANGING THE ACCESS MECHANISMS

Potential problems

The following differences, and thus potential problems, between Ethernet and Twentenet have been outlined above:

- in Ethernet, preamble generation is a data link function — in Twentenet, preamble generation is a physical layer function;
- Ethernet has an interframe gap of $9.6 \mu\text{s}$ — Twentenet does not have such a gap;
- Ethernet does not require that a receiver detects a collision in all circumstances — Twentenet does;
- if an Ethernet transmitter detects a collision, a jam signal will be transmitted — if a Twentenet station detects a collision, a beep signal will be transmitted;
- the maximum data unit size at link level for Twentenet is 128 Kbytes. The Ethernet physical layer will interrupt the transmission of a message after 20 ms (worst case);
- Twentenet has a variable throughput mechanism — Ethernet has not; and
- the Ethernet's local loop command has no acknowledgement.

Approach

To change the Ethernet access mechanism to the Twentenet access mechanism, the Ethernet physical service was enhanced towards the service required by the Twentenet access mechanism. This enhancement introduced a new layer on top of the physical layer. The service that is provided to the Twentenet access mechanism, by the combination of this new layer plus the Ethernet physical service, is now the same as the service provided by the original Twentenet physical layer. This enhancement is shown in Figure 11.

The advantage of this approach is the fact that neither the Ethernet physical service provider or the Twentenet access mechanisms are affected. This means that existing Ethernet and Twentenet components can be used without modification.

Solutions

As the functions that are needed because of the change of

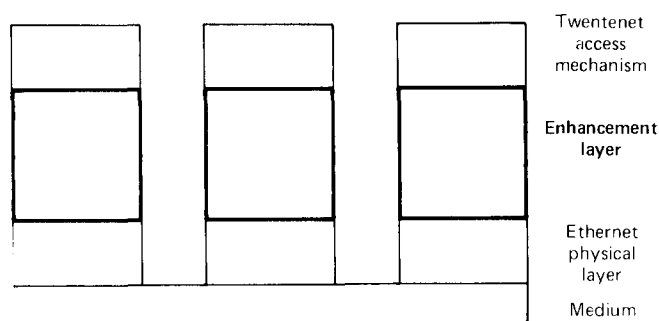


Figure 11. Enhancement layer

access mechanism are concentrated in the enhancement layer, we only have to discuss this layer.

The first function performed by this enhancement layer is the generation of the preamble and start of frame delimiter (see Figure 12). After the MAC sublayer activates the 'request to send' interface signal, the enhancement layer activates the 'data valid' signal and preamble generation starts. The preamble format is the same as in the original Ethernet design. The individual bits of the preamble are mapped on the 'data' signal. After the enhancement layer completes transmission of the start of frame delimiter (10101011), the 'clear to send' signal is sent to the MAC sublayer. As a response, this sublayer now starts to transmit the user data. This user data, offered to the enhancement layer over the 'data' interface signal, is directly mapped onto the physical layer's 'data' signal.

Once a complete frame has been transmitted, each station delays new transmission attempts until the interframe gap period of $9.6 \mu\text{s}$ is over. The enhancement layer is responsible for maintaining this period of silence. How this is accomplished depends on whether the station has just received, or has just transmitted, a frame. If a station has just received a complete frame (the PLS 'carrier indication' disappears and no PLS signal indication has occurred), a timer is started by the enhancement layer. Until this timer expires (after $9.6 \mu\text{s}$), the access mechanism is informed that a carrier is detected (by means of the 'carrier sense' signal), but no valid data is received ('receive ready' is not activated). As a result, the access mechanism postpones new transmission attempts until the 'carrier sense' signal disappears.

On the other hand, if a station has just transmitted a complete frame (the 'request to send' disappears and no PLS signal indication has occurred), the enhancement layer performs the same procedure explained above. In this case it is not possible to start the procedure after the PLS 'carrier indication' disappears because of the 'heart-beat' test performed by the physical layer.

To guarantee synchronization of the Twentenet access mechanism (each station should be in the same state), collisions must be detected by all stations. Since the

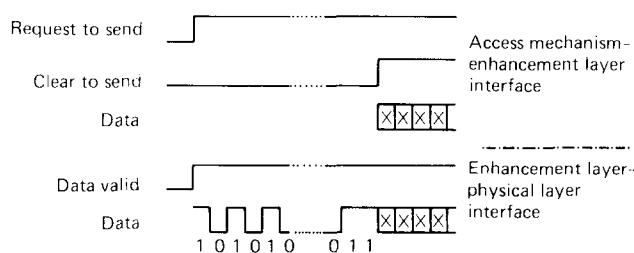


Figure 12. Time relation between interface signals

Ethernet standard does not prescribe that a physical layer implementation should in all circumstances detect a collision caused by only two transmitters, extra measures are needed. These measures are performed by all enhancement entities (and also by the receiving entities), and consist of the generation of a jam signal upon detection of a collision. As a result of this jam signal, other stations also detect the collision, and all stations are finally informed. Transmission of the jam signal lasts twice the propagation delay over the medium, and is started after the 'beep out' signal has become active.

A problem that cannot be overcome by the enhancement layer is the mismatch between Twentenet's maximum data unit size and the time window of Ethernet's jabber function. This mismatch becomes even more severe if Twentenet's throughput reduction function is enabled. The only solution possible in this case is a reduction of the data unit size at a higher level; this reduction should be such that no transmission will ever take more than 20 ms^* .

The last difference identified was the absence of the acknowledgement of Ethernet's local loop command. This did not cause a problem, though, as the enhancement layer was designed so that the 'local loop acknowledge' is automatically activated some time after the occurrence of the 'local loop' signal.

CONCLUSION

By means of an example it has been demonstrated that it is possible to isolate the access mechanism of a specific LAN, and to use this access mechanism on top of another LAN's physical service. In our example, the access mechanism expected a different type of underlying service than the one offered by the physical service provider. An enhancement layer was therefore needed to align this mismatch.

ACKNOWLEDGEMENTS

The author would like to thank Bas Olde Hampsink, Tonnie Tibben, Dave de Vries and Richard de Vries for their ideas, knowledge and implementation work that demonstrated the validity of the above ideas.

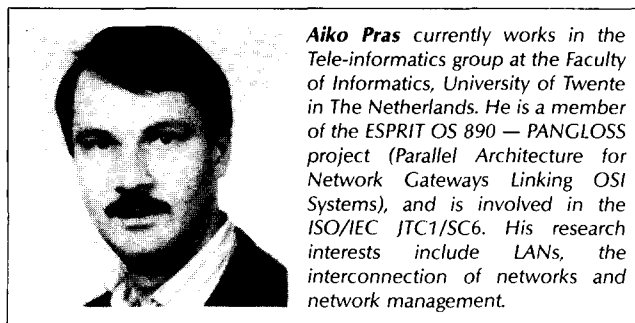
REFERENCES

- 1 *Information Processing Systems — Local Area Networks — Part 3: Carrier Sense Multiple Access with Collision Detection; Addendum 1: Medium Attachment Unit and Baseband specifications for Type 10BASE2 (ISO 8802/3/DAD1)*, ISO, Geneva, Switzerland (1987)
- 2 *Information Processing Systems — Local Area Networks — Part 3: Carrier Sense Multiple Access with Collision Detection (ISO/DIS 8802/3)* ISO, Geneva, Switzerland (1987)
- 3 *Information Processing Systems — Open Systems*

*A number of Ethernet physical layer implementations do not adhere to the standard, since they interrupt a transmission after only 5 ms.

Interconnection — Basic Reference Model (ISO 7498)
ISO, Geneva, Switzerland (1984)

- 4 **de Vries, D B** *Data Link Protocol voor het Twentenet*
Internal report INF/IPS/5, Twente University, Netherlands (1983)
- 5 **Niemegeers, I G and Vissers, C A** 'Twentenet, a LAN with message priorities, design and performance considerations' *SIGCOM 84* Montreal, Canada (1984)
- 6 **Pras, A** 'Twentenet for industrial applications' *Proc. EFOC/LAN 86* Amsterdam, Netherlands (23-27 June 1986) pp 334-338



Aiko Pras currently works in the Tele-informatics group at the Faculty of Informatics, University of Twente in The Netherlands. He is a member of the ESPRIT OS 890 — PANGLOSS project (Parallel Architecture for Network Gateways Linking OSI Systems), and is involved in the ISO/IEC JTC1/SC6. His research interests include LANs, the interconnection of networks and network management.