CIDER - Envisaging a COTS Communication Infrastructure for Evolutionary Dependable Real-Time Systems

Mário Alves
ISEP-IPP, Portugal
malves@dee.isep.ipp.pt

Eduardo Tovar
ISEP-IPP, Portugal
emt@dei.isep.ipp.pt

Gerhard Fohler
Mälardalen University, Sweden
gfr@mdh.se

Giorgio Buttazzo
University of Pavia, Italy
giorgio@sssup.it

Abstract

It is foreseen that future dependable real-time systems will also have to meet flexibility, adaptability and reconfigurability requirements. Considering the distributed nature of these computing systems, a communication infrastructure that permits to fulfil all those requirements is thus of major importance. Although Ethernet has been used primarily as an information network, there is a strong belief that some very recent technological advances will enable its use in dependable applications with real-time requirements. Indeed, several recently standardised mechanisms associated with Switched-Ethernet seem to be promising to enable communication infrastructures to support hard real-time, reliability and flexible distributed applications. This paper describes the motivation and the work being developed within the CIDER (Communication Infrastructure for Dependable Evolvable Real-Time Systems) project, which envisages the use of COTS Ethernet as an enabling technology for future dependable real-time systems. It is foreseen that the CIDER approach will constitute a relevant stream of research since it will bring together cutting edge research in the field of real-time and dependable distributed systems and the industrial eagerness to expand Ethernet responsibilities to support dependable real-time applications.

1. Motivation

The current design practices followed in real-time systems are quite inadequate to address several issues such as complexity, adaptability, and reconfiguration requirements of future systems. The main drawback is that present systems typically assume fixed hardware structures with fixed assignment of tasks and hence are not suitable for the evolutionary process of dependable systems with long life periods.

Indeed, there is an increased number of dependable real-time systems which need to face demands for changes during their lifetime: environment and controlled processes undergo different operational modes and reliability requirements, process improvements necessitate modification of parts of the real-time computing system itself, equipment is upgraded or repaired.

Current solutions focus on pre-planned, fixed systems and applications. In particular, handling of changes in the hardware structure of the system, such as dynamically arriving nodes has not been given attention in a real-time context.

In [1] the authors describe a "Self-Evolving Systems" approach in order to impart flexibility into the typically rigid design methodologies. Their approach is based on two basic principles: evolutionary nature and separation between services and server. The basic principles are the following. Services broadcast admission requests to servers, and assign themselves to servers that reply positively. Service requests include processing, networking, and reliability demands. A server is added by connecting itself to the network, listening, and accepting services. Should a server node be removed or fail, its services reassign themselves to other server(s). In an initialisation phase, a guaranteed core of services can be configured, allowing for a trade-off between determinism and flexibility.

This approach opens a vast and innovative field of research where there are a number of open issues that need to be critically examined. Indeed, the work described in [1] is more at the conceptual level and from the real-time and dependability systems perspective. The CIDER project relates to this approach by focusing research on aspects related to the communication infrastructure.

Though in terms of technologies the main focus of CIDER will be on Switched-Ethernet technologies, also from the implementation perspective, middleware platforms such as Jini [2], proposed by Sun Microsystems as a general purpose networking extension of Java [3] or Inferno Networks [4], proposed by Lucent Technologies, will be looked at in terms of assessing their ability to encompass the Self-Evolving Systems approach. Later (Section 6.4), these technologies will be briefly discussed in the CIDER context. Additionally, this evolutionary system's nature inherent to CIDER demands for innovative solutions in scheduling.

2. New Paradigms in Scheduling

A change in the environment may cause events to arrive more frequently or even trigger new computational activities as a reaction, hence affecting the activation rates and the load distribution in the system (both in the processors and in the network). A change in the system structure may depend on maintenance, network re-configuration, new hardware installation, or component failures.

In both cases, the system should ensure a minimum level of performance for the most critical services, while providing the best possible quality of service for the non-critical ones. Performance is defined in terms of functionality, dependability and timeliness.

From the processor perspective, the possibility of varying the tasks' rates increases the flexibility of the system in handling overload conditions, providing a more general admission control mechanism. For example, whenever a new task cannot be guaranteed by the system, instead of rejecting it, the system can try to reduce the utilisation of the other tasks (by increasing their periods in a controlled fashion), in order to decrease the total load and accommodate the new request.

Unfortunately, there is no uniform approach to deal with these situations. Several methods have been proposed to cope with dynamic environments. However, the degree of flexibility exhibited by the algorithms proposed in the literature (e.g., [5][6][7]) is still insufficient to support highly adaptive systems.

This is another important research area that will be developed in the context of CIDER. Again, though the CIDER focuses on networking technologies, these will be assessed also in terms of their ability to support flexible scheduling and coexistence of critical/non-critical services.

3. Tech. Advances in COTS LAN Networks

Today, Ethernet is used primarily as an information network. However, there is a strong belief that some recent technological advances will enable its use in hard real-time dependable applications. One of the most common arguments that has been traditionally put forward against its use in real-time applications is that Ethernet has a non-deterministic access delay, leading to an unpredictable timing behaviour of the supported applications.

An important recent development concerning Ethernet technologies has been in the switching domain. Mechanisms such as micro-segmentation, full-duplex operation, traffic prioritisation, on-the-fly commutation, flow control or spanning-tree protocols are emerging features that create an eagerness to expand Ethernet responsibilities to support dependable applications with hard real-time requirements.

Switched-Ethernet may also enable crucial system's aspects concerning the flexibility and re-configurability requirements. Features such as dynamic switching, auto negotiation, automatic load balancing and Virtual LANs are in the authors' opinion worthwhile to be explored in that direction.

Additionally, there is an eagerness in technology providers to work closely with end users in order to determine whether their real-time industrial applications would benefit from using recent Ethernet technologies. Moreover, the Industrial Automation Networking Alliance - IAONA [8] - and its recently established (November 1999) European-based sister organisation - IAONA Europe [9] - aim establishing Ethernet as the standard in the industrial environment. The Industrial Ethernet Association - IEA [10], formed in (early) 1999, intends to establish standards for the use of Ethernet products in the industrial marketplace. It will be our concern, during the CIDER project, both to keep updated with the outputs from these initiatives as well as making available to them relevant results achieved in CIDER.

4. Why Ethernet?

The CIDER approach is a kind of "back to the future".

The first Ethernet specification was approved and released as IEEE Std 802.3 [11] back in 1983. Since then Ethernet has become the most popular LAN technology. Despite its popularity and low-cost, Ethernet has a serious drawback concerning the support of real-time distributed systems. Since in an Ethernet LAN messages may collide with each other, it is difficult to guarantee predictable delays in delivering messages to network nodes. Therefore, and since the early 80's, several other networks have been introduced (and most times standardised), having in mind the support of hard real-time distributed applications (e.g., [12][13]).

However, and again because Ethernet has always drawn a significant interest due to its extremely low price, maturity and stability, real-time and Ethernet have not always being apart. Several methods have been proposed in the literature to improve the time characteristics of Ethernet networks. The simplest solution is to use a collision-free environment by means of a TDMA strategy [14]. The inherent disadvantages are non-flexibility and inefficiency. Other approaches use token-passing strategies [15][16]. However, such mechanisms are either incompatible with standard hardware or very inefficient. Off-line scheduling strategies [17] (also collision-free) impose significant overheads in messages.

While these methods lead to a collision-free behaviour, some others focus on modifications of the collision resolution algorithm. The simplest [18][19] manipulate the collision counter to avoid the capture effect. However, they do not provide a deterministic behaviour. A deterministic collision resolution approach has also been proposed [20], however demanding firmware modifications in Ethernet boards. Finally, three classes of methods constrain traffic generation: Virtual Time Protocols (VTP), Window Protocols (WP), and Traffic Smoothing (TS). VTPs use packet release delay strategies as function of laxity [21] or priority [22]. WPs [23] use a global dynamic window to schedule medium access. Traffic smoothing [24] constrains non-real-time traffic bursts. None of these methods enable guaranteed approaches for real-time design.

So, what is the future of Ethernet to support the "future dependable real-time systems"?

5. Because…

Recently, there have been promising technological breakthroughs in Ethernet. Ethernet switches provide flexible and scalable solutions through the use of micro-segmentation and full-duplex operation. Simultaneously, IEEE 802.1p (ratified in September 1998) [25] gives Layer 2 switches the ability to prioritise Ethernet traffic and the IEEE Std 802.3x Flow Control [11] provides means to control the generated traffic. All these mechanisms may be exploited to improve the timing characteristics of Ethernet networks.

Additionally, the use of Switched-Ethernet can be exploited to support evolutionary dependable real-time systems. The IEEE Std 802.1D (Spanning Tree Protocol) [25] can be used to provide redundant network paths. Port Trunking establishes backbone links by treating multiple links as a single network pipe, also providing link redundancy. Virtual LANs (IEEE 802.1Q) [26] are very promising to deal with flexibility and mobility. Using VLANs, a workgroup can be defined not by physical locations, but by multicast filters. If a device moves within the switched network, multicast filters change automatically, maintaining logical connectivity. Furthermore, other features such as dynamic switching [27], auto negotiation [11], automatic load balancing [28] can also be exploited to enable the dynamical behaviour of the computer system.

The following table summarises the major technological achievements that may enable the use of Switched-Ethernet in dependable and adaptable real-time systems.

Input	**Description**	**Pros for**
Fast and Gigabit	Fast Ethernet (100 Mbit/s) and Gigabit Ethernet (1 Gbit/s) are already available	Real-Time
Micro-segmentation	Each station has a dedicated switch port	Real-Time
Full duplex (with micro-segmentation)	Simultaneous communication between switch and station, in both directions.	Real-Time
Traffic prioritisation	Switches group incoming packets into separate traffic classes (eight priority levels – 0 to 7).	Real-Time
Flow control	Flow control eliminates dropped packets on congested ports by “warning” stations that are overloading the switch.	Real-Time and Dependability
Spanning Tree Protocol	Management of redundant paths	Dependability
Auto-Negotiation	Auto-negotiation of data-rate and duplex mode	Adaptability
Virtual LANs	Definition of logical groups of stations	Adaptability
Automatic Load Balancing	Moving stations between segments in order to minimise traffic in the switch	Real-Time and Adaptability

Since these technologies are quite recent, there is not much research work on the analysis of real-time and dependable communications yet. The case of a centralised control system [29] has been studied for different topologies, in the context of the simple case of polling strategies. Also, a systematic load analysis of switched Ethernet networks is presented in [30], where the network topology is modelled by a graph and the network traffic represented in the form of a matrix. However, to our best knowledge, there is no work on the schedulability analysis of real-time traffic considering the recent technological breakthroughs, and being able to encompass flexibility, adaptability and reconfigurability. Although results are available concerning on-line scheduling algorithms [31], these most usually consider slot-shifting approaches (to have time-bounded on-line algorithms), which may show to be inappropriate for Ethernet networks.

6. Ongoing Work / Future Work

The ongoing work and future work within CIDER will develop along four major research lines, all with a particular focus in communication technologies. These are briefly described below.

6.1. On Switched-Ethernet Technologies

The assessment of the recently introduced Ethernet standards and technologies, not only as regarding their potential to improve the timing characteristics of Ethernet-based networks, but also as regarding their capabilities to support system's flexibility, adaptability, reconfigurability and dependability.

6.2. On Scheduling Strategies

Traffic schedulability analysis will be a main focus. Typically, these are worst-case analysis allowing the determination of upper bounds to the transmission delay of messages. Such studies have not yet been developed for the Ethernet context wherein unpredictability is avoided by using the techniques recently made available for Ethernet (traffic prioritisation, micro-segmentation and full-duplex operation).

An important aspect that must be considered is the evolutionary nature of the system. Hence, the analysis that will be developed must be executable on-line so that it can be invoked whenever there is a demand for a change in the system configuration. These algorithms have to be time-bounded, and their bound must be sufficiently small to produce feasible systems. Flexible scheduling and coexistence of critical/non critical traffic will be targeted, taking into account mechanisms recently made available for Ethernet (automatic load balancing, flow-control, traffic prioritisation VLANS).

6.3. On Dependability Issues

Several fault-tolerance mechanisms are available through spanning trees and port trunking. The Spanning Tree Protocol [25] can be used to provide redundant network paths [32], still protecting against network loops. The spanning tree algorithm may allow the design of redundant links [33]. Port Trunking establishes backbone links by treating multiple parallel links as a single network pipe [32]. 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.

Finally, it is important to note that the broadcast mechanism in the network is a central idea for achieving the ideas described in Section 1.

6.4. On Upper-Level Implementation Aspects

Possible implementations of the concepts described in Section 1 may rely on the use of emerging platforms such as Jini/Java or Inferno. Although this is a less import part of the CIDER research effort, the potential of these platforms is outlined next.

Sun's Jini technology, enables Java-based devices to communicate over any IP-based network. Jini infrastructure provides resources for executing Java language objects, communication facilities between those objects, and the ability to find and exploit services on the network. By using the Java's RMI (Remote Method Invocation), Jini infrastructure provides communication between objects across device boundaries that enables those objects to work together. RMI enables activation of objects and the use of multicast to contact replicated objects, providing highly reliable objects to be easily implemented in the Jini framework. Jini technology provides a lookup service allowing services connected by the communication infrastructure to be found. Jini infrastructure provides a mechanism - called discovery/join - for Jini-enabled devices to discover the appropriate lookup service and join into the overall Jini system. When a device joins a Jini system, its services are added to that lookup service. Symmetrically, when a Jini enabled device leaves a Jini system (by being removed or becoming faulty) its services are deleted from the lookup service. Jini technology also provides the possibility to compose systems to meet specific requirements, rather than relying on a general-purpose system.

All these aspects of Jini/Java go very much along with the requirements of the self-evolving systems approach. Still, the separation between server and services in the self-evolving systems approach do not exactly match the concept of a Jini system: a Jini system represents a set of available services - some are software and others are hardware.

Concerning real-time aspects, there is still the open issue of real-time Java. Java is a heap-oriented language with a garbage collection mechanism. The use of a garbage collector results in a non-deterministic timing behaviour of Java applications, which is a major drawback for a real-time system. There is of course the effort to get a real-time Java (J-Consortium [34]). The introduction of a deterministic garbage collector, the language extension to cope with hardware interfacing and the introduction of more elaborated concurrency mechanisms are being considered. However, deeply modifying the Java language may reduce some of its strong arguments: portability and simplicity.

Lucent's InfernoSpaces / InfernoNetworks [4] are also promising and are briefly discussed next.

While Java is a programming language, Inferno is a full network operating system including a kernel, programming language, communications protocols, libraries, security and authentication, naming protocols, APIs, etc. Inferno is therefore a more comprehensive offering, one that addresses more of the problems of building networked applications. Moreover, although Inferno includes a programming language, called Limbo, it is expected to support others, and Java is an obvious candidate.

Concerning the real-time aspects of Limbo, concurrency and communication are an intrinsic part of the language and of the virtual machine, and are used extensively in the programming model (concurrency support better than in Java). Limbo's garbage collector has a constant overhead, so its operation does not conflict with real-time constraints. Also, the collector reclaims memory as soon as the last reference is released, in order to minimise the memory needed for execution. Limbo completely manages the lifetime of system resources by tying windows, network connections, and file descriptors to the garbage collector.

Concerning the supported networks, an Inferno application can access the resources and services in its name space even though they may be distributed throughout the network. Additionally, the Styx component of Inferno provides transparent communications over a variety of networks with strong built-in security capabilities.

7. References

[1] G. Fohler, S. 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.

[2] Sun Microsystems, *Jini Architectural Overview*, Technical White Paper, Sun Microsystems, 1999.

[3] R. Atherton, *Moving Java to the Factory*, IEEE Spectrum, pp. 18-23, December 1998.

[4] S. Doward, R. Pike, D. Presotto, D. Ritchie, H. Trickey, P. Winterbottom, *The Inferno Operating System*, Bell Labs Technical Journal, pp. 5-18, Winter 1997.

[5] T. Nakajima, *Resource Reservation for Adaptive QOS Mapping in Real-Time Mach*, in Proceedings of the Sixth International Workshop on Parallel and Distributed Real-Time Systems, April 1998.

[6] H. Kaneko, J. Stankovic, S. Sen, K. Ramamritham, *Integrated scheduling of multimedia and hard real-time tasks*, in Proceedings of the IEEE Real-Time Systems Symposium (RTSS '96), December 1996.

[7] T. Abdelzaher, E. Atkins, K. Shin, *QoS Negotiation in Real-Time Systems and Its Applications to Automated Flight Control'*, in Proceedings of the IEEE Real-Time Technology and Applications Symposium, Montreal, Canada, June 1997.

[8] IAONA – Industrial Automation Networking Alliance, USA, http://www.iaona.com

[9] IAONA Europe – Industrial Automation Networking Alliance, Europe, http://www.iaona-eu.com

[10] IEA – Industrial Ethernet Association, http://www.industrialethernet.com

[11] 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.

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

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

[14] H. Kopetz, A. Damm, Ch. Koza, M. Mulazzani, W. Schwabl, C. Senft, R. Zailinger, *Distributed Fault-Tolerant Real-Time Systems: The MARS Approach*, IEEE Micro, 9(1):25-40, February 1989.

[15] 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.

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

[17] 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.

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

[19] M. 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).

[20] 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.

[21] 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.

[22] 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.

[23] W. Zhao, J. Stankovic, *A window protocol for transmission of time-constrained messages*, Proc. of IEEE Transactions on Computers, Vol. 39, No. 9, pp. 1186-1203, September 1990.

[24] S-K. Kweon, K. Shin, Q. 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] (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.

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

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

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

[29] 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.

[30] 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.

[31] G. Fohler, *Adaptive Fault-tolerance with Statically Scheduled Real-Time Systems*, in Proceedings of Euromicro Workshop on Real-Time Systems, pp. 161-167, 1997.

[32] 3COM, *Superstack II Features*, http://www.3com.com/products/dsheets/400260a.html, 1999.

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

[34] The J-Consortium - Accelerating Java technology into Real-Time. http://www.j-consortium.org/