Designing, Modelling and Evaluating Switched Ethernet Networks in Factory Communication Systems

N. Krommenacker, J.P. Georges, E. Rondeau, T. Divoux
CRAN – CNRS UMR 7039, Université Henri Poincaré - Nancy 1
BP 239 - 54506 Vandoeuvre lès Nancy - FRANCE
E-mail : <nicolas.krommenacker@cran.uhp-nancy.fr

Abstract

The Ethernet network is increasingly used for the industrial communications which are strongly time-constrained. This paper presents both a method (based on the genetic algorithms) to design switched Ethernet architectures and a method (based on the network calculus) to evaluate these network architectures in term of maximum end-to-end delay.

1. Introduction

The industrial communications are currently based on specific networks called fieldbuses such as FIP, Profibus, CAN. They interconnect programmable controllers, CNC, robots, … to exchange technical data for monitoring, controlling, and synchronising industrial processes. Their main characteristic is to ensure that the end-to-end delays of messages remain limited, compared with the time-cycle of the applications. Thus, these networks are deterministic and some protocols satisfy the integrity constraints of the information. In opposition, the Ethernet networks based on the CSMA/CD protocol are increasingly used to interconnect industrial devices. Some applications confirm this evolution in different industrial areas : cars (Jaguar), pharmaceutics (Boehringer Ingelheim),… Moreover, different organisations such as the Industrial Automation Networking Alliance (IAONA), the Industrial Ethernet Association (IEA) promote Ethernet as "the standard in the industrial environment". Finally several research projects such as CIDER (Communication Infrastructure for Dependable Evolvable Real-Time Systems) [1] study the use of Ethernet as an enabling technology for future dependable real-time systems.

In the first part of this paper, we propose to design a switched Ethernet architecture which satisfies the time requirements. Our objective is to optimise the network organisation at the physical layer level. To define efficient topologies, we use graph partitioning techniques in order to distribute industrial devices on the different Ethernet switches. The graph partitioning is an NP-complete problem for which several heuristics have been developed. The size of the search space for an arbitrary problem instance is defined by the number of nodes and partitions. We can distinguish two types of approaches. The constructive methods elaborate a solution *ex nihilo* while the improvement methods try to improve a given solution with the help of a heuristic. In this study, we suggest the use of such a heuristic : the Genetic Algorithms.

In the second part, we describe an approach to evaluate the network organisation obtained from the network design step based on genetic algorithms. In the literature, several works use a stochastic modelling in order to evaluate the switched Ethernet performances. Nevertheless, these surveys study the communication system with input services such as Bernoulli processes, Poisson processes, … which are not representative to the messages sent by the industrial devices. Basically, a programmable controller periodically sends data on the network with critical temporal constraints and also sends aperiodic messages. Thus we propose to model these communications by using a strict arrival curve and in particular by using the network calculus. The network calculus introduced by R. Cruz [2][3] only assumes that the number of bytes sent on the network links does not exceed an arrival curve service (traditionally, a leaky bucket) which can represent both the periodic and the aperiodic traffic. The other interest to use the network calculus is to be able to model the Ethernet switches and their interconnection by assembling basic components such as multiplexer, demultiplexer, buffer, … whose temporal properties were given by R. Cruz. The network calculus enables then to determine the packet maximum end-to-end delay or jitters from exchange matrices which model the traffic between the industrial devices.

Finally, we study in this paper a communication scenario between programmable controllers in order to illustrate the interests of our proposals.

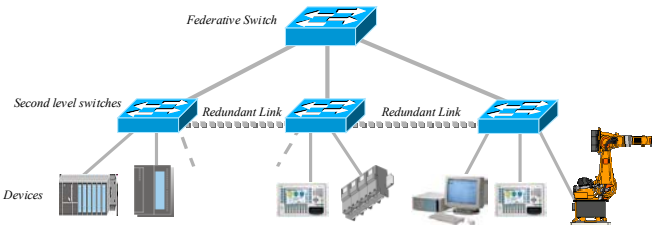

Figure 1. Network Topology

2. Designing switched Ethernet networks

2.1. Introduction

It exists two main kinds of topologies to connect the industrial devices on switched Ethernet networks. The most common are the hierarchical and the linear topologies. In this paper, we analyse network architectures based on these two topologies (figure 1). The hierarchical topology is activated in the normal functioning mode and when it is out of order, the linear topology is activated in order to guarantee the network connectivity.

The general method to design local area network always consists both in concentrating the traffic inside each network segment and in balancing the load between the different segments. For this, the exchanges are represented on a graph where the vertices correspond to the manufacturing devices and the weighted edges model the quantity of communication between them. Then, the network segmentation step is achieved by using graph partitioning techniques. The originality of our approach, in opposition to previous works [4][5], is to define a fitness function which is very efficient since in a quick time we can obtain a good network architecture.

We present in the next sections, the main principles to design network architecture by using the genetic algorithms[6].

2.2. Coding structure and Parameters

For the graph partitioning problem, several encoding methods exist. The method that is the most often used is the N-string representation. Each chromosome corresponds to a vector in which the i^{th} element of an individual is j if the j^{th} vertex of the graph is allocated to the partition labelled j. There is as many elements in the vector as vertices in the graph. The multiway partitioning problem uses the k-ary alphabet [0,1,…,k-1]. For instance, the string [001221120] represents the mapping that assigns vertices 1, 2, 9 to partition 0, vertices 3, 6, 7 to partition 1 and vertices 4, 5, 8 to partition 2. The figure 2 illustrates the encoding of this solution and the decoding of this solution as a hierarchical topology with redundancy.

[6] defined conventional GA parameters such as population size (pop_{size}), crossover (p_c) and mutation (p_m) probability. These parameters influence the performance of the algorithm. For example, the size of the initial population influences on the quality of the final solution to the detriment of the execution time. For this study, the good results were found with the following parameters : pop_{size}=15, p_c=0.2 and max_{gen}=500.

To start the GA, the population has to be initialised. The common method is to create solutions at random. Nevertheless, the random method must be controlled in order to avoid to create absurd chromosomes such as [002220020] or [11111111] where the final expected number of groups is not respected. For the selection step, the "roulette wheel" approach has been chosen. This is a very common parent selection method where each chromosome selected as a parent has a probability that is proportional to its fitness. Moreover, a elitist strategy is used to keep the best individuals from the current generation to the next generation.

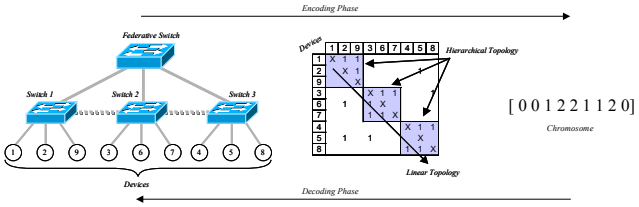

Figure 2. Encoding Network Topology

2.2. Crossover and Mutation operators

In GA, the crossover and mutation operators are the two most important space exploration operators. A crossover operator creates a new offspring chromosome by combining parts of the two parent chromosomes. A two-point crossover is employed in our algorithm. Two cut points are randomly selected which are the same on both parent chromosomes. Each chromosome is also divided into three disjoint parts. A new chromosome is formed randomly copying a part from one parent to the corresponding part of the offspring (see figure 3).

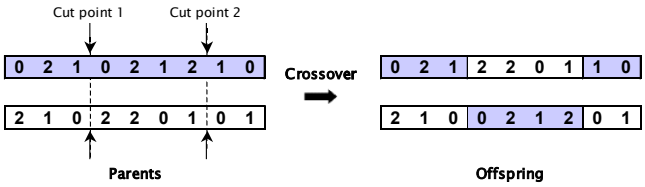

Figure 3. Two-point crossover operator

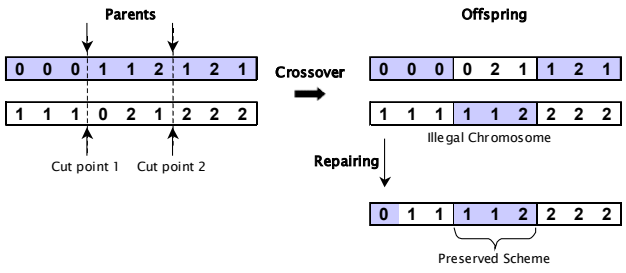

Figure 4. Repairing step

The crossover operator can generate illegal chromosomes as in the initialisation step. In the figure 4, an offspring violated the number of group. We propose a repairing procedure which consists in checking the group

number constraint. If a chromosome violates this constraint, it is repaired with permutations from parent chromosomes. The permutations are applied out of the cut-points in order to preserve the scheme of parent chromosome.

The mutation operator enables to introduce unexplored search space to the population. The method that is the most often used is the bit-flop mutation. However, a swap mutation is used in order to control the constraints of the group number.

2.3. Fitness function

To determine the quality of a solution, we have used two metrics. The first one is the edge-cut criterion which represents the volume of the inter-group exchanges. It is the sum of all the edges which connect a vertex in one partition to a vertex in an another partition. The second one is the set size criterion which measures the balance between groups. [7] defined a grouping efficacy measure that minimises the number of outside elements of the partitions (exceptional elements) and void elements inside the partitions. For each solution, a fitness value is calculated with the following fitness function :

$$f(e, e_V, e_0) = \frac{e - e_0}{e + e_V}$$

with :

- e : sum of elements inside matrix
- e_0 : sum of elements outside partitions
- e_v : sum of null elements inside partitions

Thus, the objective is to identify a network architecture which maximises the fitness function f.

3. Modelling switched Ethernet networks

In this section, we propose to model an Ethernet switch. [8] and [9] decompose the switching architecture in three main components :

- the queuing model refers to the buffering and the congestion mechanisms located in the switch. In our modelling, we consider an organisation based on the shared memory queuing.
- the switching implementation refers to the decision making process within the switch (how and where a switching decision is made). We select a centralised switching implementation.
- And the switching fabric is the path that data takes to move from one port to another. From previous choices, the shared-memory architecture is chosen.

The figure 5 lists the different components retained in this paper to model a switching architecture. It is constituted of a sequence of three components : one multiplexer, one FIFO queue and one demultiplexer. This model could be applied to different industry products, like the Cisco Catalyst 2009XL. The interconnection of the switches enabling to build the network architecture is achieved by using links which are modelled by buffers. The link transmission mode in this work is supposed to be the full-duplex mode in order to avoid collision problems introduced in the half-duplex mode.

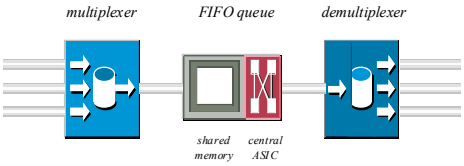

Figure 5. Switch modelling

4. Evaluating switched Ethernet networks

4.1. Traffic modelling

In the industrial context, there are two kinds of communications between the devices : the periodic and aperiodic messages. To represent the traffic, we use the leaky bucket controller concept (definition 1.3.2 in [10]). It imposes the traffic generation to be bounded by an affine function named leaky bucket, noted b(t), in which a variable burst value σ is associated to a constant rate ρ. [2] proposes also to associate a burstiness constraint based on the leaky bucket to each stream and in which : $b(t)=\sigma+\rho t$

$$\text{And } R(t)<b(t) \Leftrightarrow \int_{t=x}^{y} R(t)dt < \sigma + \rho(y - x)$$

Where :

- R(t) represents the instantaneous rate of all traffics merged (periodic and aperiodic) from the stream,
- σ is the maximum amount of traffic that can arrive in a burst,
- ρ is an upper bound on the long-term average rate of the traffic flow.

This peakedness characterisation of the traffic presents two advantages. First, the leaky bucket gives a deterministic representation since we are sure that the number of arrived bytes from a stream i at a time t is bounded by $b_i(t)$. Secondly, as the leaky bucket form is an affine function, it is possible to merge periodic and aperiodic data. Since the leaky bucket function is affine, the number of data transmitted by the programmable controller is bounded by : $b_{total}(t) = b_{periodic}(t) + b_{aperiodic}(t)$.

4.2. End-to-end delay

We have defined a general model of a switched Ethernet architecture which is based on four components (multiplexer, FIFO queue, demultiplexer and buffer). The network calculus developed by R. Cruz enables to obtain the upper bounded of end-to-end delay (with a lucky

bucket modelling) for each component. To determine the maximum end-to-end delay in a complete switched communication system, we must add the different maximum end-to-end delays. The general formula which we obtain is :

$$\overline{D_{switch,i}} = \alpha\left[\sigma_1 + \sigma_2 + ... + \beta_{\sigma_1} + ... + \Lambda\sigma_K + ... + \sigma_M - \delta\right]$$

and is explained in [11]

5. Example

To illustrate the behaviour of the network design algorithm proposed in this paper, we study a traffic matrix which collects the information exchanges between 40 programmable controllers (Figure 6). The whole messages is represented by a binary matrix and each n in the matrix corresponds to one packet sent every 5 milliseconds. The size of each packet is equal to 64 bytes (minimal size of a Ethernet packet). We use 12-ports Ethernet switches with full-duplex links at 10Mb/s. The objective is to find a good distribution of the programmable controllers on five switches.

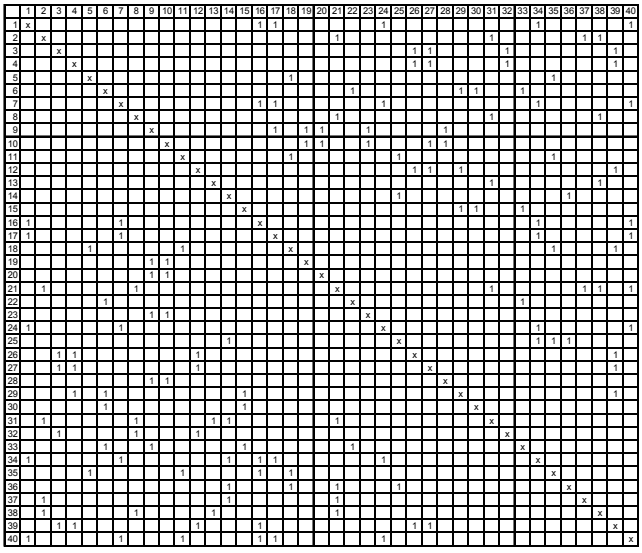

Figure 6. Exchange matrix

Before analysing the topology obtained from our algorithm, we have searched a bad partitioning which is composed of five groups constituted of the following programmable controllers : 1-9, 10-17, 18-26, 27-33 and 34-40. In this solution, all the communication are inter-group communications. The total traffic is outside partitions and the fitness function is also null. The mapping of this solution corresponds to an architecture with a federative switch which supports all the traffic defined in the matrix. The genetic algorithm proposed in this paper is implemented in Java language. Since GA is a stochastic search algorithm, it is performed 20 times. The best partition obtained is : {1,7,16,17,24,34,40}, {3,4,12,15,26,27,29,30,32,39}, {5,11,14,18,25,35,36}, {2,8,13,21,31,37,38}, {6,9,10,19,20,22,23,28,33}

From the bad and good organisations, we can compare the end-to-end delays by using the network calculus. We find for the bad solution a maximum end-to-end delay equal to 914 µs and the means of all maximum end-to-end delays is equal to 819 µs. The same operations applied to the good solution give respectively 880 µs and 500 µs. We observe a significant difference since between the bad and the good organisations the means of all maximum end-to-end delays are divided by two

6. Conclusion

This paper proposes a general method to design and evaluate a switched Ethernet architecture for real-time applications. The next steps of this work are to implement in the switch model, some mechanisms of Quality Of service such as priority as defined in 802.1p and to introduce directly in the fitness function the maximum end-to-end delay formula.

7. References

[1] M. Alves, E. Tovar, G. Fohler, G. Buttazo "CIDER – envisaging a cots communication infrastructure for evolutionnary dependable real-time", *12th Euromicro conference on Real-Time Systems*, Stockolm, pp.19-22.

[2] R. Cruz, "A calculus for network delay, Part I : Network elements in isolation " *IEEE Transactions on Information Theory.* Vol. 37, pp. 114-131, 1991

[3] R. Cruz, "A calculus for network delay, Part II : Network analysis" *IEEE Transactions on Information Theory.* Vol. 37, pp. 132-141, 1991

[4] R. Elbaum, R. Sidi "Topological design of local area networks using genetic algorithms", *IEEE/ACM Transactions on Networking*, Vol. 4, n°5, pp.766-778, 1996

[5] M. Nusekabel "Switched network partitioning using Tabu search", *Thesis for Master of Science in Computer Science*, University of South Florida, 1997.

[6] D.E. Goldberg "Genetic algorithms in search optimization, and machine learning". *Addison Wesley*, 1989.

[7] C.S. Kumar, M.P. Chandrasekharan "Grouping efficacy : a quantitative criterion for goodness of block diagonal forms of binary matrices in group technology" *Journal of Production Research*, Vol. 28, n°2, pp. 233-243, 1990.

[8] M. Philipps "A guide to LAN switch architectures and performance" Packet CISCO Systems users magazine, pp.54-57, 1999

[9] R. Seifert "The switch book" *John Wiley and sons*, New-york, 2000

[10] J.Y. Le Boudec, P. Thirian "Network calculus" *Springer Verlag*, 1990.

[11] J.P. Georges, E. Rondeau, T. Divoux "Using the network calculus to evaluate the performances of a switched Ethernet network in an industrial context" *Internal report*, CRAN, 2002.