IPP-HURRAY! Research Group

http://www.hurray.isep.ipp.pt

Polytechnic Institute of Porto
School of Engineering (ISEP-IPP)

Ethernet Goes Real-time: a Survey on Research and Technological Developments

Mário ALVES
Eduardo TOVAR
Francisco VASQUES (FEUP)

HURRAY-TR-0001
January 2000

relatório técnico

technical report

THIS WORK IS PARTIALLY SUPPORTED BY FCT UNDER PROJECT DEAR-COTS (Praxis/P/EEI/17/1998) AND BY FLAD UNDER PROJECT SISTER (Proj.471/97)

Ethernet Goes Real-time: a Survey on Research and Technological Developments

Mário ALVES, Eduardo TOVAR

IPP-HURRAY! Research Group
Polytechnic Institute of Porto (ISEP-IPP)
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto
Portugal
Tel.: +351.22.8340529, Fax: +351.22.8321159
E-mail: {malves@dee, emt@dei}.isep.ipp.pt
http://www.hurray.isep.ipp.pt

Francisco VASQUES

University of Porto (FEUP)
Rua dos Bragas
4050-123 Porto
Portugal
Tel.: +351.22.2041774, Fax: +351.22.2074241
E-mail: vasques@fe.up.pt
http://www.fe.up.pt/~vasques

Abstract:

Ethernet is the most popular LAN technology. Its low price and robustness, resulting from its wide acceptance and deployment, has created an eagerness to expand its responsibilities to the factory-floor, where real-time requirements are to be fulfilled. However, it is difficult to build a real-time control network using Ethernet, because its MAC protocol, the 1-persistent CSMA/CD protocol with the BEB collision resolution algorithm, has unpredictable delay characteristics. Many anticipate that the recent technological advances in Ethernet such as the emerging Fast/Gigabit Ethernet, micro-segmentation and full-duplex operation using switches will also enable it to support time-critical applications. This technical report provides a comprehensive look at the unpredictability inherent to Ethernet and at recent technological advances towards real-time operation.

1. Motivation

The Xerox Palo Alto Research Centre developed Ethernet in the 1970s primarily for use as a local area network (LAN) technology for office environments. In 1979, Digital Equipment Corporation and Intel joined Xerox in partnership to promote the new network, and in 1980 published the first Ethernet specification. Ownership of the Ethernet specification was transitioned to the IEEE (Institute of Electrical and Electronic Engineers), who approved and released it as IEEE Std 802.3 [1] in 1983. In 1985 the International Standards Organisation (ISO) released the first international draft on the standard as ISO/IEC 8802-3, which established Ethernet as a true international standard. The IEC is the International Electrotechnical Commission.

Since then, Ethernet has become the most popular LAN technology. In spite of the availability of various high-speed networks like ATM (Asynchronous Transfer Mode) [2] and FDDI (Fibre Distributed Data Interface) [3], Ethernet has drawn a significant interest due to its extremely low price, maturity and stability. Despite its popularity and low-cost, Ethernet has a serious drawback concerning the support of real-time control messages. Since in an Ethernet LAN, messages may collide with each other, it is difficult to guarantee predictable delays in delivering messages to network nodes.

Currently, time critical applications such as those found in manufacturing automation are supported by industrial communication networks, such as PROFIBUS (Process Field Bus) [4], FIP (Factory Instrumentation Protocol) [5] and CAN (Controller Area Network) [6], each of these specifying the use of a deterministic Medium Access Control (MAC). However, there is a strong believe that recent technological achievements such as Fast/Gigabit Ethernet [1] and micro-segmentation with full-duplex using switches ([7], [8]), combined with the appropriate real-time scheduling and fault-tolerance techniques[1], will also enable the use of Ethernet in safety-critical applications with hard real-time constraints.

There is a recent trend from industrial communication systems providers, such as Rockwell Automation [12], to work closely with end users to determine whether their real-time industrial applications would benefit from using Ethernet. Equipment Manufacturers/Application Developers like Richard Hirschmann [25] are supporting the use of Switched Ethernet as the communication infrastructure for real-time industrial applications. Moreover, the Industrial Automation Networking Alliance – IAONA [47] - was founded in the USA with the aim of establishing Ethernet as the standard in the industrial environment and has now more than 25 members. An European-based sister organisation – IAONA Europe [48] – was very recently established (November 1999), with the objective of promoting the use of open networking in industrial and embedded applications. The Industrial Ethernet Association – IEA [49] - is another association formed in (early) 1999 to establish standards for the use of Ethernet products in the industrial marketplace.

This technical report provides a comprehensive look at the current state of Ethernet technology and its recent advances. We will also focus on available techniques to guarantee the timing requirements of Ethernet-based systems. The remainder of this technical report is organised as follows.

In Section 2, a brief overview of the Ethernet family of standards is given. In Section 3, the reasons for non-determinism and unfairness in Ethernet networks are explained. This section also includes a survey on relevant research work devoted to the use of Shared Ethernet networks in real-time applications. Section 4 describes recent Ethernet technological advances that are relevant to support real-time applications. A special relevance is given to micro-segmentation with full-duplex operation. Finally, in Section 5, some conclusions and details about future research are drawn.

2. Ethernet standards

2.1. Common misconceptions about Ethernet

Ethernet is a specification of layers 1 (physical layer) and 2 (data link layer) of the OSI Reference Model, which is equivalent to the specification made by the IEEE 802.3 standard [1]. So, it is common to use the terms Ethernet and IEEE 802.3 interchangeably.

Ethernet is a profile specification of the well known CSMA/CD (Carrier Sense Multiple Access with Collision Detection) in its 1-persistent version. Moreover, CSMA/CD, by its own, does not define the algorithm to resolve the problem of collisions. The Ethernet specification also defines the Binary Exponential Back-Off (BEB) algorithm for collision resolution and several other parameters like slot time, bit rate or maximum/minimum packet length and also the supported physical mediums like 10BaseT or 10Base5. So, Ethernet inherently contains much more detailed information than CSMA/CD.

[1] While Profibus and FIP are complete architectures (Physical, Data Link and Application Layers), Ethernet only defines Layers 1 and 2, upper layer. Thus, Ethernet should be combined with upper layer protocols.

2.2. The Ethernet standard

Within the CSMA/CD protocol, when a station wants to access the medium (transmit), it must "listen" (carrier sense) if there is network traffic before transmitting. Ethernet is a 1-persistent CSMA/CD protocol, because a host that becomes ready to transmit will transmit as soon as the channel is free, with probability 1.

If two (or more) stations decide to transmit "at the same time", that is, they have confirmed an idle shared medium and decided to transmit within a time span smaller than the signal propagation delay between the stations, a collision will occur. Every station is able to detect the collision and the ones involved in the collision start a back-off process, using the Binary Exponential Back-Off (BEB) algorithm, which is briefly described next.

Each station has a collision counter – n – that is set to 0 at start up. When the station experiences a collision, it increments n of 1. Each colliding station chooses a back-off time value that is uniformly distributed from 0 to $(2^n - 1)$ slot time[2]. If n is 1 (first collision), for example, this back-off time may be 0 or 1 slots time. For $10 \leq n \leq 15$, the maximum back-off time is constant and equal to $(2^{10} - 1)$ slots. For a 10 Mbit/s network, each station will wait for a randomly selected time period that is 0 or 102 microseconds for the first retry and ranges from 0 to 51 milliseconds on the 16^{th} consecutive retry [1].

The Ethernet standard [1] also defines several physical layer implementations, covering diverse types of cabling technologies and network topologies. All specifications work at data rates of 10 Mbit/s. Bus topologies, using coaxial cable, include the following specifications: 10Base5 (maximum of 500 metres length and 100 stations in one segment); 10Base2 (maximum of 185 metres length and 30 stations in one segment) and 10Broad36 (CATV technology with maximum of 3600 metres between stations). In 10BaseT, stations are connected to a hub via unshielded twisted pair (100 metres maximum) cables, in a star topology. The IEEE Std 802.3 also defines optic fibber implementations such as the 10BaseFL, the 10BaseFB and 10BaseFP.

2.3. The Fast Ethernet and the Gigabit Ethernet

Fast Ethernet, specified in 1995 by the IEEE Std 803.3u [1], runs at 100 Mbit/s and has two possible physical medium implementations: unshielded twisted pair and optic fibber, corresponding to the 100BaseTX/100BaseT4 and 100BaseFX specifications, respectively. An even faster Ethernet (1 Gbit/s) is specified (since 1998) in the IEEE 802.3z standard [1], commonly known as the Gigabit Ethernet, also supported by copper or optic fibber cables. Terabit Ethernet is currently under research.

2.4. Other related standards

There are a number of related standards, such as IEEE 802.3x [1] (full duplex and flow control extensions), IEEE 802.1p [11] (prioritisation scheme over Ethernet), IEEE 802.1Q [9] (configuration and auto-learning in VLANs) and IEEE 802.1D [11] (spanning trees), all deserving special attention when dealing with recent Ethernet technology. IEEE has also addressed Link Aggregation in its 802.3ad standard [10].

It is also important to make a reference to 100VGAnyLAN, which was developed by Hewlett Packard and later modified by the IEEE 802.12 committee. This 100 Mbit/s network supports a token-passing style of architecture rather than the collision strategy of Ethernet. In spite of its inherent determinism, it has not earned the same popularity as Fast Ethernet.

3. Towards real-time using Shared-Ethernet

Today, Ethernet is used primarily as an information network. However there is a strong believe that some recent technological advances will enable the use of Ethernet in dependable applications with real-time constraints.

The support of real-time industrial applications demands for the use of communication protocols with predictable timing characteristics. This is true since real-time computing systems are defined as those systems in which the correctness of the system depends not only on the logical result of computation, but also on the time at which the results are produced [13].

One of the most common arguments that has been traditionally put forward against the use of Ethernet in real-time applications is that Ethernet has a non-deterministic access delay, leading to an unpredictable timing behaviour of the communications.

The term Shared Ethernet is used when the physical communication medium is either a bus or a hub, i.e., when all generated traffic is broadcast to every station in the network, possibly causing collisions. In this section we identify the main reasons why (Shared) Ethernet has such unpredictable timing behaviour. We also survey the most relevant research works that address this problem.

[2] A slot time is 512 bit time (e.g., 51.2 μs for a 10 Mbit/s network and 5.12 μs for a 100 Mbit/s network) [22].

3.1. Non-determinism and unfairness in Ethernet

In the absence of collisions, a station that wants to transmit a packet will have a bounded access time equal to the maximum length of an Ethernet packet plus the inter-frame gap, since after the waiting period, the 1-persistent nature of the Ethernet protocol forces the station to transmit.

The *non-determinism* in Ethernet arises when a packet experiences a collision, since there is an unknown delay to re-transmit that packet due to the BEB algorithm. The non-determinism of the BEB algorithm leads to an additional phenomenon: *unfairness*. Unfairness appears because a station may capture the channel for consecutive transmissions, while others may have (eventually more critical) packets to transmit. The capture effect [16] and the packet starvation effect [15] are described next.

The Ethernet Capture Effect is the behaviour wherein under high load, one station is able to hold the channel to transmit packets consecutively, in spite of other stations contending for access. In [16], the authors give a very clear example of how this can happen. Consider two stations – station 1 (with *data1*) and station 2 (with *ack1*) – both attempting to transmit simultaneously (within a slot time of 51.2 microseconds). Each station has a collision counter – n – that is zero to start with. They experience a collision, incrementing n to 1. Each station picks a back-off time value that is uniformly distributed from 0 to $(2^n–1)$ slots, either 0 or 1, in this case. If station 2 picks a back-off value of 1 (50% probability) and station 1 picks a back-off value of 0, then station 1 successfully transmits its packet – *data1*. Station 2 waits for completion of *data1* before attempting to transmit *ack1*. The collision counter at station 2 remains at 1 while the collision counter at station 1 is reset to 0. If station 1 has another packet to transmit (*data2*), this will now contend for the channel with *ack1*. If these collide, the chosen back-off values are: 0 or 1 for station 1 and 0, 1, 2 or 3 for station 2 (since the collision counter is now 2, for this station). So, there is a higher likelihood for station 1 to succeed when resolving this collision and transmit *data2*, while *ack1* from station 2 will begin deferral when it completes its back-off interval. This would (potentially) happen up to the maximum number of allowable consecutive collisions, at which point the packet (*ack1*) would be discarded.

The packet starvation effect (PSE) is a consequence of the capture effect, i.e., some packets may starve when accessing the communication medium. In [15], some conclusions about the consequences of the PSE are drawn: the PSE causes some packets to experience latencies up to 100 times the average or to completely starve out due to 16 (consecutive) collisions; it causes some packets to experience high delays at realistic offered loads as low as 40% and causes complete starvation of some packets at offered loads as low as 60%. While the PSE has not been a problem in the past (as shown by the overwhelming success of the Ethernet), it is becoming a problem for modern traffic patterns requiring higher bandwidths and presenting real-time requirements on packet latency.

Concerning the use of Ethernet networks to support real-time traffic, both the capture effect and the packet starvation effect induce unpredictable delays and unfairness to access the network, leading to an unpredictable timing behaviour of the supported application.

A relationship between maximum transmission delay versus total traffic load in Ethernet networks is presented in [46], as result of a simulation. It is stated that while the total traffic in a segment is under 70% of the physical bandwidth, the occurrence of collisions is quite rare and thus the maximum transmission delay is negligibly small (a few milliseconds in a 10 Mbit/s network). If the total traffic exceeds 75%, the maximum delay increases radically, reaching a few hundred milliseconds (in a 10 Mbit/s network). If the traffic increases further, many transmission attempts will be frustrated due to the retransmission count exceeding 16.

Nevertheless, the widely available technology (hardware and software) to build Ethernet networks and its small price has created eagerness to its use on the factory-floor, where real-time requirements are to be fulfilled. One advantage of using Ethernet networks to support soft real-time is that the network access delay can be as small as zero, providing that that the overall network utilisation is kept small. For hard real-time applications this feature is no longer valid, since for the critical instant [51], the BEB may lead to an unpredictable timing behaviour of the supported application.

This global tendency to use Ethernet networks to support real-time applications lead the scientific community to develop techniques and methodologies to increase fairness and/or achieve determinism in Ethernet-based networks.

3.2. Solutions to enforce determinism and/or fairness in Shared Ethernet

Since the late 80's, several methods have been proposed in order to achieve a predictable timing behaviour using Ethernet networks.

The most straightforward solution to guarantee a bounded access time is to use a Time Division Multiple Access (TDMA) strategy, where each station has pre-allocated time intervals to transmit its packets. Therefore, a collision-free Ethernet with a predictable timing behaviour is achieved. Its main disadvantage is the inherent non-flexibility, since even if a station has nothing to transmit, it will "use" its share of time, resulting in a non-used time period. TDMA has been used as a basis for some real-time communication protocols, namely by the MARS Bus Protocol [45].

Two other proposals provide a collision-free environment through the use of a (virtual) token-passing mechanism - RETHER [14] and TEMPRA [28]. In RETHER, the network is in (normal) CSMA/CD mode when there is no need for real-time traffic (non real-time – NRT). If a node needs to send a real-time message (RT), it requests a change to the

real-time mode and waits for an acknowledgement from all other nodes. Inefficiency arises when these receiving nodes still have messages in the back-off phase. TEMPRA [28] also has two modes of operation (RT and NRT). The real-time mode is based in a timed packet release access mechanism, which controls the packet release through the use of a slot pulse (common time reference) sent by a monitor node. This method is quite unfair, since the network access delay is function of the distance to the monitor node (nearer implies higher priority). Moreover, as it demands special hardware and software platforms, it is not Ethernet compatible.

While TDMA methods pre-allocate the channel in a station-oriented way, PCSMA (Predictable CSMA) [38] demands an off-line traffic scheduling, considering that all real-time messages are periodic (message-oriented scheduling). Every station will have a global knowledge about all the generated messages (period, length, etc.). While this approach can be collision-free (if no NRT traffic is assumed), it has an overhead (in every message) inherent to the off-line scheduling.

While these methods provide a predictable timing behaviour based on collision avoidance techniques, some other methods focus on the modification of the Ethernet collision resolution algorithm to achieve the same target.

Maybe the most well known method based on such algorithm modification is the CSMA/DCR [27]. In CSMA/DCR (CSMA with Deterministic Collision Resolution), the probabilistic BEB algorithm is replaced by a deterministic binary tree search algorithm. In the absence of collisions, the channel is randomly accessed as in Ethernet. When a collision occurs, a binary tree search is executed, scheduling messages with a station address-based policy. As this address-based priority may turn out to be inadequate for real-time systems, a deadline-oriented version has also been proposed: DOD-CSMA/CD (Deadline Oriented Deterministic CSMA/CD) [27]. This algorithm is similar to CSMA/DCR, the major difference being the use of deadline classes, implementing the EDF (earliest deadline first) scheduling policy.

CSMA/PDCR (CSMA with Priority Deterministic Collisions Resolution) [40] merges the ideas proposed in P-CSMA and CSMA/DCR. The major difference is that, unlike P-CSMA, CSMA/PDCR only comes to action after a collision occurs. Then, it is resolved using a deterministic tree algorithm, possibly similar to DCR (it is not described in the paper).

Another class of methods was developed in the framework of Ethernet-like protocols, intended to deal with the capture/packet starvation effect, and thus with the objective to enforce fairness in the medium access. This class of methods is only suitable to support soft real-time applications since it does not guarantee a bounded network access delay.

Probably the simplest methods to improve fairness in the medium access are CABEB [16] and BLAM [29]. Both are based in a different (when compared to BEB) manipulation of the collision counter, in order to avoid that a station can capture the channel for an undetermined period of time. The major differences between these two methods is that while CABEB uses a distinct collision counter for each station, BLAM uses a global collision counter and also defines a channel holding time limit to every station.

P-CSMA (Prioritised CSMA) [39] is a TDMA-like method based on message priorities. Following a channel utilisation, the channel time is divided in *n* (reservation) slots, where *n* is the number of priority levels. The first slot is the highest priority one. A station may only transmit its highest priority message in the corresponding slot. This scheme guarantees that no collisions between messages of different priorities will occur, thus improving the fairness of the medium access. Concerning the collision between messages of the same priority, the author only mentions that they may be resolved using a random delay (the BEB may be considered).

FDDQ [15] is a tree-based collision resolution mechanism that provides two first-come, first-served (FCFS) access priorities to the network by maintaining two globally distributed queues of waiting senders across the controllers, one for high priority (real-time) traffic and the other for low priority (non real-time) traffic. The main objective of this algorithm is to eliminate the unfairness due to the packet starvation effect.

Finally, it is worthwhile to mention three classes of methods that constrain the generated traffic in a fair way, without making changes to the Ethernet protocol - Virtual Time Protocols [33] [37], Window Protocols [30] [31] [34] [41] and Traffic Smoothing [23] [24]. These methods execute a local scheduling function of a global knowledge of the network state, increasing the network access fairness and reducing the collisions number based on some priority criterion.

Virtual Time Protocols implement a packet release delay (PRD) mechanism, which is function of some relevant parameter, usually laxity [33] or (other) priority [37]. Message release is delayed by a time proportional to the message laxity, for instance. Window protocols are based on the following principle [32][43]. If a station has a message within some time/priority window, it is allowed to transmit. The window size is dynamically changed depending on the channel state: idle, busy or collision. The most basic approach arbitrates access to the channel on a first-come-first-served basis [42]. Other window protocols followed, implementing laxity/deadline [30] [34] or (other) priority-based [31][41] policies. Finally, traffic smoothing statistically bounds the medium access time by limiting the packet arrival rate at the MAC layer [23] (smoothing non real-time traffic bursts).

4. Towards a real-time Ethernet: a survey on technological developments

4.1. Moving to faster Ethernet networks

Since the original 10 Mbit/s, Ethernet's data rate has been increased by two orders of magnitude, firstly in 1995 (Fast Ethernet) to 100 Mbit/s and later in 1998 (Gigabit Ethernet) to 1 Gbit/s [1]. Fast Ethernet is basically Ethernet running at 100 Mbits/s, with the same frame structure, addressing scheme, and CSMA/CD method. However, all network-timing parameters are scaled by a factor of 10 when configuring a Fast Ethernet network. This tends to reduce the maximum segment lengths in some configurations when compared to a 10 Mbit/s network [26].

Most applications will not enjoy a substantial performance increase due to the increased data rate alone. In particular, a plant-floor network of small microprocessor-based I/O racks, sensors, actuators, drives and other device interfaces are likely to consume and produce small amounts of data encapsulated in 64 byte Ethernet frames (the smallest frame size supported by Ethernet). The performance of these devices is more likely to be limited by the speed of their microprocessor and embedded firmware than the network communication speed. It is unlikely that a network of such devices would fully utilise the 100 Mbit/s Ethernet bandwidth, unless an efficient application-layer protocol was used.

Nevertheless, if the tendency to have heterogeneous traffic in industrial communication systems is confirmed, bandwidth-hungry applications such as voice or video will surely alter this scenario.

One area of performance wherein 100 Mbit/s Ethernet shows noticeable improvement over 10 Mbit/s Ethernet is in the area of collision recovery. The back-off times for a 100 Mbit/s Ethernet are 1/10th of those for 10 Mbit/s Ethernet. On a loaded network where collisions are an issue, 100 Mbit/s Ethernet may show noticeably better performance than 10 Mbit/s Ethernet. As mentioned earlier [15][16][46], a hypothetical band of 60-70% of total bandwidth may be drawn to threshold negligible delays from considerable delays. Thus, speed can make the difference. However, if loading and collisions are not already an issue on a 10 Mbit/s Ethernet network, simply upgrading to 100 Mbit/s Ethernet may not show improvement to justify the investment.

More importantly, note that the data rate increase, by its own, does not allow overcoming the problem of non-determinism in the network access delay. The foreseen solution to the non-determinism problem is either to solve collisions deterministically or to completely eliminate them. The Switched-Ethernet technology permits to completely eliminate collisions, as will be explained in the next sub-section.

4.2. Micro-segmentation with full-duplex operation

An important development concerning Ethernet technologies is the **switch** technology. Switches break up collision domains into single devices or small groups of devices, effectively reducing or even eliminating the number of collisions.

Switches provide a flexible and scalable solution to the problems and limitations inherent to Shared Ethernet networks (bus or hub-based), through the use of new mechanisms such as micro-segmentation and full-duplex operation. These mechanisms, which will be briefly described next, may improve the timing determinism and performance of Ethernet to a great extent.

Ethernet networks have been based on hubs (e.g., the 10BaseT specification) or buses (e.g., the 10Base5 specification), where network nodes shared the same physical medium (broadcast network). This means that only one node could send data at a time. If the number of nodes increased (potentially increasing the network load), a bridge or a router could be used to segment the traffic, splitting the network into different collision domains.

A switch is an intelligent hub that can read and process the destination address of the incoming data and send it only to the required ports [9]. While in a hub/repeater, data sent by one node will be broadcast to all other nodes, in a switch, data is only sent to the destination node(s), which means that several nodes can transmit at the same time. A bridge also overcomes such problem (common traffic between segments), but lacks some of the technological features inherent to switches (VLANs, flow control, etc.). Moreover, a switch has several ports, while a bridge usually has only two. Comparing to routers, switches make their forwarding decisions based on link-layer addresses, while routers must parse network layer (Layer 3) header information, and make changes in the packet, before forwarding it to the destination. Thus, a switch avoids the processing overhead inherent to a router.

If segmentation in a switch is taken to an extreme, each device is isolated in its own segment and has the entire port throughput for its own use. Micro-segmentation removes one of the primary causes of LAN congestion. This means that there are no more problems with the aggregation of traffic from multiple stations, since there can be only one station offering load to the segment at any given time. In these conditions, the congestion burden is shifted to the central switch. This may not be a problem, as the switch can be built to handle the total aggregate load of all the attached devices (non-blocking feature). A possibility of congestion still remains due to traffic patterns causing load to converge on a given port. If many devices are all attempting to communicate with a single device, then it is possible to have a

congestion problem, even though every device has its own LAN (flow control techniques must be used to overcome this problem).
Ethernet is normally a half-duplex communication system. While data can be transferred in both directions, a station is either transmitting or receiving, at a given time. On the original physical media used with Ethernet (i.e., coaxial cable), this was the only way to communicate, since the same wire (and frequency) was used for transmission and reception.
10BaseT (twisted pair) technology has separate pairs for transmission and reception. However, since many devices are typically sharing the medium, there is the need for a mean to prevent multiple simultaneous transmissions. 10BaseT uses the presence of received information during a transmission to indicate a collision to the transmitting stations, invoking the back-off and retransmission algorithm necessary for proper Ethernet operation. As a consequence, full duplex cannot be used in a hub-based network.
A micro-segmented topology (where at most one device is connected to each port), assures that there is only one device wishing to use each wire pair. Only the attached station ever speaks to the switch (using the transmit pair of the cable), and only the switch ever speaks to the attached station (using the receive pair of the cable) – full-duplex operation. There is never contention between stations for the use of the medium, oppositely to a Shared Ethernet topology.
With the possibility of contention removed, there is no longer any real need for the collision detection and back-off function. They can be eliminated and both the station and the switch are allowed to transmit at will, in both directions simultaneously – full-duplex operation. Theoretically, this mode of operation doubles the data transfer rate.

4.3. Advances in switching

Another very important difference between a switch and a bridge is that the former is able to behave somehow like a massively parallel LAN bridge [7], allowing information on any port to pass to any other port, simultaneously.
LAN switches may have two basic switching modes – cut-through and store-and-forward [17]. Cut-through provides *on-the-fly commutation*, i.e., a packet is forwarded (by the switch) as fast as possible. The switch inspects each incoming frame for the destination address of the target. It quickly determines the appropriate output port by consulting its internal address map (similarly to a bridge). If the output port is available, the switch immediately forwards the frame to the destination, reducing the latency inherent to most bridge architectures (that have to receive the entire frame before making a forwarding decision). Nevertheless, it is important to note that a switch still has a longer latency time than a hub/repeater. A store-and-forward switch, on the other hand, accepts and analyses the entire packet before forwarding it to the destination. It takes more time to examine the packet, but it allows the switch to catch certain packet errors and keep them from propagating bad packets through the network.
Today, the speed of store-and-forward switches has caught up with cut-through switches to the point where the difference between the two is minimal [17]. Also, there are a large number of hybrid switches available that mixes both architectures, at choice. A compromise must be made by the communication system designer/manager between the two modes of operation. While cut-through mode provides smaller latency, thus suitable for communication systems with real-time constraints, store-and-forward may be preferred when dealing with fault tolerance constraints.
Intel [50] considers a third switching mode – fragment-free – that filters out most error packets but does not necessarily prevent the propagation of errors throughout the network. It also considers an adaptive switching technology that chooses the optimal forwarding mode (for each port independently) based on real-time error monitoring.
Another important issue in switching is the blocking/non-blocking feature. Considering a switch specification and adding up all the ports at the theoretical maximum speed, the result will be the theoretical sum of the switch throughput. If the switching bus, or switching components cannot handle the theoretical total bandwidth, the switch is considered to be a *blocking switch*, otherwise it is a *non-blocking switch* [17].
For most applications, a *blocking switch* that has an acceptable and reasonable throughput level will work just fine [17]. Consider an eight port 10/100 switch. Since each port can theoretically handle 200 Mbit/s (full duplex), there is a theoretical need for 1.6 Gbit/s. Though, considering the simultaneity coefficient, each port will not probably exceed 50% utilisation, so a 800 Mbit/s switching bus might be adequate.
Of course that when designing a real-time communication system over Switched Ethernet, the blocking/non-blocking characteristic, and how low/high should be the data-rate (in a blocking solution), becomes an important issue that should be object of a thorough analysis.

4.4. Traffic prioritisation

One of the latest Ethernet breakthroughs is intended to allow for different traffic classes, using priorities. IEEE 802.1p (ratified in September 1998) [11] is a specification for giving Layer 2 switches the ability to prioritise traffic and perform dynamic multicast filtering. Only the former will be addressed.
The prioritisation specification works at the MAC sub-layer of the OSI model. To be compliant with 802.1p, Layer 2 switches must be capable of grouping incoming LAN packets into separate traffic classes. The standard defines eight priority levels – 0 to 7 – used according to traffic criticality, being seven the highest priority. Nevertheless, many switch vendors only support up to four traffic queues.

The 802.1p standard demands the use of priority fields within the packet to signal the switch of the priority-handling requirements. Such priority fields were defined by the IEEE's 802.1Q working group, which is focusing on virtual LAN identification. The IEEE 802.1Q specification defines a tag header consisting of 32 bits, all inserted after the packet header's normal destination and source addresses. Of these, three priority bits are used for signalling 802.1p switches. Switches, routers, servers, even desktop systems can set these priority bits.
In order to deal with traffic priorities, switches have to be capable of employing multiple buffer queues for each output port. In a conventional single-buffer switch, when congestion occurs, all packets must wait their turn to move on. By contrast, a switch with multiple-queue hardware can give higher priority packets fast handling inside the switch; some will actually overtake lower priority packets in the few milliseconds required to move through the switch. The 802.1p specification anticipates multiple-queue hardware by recommending how the eight traffic classes should be assigned in systems with two, three, four or more queues per port.
Interesting considerations are made in [35], where the authors defend that prioritisation is meaningless unless the congestion level overpasses 100% on a given output port. For example, if a switch is supporting two streams at 80% port capacity, prioritisation services do not kick in because there is no contention for bandwidth. Nevertheless, setting priority bits and implementing class-based forwarding may turn out to be significant steps towards real-time priority-based communication systems. In fact, in a (hard) real-time communication system, bounded access times must be guaranteed, independently of the network load.

4.5. Flow control

Flow control is a mechanism for limiting traffic load to a certain level. While remembering traffic smoothing [23] [24], flow control may be crucial to avoid the degradation of real-time communication networks performance.
Flow control is an important switch feature that eliminates dropped packets on congested full duplex ports (on half-duplex, congestion is avoided through carrier-sense jamming). Flow control achieves this objective by "warning" stations that are overloading the switch.
Though, some defend that flow control may degrade network performance, in some situations [36]. Flow control can ensure better throughput by throttling devices that send traffic when the switch is overloaded, thereby reducing potential frame loss when a switch gets congested. On the other side, flow control can add latency (and traffic, because of the so-called "pause frames") to a potentially congested segment or, even worse, to an uncongested one. This is an important drawback for real-time communications. The flow control mechanism is briefly described next.
If network traffic becomes so high that a switch can no longer process all the data, one of two things may happen. The switch may drop data (not retransmit all of it). Network software protocols will need to recover lost data and the network will be significantly slower. Alternatively, a switch may be able to employ methods to ensure that data is not lost, by controlling the flow of the traffic. IEEE Std 802.3x Flow Control [1] allows a switch to do this.
Flow control can be implemented on a link-by-link or end-to-end basis and allows all devices on the path to reduce the amount of data they receive [19]. Because flow control has implications beyond full duplex and the MAC sub-layer of the data link layer, methods and standards are still under consideration by the IEEE 802.3u committee.
Link-to-link flow control examines only an individual link between switches or stations. When the recipient of a transmission becomes busy, it will send a signal to the directly linked transmitter. If that transmitter is not the originator of the traffic, this signal would have to be propagated back through each link in order to reach the originator of the traffic. A more complete solution would be to identify the specific traffic causing congestion, but this may require upper-level protocols.
End-to-end flow control means that the switches at each end of the link communicate to throttle back the end stations that are originating the traffic. Until this information is propagated, packets must be stored or dropped, meaning that flow control reduces, but does not eliminate, the need for buffers.
Another related problem is the one of switch buffer limitations. As packets are processed in a switch, they are held in buffers [17]. If the destination address is congested, the switch holds on to the packet as it waits for bandwidth to become available on the loaded segment. Buffers that are full present a problem. Therefore, some analysis of the buffer sizes and strategies for handling overflows are needed for the network design. In real world networks, loaded segments cause many problems, so their impact on switch consideration is not important for most users, since networks should be designed to eliminate congested segments.
There are two strategies for handling full buffers. One is back-pressure flow control, which sends packets back upstream to the source nodes of packets that find a full buffer. This compares to the alternative of simply dropping the packet, and relying on the fault-tolerance features in networks to retransmit automatically. One solution spreads the problem in one segment to other segments, propagating the problem. The other solution causes retransmissions, thus producing a non-optimal load.
Neither strategy solves the problem, so switch vendors use large buffers and advise network managers to design switched network topologies to eliminate the source of the problem – congested segments.

4.6. Summary

The following table summarises the major technological achievements that may lead to the use of Switched Ethernet as a real-time communication infrastructure:

Input	Short Description	Pros For Real-Time
Fast and Gigabit	Fast Ethernet (100 Mbit/s) and Gigabit Ethernet (1 Gbit/s) are already available	Communication time is divided by 10, 100 ⇒ smaller load⇒ smaller access time.
Traffic segmentation	Traffic destined to a specific station only flows through the correspondent switch port.	Smaller load ⇒ less collisions ⇒ smaller access time
Micro-segmentation	Each station has a dedicated switch port	No contention due to other stations' traffic) (contention limited to trying to transmit while receiving) ⇒ smaller access time.
Full duplex (with micro-segmentation)	Simultaneous communication between switch and station, in both directions (only works in micro-segmented network.).	Theoretically, can double the bit rate. Contention and collisions removed implying no need for CSMA/CD or BEB, respectively.
Traffic prioritisation	Switches group incoming packets into separate traffic classes (eight priority levels – 0 to 7).	Switching performed according to traffic criticality.
Flow control	Flow control eliminates dropped packets on congested ports by "warning" stations that are overloading the switch.	Allows stations to be aware of switch congestion and to "smooth" traffic accordingly.

5. Concluding Remarks

5.1. Concerning this work

The consolidated use of Ethernet as a lower layer communication protocol in Local Area Networks has been turning this infrastructure increasingly cheaper, faster and more dependable. Although Ethernet is known because of its non-deterministic timing behaviour, the presented technological improvements makes it very appealing for real-time applications.

This technical report presented the state of the art concerning the use of Ethernet in real-time applications. A preliminary analysis on the shortcomings of the Ethernet protocol, namely on its unpredictable timing behaviour is undertaken, describing the capture and packet starvation effects. Some techniques to guarantee the timing requirements of Shared-Ethernet systems were object of a general overview. Focus was then put in recent technological advances that may lead to a inherent deterministic behaviour, namely micro-segmentation and full-duplex operation.

5.2. Concerning future work

Future work will include an analysis of access time guarantees in switched Ethernet networks and the application of Switched Ethernet as a communication infrastructure for real-time, fault-tolerant, self-evolving systems [18].

For this purpose, some additional switch's characteristics must be thoroughly analysed, since they are relevant to the design and implementation of the referred kind of computer systems. For instance, several fault-tolerance mechanisms are available at Layer 2 through spanning trees and port trunking. The IEEE Std 802.1D [11] (Spanning Tree Protocol) can be used to provide redundant network paths [21], still protecting against network loops. The spanning tree algorithm allows a network manager to design in redundant links, with switches attached in loops [17]. Port Trunking establishes backbone links by treating multiple parallel links as a single network pipe [21]. It also provides link redundancy, i.e., traffic on any failed link comprising a network trunk, automatically switches over to the other links in the trunk.

Envisaging a communication network infrastructure for self-evolving computer systems [18], switches now include several technological characteristics that enable an inherent dynamic behaviour, namely dynamic switching, auto-negotiation, automatic load balancing and Virtual LANs.

Switches also are very appealing for dealing with distributed systems where it is necessary to support new nodes in the network, nodes giving up of the access, or just mobile nodes (like in [18]). A switch can dynamically determine the mapping of link addresses to its ports [7]. This characteristic is important because: devices may be moved from port to port, interface hardware may change, thus changing the globally unique address along with it and topology changes in the network may make devices appear to move relative to the switch ports.

The Fast Ethernet specification [1] describes a negotiation process that allows devices at each end of a network link to automatically exchange information about their capabilities and perform the necessary configuration to operate together

at their maximum common level [19] and to define the duplex mode [20] – this feature is known as auto-negotiation. For example, auto-negotiation determines whether a 100 Mbit/s hub or switch is connected to a 10 Mbit/s or 100 Mbit/s adapter and then adjust its mode of operation accordingly.
Virtual LANs (VLANs) are defined in IEEE 802.1Q [9] (configuration and auto-learning in VLANs). The added value is enormous when dealing with emerging flexibility and mobility requirements. Before switching, a workgroup was defined by those devices plugged into a given hub, to a set of cable segments connected by repeaters or to a segment of cable (when using bridges). Using VLAN technology, we define a workgroup not by where the device is plugged in, but by the multicast filters defined in the switch, associated with the applications running on that device. If a device moves within the switched network, multicast filters change automatically, maintaining logical connectivity without having to physically intervene. Currently, a network administrator must define these multicast filters, under manual control. Work is ongoing in the industry to develop standards for automatic registration (and authentication) of multicast filters in a switch, by the attached stations. This will further enhance the power of VLAN technology.
A commercially-originated solution named Automatic Load Balancing (developed by 3COM) also deserves to be mentioned. It pursues two objectives: to minimize traffic traveling between segments and to reduce load on the segment switch, in order to confine the heaviest data exchanges to end stations on the same segment. Switches feature built-in algorithms to perform the necessary calculations for obtaining the best possible distribution of traffic. Automatic Load Balancing acts on these calculations and moves end stations between segments accordingly.
A thorough analysis of traffic prioritisation (IEEE 802.1p) will also be necessary, manly focused on how this scheme is implemented in practise, in switches from several vendors.
Very little research concerning the analysis of Switched Ethernet as a real-time communication infrastructure has been undertaken until this moment. Nevertheless, two recently published papers focus on timing guarantees [8] and load analysis [44] for Switched Ethernet networks.
In [8] the authors show how it is possible to achieve a completely deterministic Ethernet system using switches. For this purpose, several simplifications are assumed. An automation system is composed of a *control node* and of N *field devices* and the systems runs in a cyclic way. A cycle begins with each of the field devices sending (sensor) data to the control node, the latter calculates new actuator data and sends it to each of the field devices. The paper shows how to determine the minimum cycle time, depending the number of nodes, the number of input/output bits and the topology of the switched network. Data processing time is assumed to be zero. The main contribution of this paper concerns the minimum cycle time comparison between unicast and multicast transmitting methods and between network topologies (line, ring and tree). The simplifications made somehow limit the applicability of these results, since present and future automation systems are not restricted to cyclic operation and one should count on non-zero data processing times.
An important problem in Switched Ethernet network design is to compute the amount of traffic or load on network internal nodes (switches), given the volume and pattern of traffic among network external or end nodes (stations). This is often referred to as load analysis. A systematic solution to load analysis is presented in [44] that uses the graph model of the network together with a compact representation of the network traffic in the form of a traffic matrix. The offered loads to network switch devices can be used to have a rough estimation of the delay times in the network. The authors state that this overly simplified method may be the only possible approach in many practical design problems, as the internal operation and structure of switch devices is often not provided by the vendors.
In conclusion, a deep study on the (real) operation of Ethernet switches, with all the inherent traffic-related features and a subsequent timing analysis will be necessary in order to achieve a communication infrastructure for real-time, fault-tolerant, self-evolving computer systems.

References

[1] IEEE Standard 802.3, 1998 Edition, *Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements--Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*, This edition includes all contents of the 8802-3:1996 Edition, plus IEEE Std 802.3aa-1998, IEEE Std 802.3r-1996, IEEE Std 802.3u-1995, IEEE Std 802.3x&y-1997, and IEEE802.3z-1998.

[2] ATM User Network Interface, Version 4.0, ATM Forum, 1996.

[3] ISO, *Information Processing Systems – Fibre Distributed Data Interface (FDDI) – Part 2: Token Ring Media Access Control (MAC)*, ISO International Standard 9314-2, 1989.

[4] *Profibus Standard DIN 19245 Part I and II*, Translated from German, Nutzer Organisation e.v., 1992.

[5] *FIP Bus for Exchange of Information between Transmitters, Actuators and Process Controllers*, French Standard NF C46, 1989.

[6] *Road Vehicle – Interchange of Digital Information – Controller Area Network (CAN) for High-Speed Communication*, ISO 11898, ISO, 1993.

[7] R. Seifert, *Issues in LAN Switching and Migration from a Shared LAN Environment*, Technical Report, Networks and Communications Consulting, November 1995.

[8] S. Rüping, E. Vonnhame, J. Jasperneite, *Analysis of Switched Ethernet Networks with Different Topologies Used in Automation Systems*, in Proceedings of Fieldbus Conference (FeT'99), Magdeburg, Germany, pp. 351-358, Springer-Verlag, September 1999.

[9] IEEE 802.1Q, *1998 IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridge Local Area Networks*.

[10] IEEE 802.3ad, D2.0 Supplement to IEEE Std 802.3, 1998 Edition: Link Aggregation (unapproved standard).

[11] (ISO/IEC) ANSI/IEEE Std 802.1D, 1998 Edition, *Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks - Common Specification - Media access control (MAC) bridges*. This is a revision of ISO/IEC 10038: 1993, 802.1j-1992 and 802.6k-1992. It incorporates IEEE Std 802.11c-1998, P802.1p and P802.12e.

[12] Rockwell Int. Corp., *Ethernet for Industrial Control*, An Ethernet White Paper, Rockwell International Corporation, April 21, 1998.

[13] J. Stankovic, *Real-Time Computing Systems: the Next Generation*, in Tutorial: Hard Real-Time Systems, J. Stankovic and K. Ramamritham (Editors), IEEE Computer Society Press, Los Alamitos, USA, pp. 14-38, 1988.

[14] C. Venkatramani, T. Chiueh, *Supporting real-time traffic on Ethernet*, in Proceedings of the IEEE Real-Time Systems Symposium (RTSS'94), pp. 282-286, December 1994.

[15] B. Whetten, S. Steinberg, D. Ferrari, *The packet starvation effect in CSMA/CD LANs and a solution*, in Proceedings of the 19th Conference on Local Computer Networks, pp. 206-217, 1994.

[16] K. K. Ramakrishnan, H. Yang, *The Ethernet Capture Effect: Analysis and Solution*, in Proceedings of the 19th Local Computer Networks Conference, pp.228-240, 1994.

[17] Lantronix, *Ethernet Tutorial: Network Switching*, Technology Tutorials, http://www.lantronix.com/technology/tutorials/switching.html, 1999.

[18] Gerhard Fohler, Sasi Punnekkat, *Self-Evolving Dependable Real-Time Systems*, in Proc. of the 10th European Workshop on Dependable Computing – EWDC-10), Vienna, Austria, May 6-7, 1999.

[19] Cisco Systems, Inc., *Scalable Fast Ethernet Solutions for Migrating to High Performance Networks*, White Paper, 1996.

[20] Cisco Systems, Inc., *10/100 Mbps Auto-Negotiation*, Application Note, 1996.

[21] 3COM, *Superstack II Features*, http://www.3com.com/products/dsheets/400260a.html, 1999.

[22] D. R. Boggs, J. C. Mogul, C. A. Kent, *Measured Capacity of an Ethernet: Myths and Reality*, WRL Research Report 88/4, Digital, and also published at ACM SIGCOMM, Stanford, California, USA, August 1988.

[23] Seok-Kyu Kweon, Kang G. Shin, Qin Zheng, *Statistical Real-Time Communication over Ethernet/Fast Ethernet*, Technical Report, Mitsubishi Electric Research Lab., Cambridge, MA, August 1997.

[24] Seok-Kyu Kweon, Kang G. Shin, Qin Zheng, *Statistical Real-Time Communication over Ethernet for Manufacturing Automation Systems*, in Proc. IEEE Real-Time Technology and Applications Symposium (RTAS'1999), pp. 192-202, June 1999.

[25] Hirschmann GmbH, *Hilights: Automation and Networking Solutions*, Edition 10/99, Germany.

[26] Gigabit Ethernet Alliance, *Gigabit Ethernet Comparison Summary: topology rules for maximum network distance*, http://www.gigabit-ethernet.org/technology/overview/compsum.html, 1999.

[27] G. Lann, N. Riviere, *Real-time communications over broadcast networks: the CSMA/DCR and the DOD-CSMA/CD protocols*, Technical Report 1863, INRIA, Mars 1993.

[28] D. W. Pritty, J. R. Malone, D. N. Smeed, S. K. Banerjee, N. L. Lawrie, *A Real-Time Upgrade for Ethernet Based Factory Networking*, in Proc. of IECON, pp. 1631-1637, 1995.

[29] M. L. Molle, *A New Binary Logarithmic Arbitration Method for Ethernet*, Technical Report CSRI-298, Computers Research Institute, University of Toronto, Canada, April 1994 (revised July 1994).

[30] Wei Zhao, John Stankovic, *A window protocol for transmission of time-constrained messages*, Proc. of IEEE Transactions on Computers, Vol. 39, nº9, pp 1186-1203, September 1990.

[31] M. Li, *A priority-based protocol for the 802.3 network*, in Proc. of IFAC Distributed Computer Control Systems, Toledo, Spain, pp. 19-22, 1994.

[32] N. Malcolm, W. Zhao, *Hard Real-Time Communication in Multiple Access Networks*, Journal of Real-Time Systems, vol. 8, pp. 35-37, Kluwer Academic Publishers, 1995.

[33] W. Zhao, K. Ramamritham, *A virtual time CSMA/CD protocol for hard real-time communication*, Proceedings of the IEEE Real-Time Systems Symposium, pp. 120-127, 1986.

[34] J. F. Kurose, M. Schwartz, T. Yemini, *Controlling window protocols for time-constrained communication in a multiple access environment*, in Proc. of the 8th IEEE Int. Data Communication Symp., 1983.

[35] C. Bruno, K. Tolly, *Minding your QoS p's and Q's*, Network World, March the 19th 1999.

[36] Network World Fusion, *Vendors on flow control*, http://www.nwfusion.com/netresources/0913flow2.html, September the 13th 1999.

[37] M. El-Derini, M. El-Sakka, *A CSMA Protocol under a Priority Time Constrained for Real-Time Communication*, in Proc. of Conf. on Future Trends on Distributed Computing Systems, pp. 128-134, Cairo, Egypt, 1990.
[38] R. Yavatkar, P. Pai, R. Finkel, *A Reservation-based CSMA Protocol for Integrated Manufacturing Networks*, Technical Report 216-92, Department of Computer Science, University of Kentucky, October 1992.
[39] F. Tobagi, *Carrier Sense Multiple Access with Message-Based Priority Functions*, IEEE Transactions on Communications, Vol. Com-30, nº1, January 1982.
[40] J. Turiel, J. Marinero, J. González, CSMA/PDCR: A Random Access Protocol Without Priority Inversion, Annual Conference of the IEEE Industrial Electronics Society 1996 (IECON'96), pp. 910-915, 1996.
[41] R. Jan, Y. Yeh, CSMA/CD protocol for time-constrained communication on bus networks, in IEE Proceedings-I, vol. 140, nº3, June 1993.
[42] R. Gallager, *Conflict resolution in random access broadcast networks*, in Proc. of- the AFOSR Workshop in Communication Theory and Applications, pp. 74-76, Provincetown, Massachusetts, USA, September 17-20, 1978.
[43] J. F. Kurose, M. Schwartz, T. Yemini, *Multiple-Access Protocols and Time-Constrained Communication*, Computing Surveys, Vol.16, nº1, March 1984.
[44] P. Torab, E. Kamen, *Load Analysis of Packet Switched Networks in Control Systems*, in 25th Annual Conference of the IEEE Industrial Electronics Society – IECON'99, pp. 1222-1227, San Jose, California, USA, November 29 – December 3, 1999.
[45] H. Kopetz, A. Damm, Ch. Koza, M. Mulazzani, W. Schwabl, Ch. Senft, R. Zailinger, Distributed Fault-Tolerant Real-Time Systems: The MARS Approach, IEEE Micro, 9(1):25-40, February 1989.
[46] M. Iwasaki, T. Takeuchi, M. Nakahara, T. Nakano, *Isochronous Scheduling and its Application to Traffic Control*, IEEE Real-Time Systems Symposium, pp. 14-25, Madrid, Spain, December 2-4 1999.
[47] IAONA – Industrial Automation Networking Alliance, USA, http://www.iaona.com
[48] IAONA Europe – Industrial Automation Networking Alliance, Europe, http://www.iaona-eu.com
[49] IEA – Industrial Ethernet Association, http://www.industrialethernet.com
[50] Intel Adaptive Technology – Optimising Network Performance, http://www.intel..com/network
[51] C. Liu, J. Layland, *Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment*, Journal of the ACM 20(1):46-61, 1973