TEMPORAL MATCH OF MULTIPLE SOURCE DATA IN AN ETHERNET BASED INDUSTRIAL ENVIRONMENT

Daniela Hossu and Andrei Hossu
University Politehnica of Bucharest, Faculty of Control and Computers
313 Spl. Independentei, sector 6, RO-77206, Bucharest, Romania
dana@aii.pub.ro, hossu@aii.pub.ro

Keywords: Industrial Robotic Application, Ethernet-based communication, Data transfer consistency, Artificial Vision System, Control Management System, Routing Control System.

Abstract: The actual stream in automation control systems is to distribute the control tasks among different modular and easy to integrate processing cells. It is part of this trend the increase of the use of Ethernet technology for machine-machine data communication inside this distributed based architecture. The paper presents a robotic handling application of industrial parts transferred by a transport conveyor. Data representing a set of parameters of the parts to be handle from the conveyor is provided by a Routing Control System (RCS). The Control Management System (CMS), which controls a number of robotic cells is receiving this data from RCS and merge it with the information provided by an Inspection System (Artificial Vision System). The communication between these two Control Systems (RCS and CMS) is Ethernet-based. Ethernet technology is good, reliable and fast for large amount of data, but because of its non-deterministic character, it has a lack of tools for data synchronization. The paper includes an analysis of the experimental results of the measurements of the non-deterministic factor of the existing network. The "worst case scenario" of the largest communication delay caused by Ethernet traffic and the minimum time between two consecutive data commands, reveals that application requirements could not be achieved without recovering data transfer time-consistency. The paper is presenting a mechanism developed at protocol level, in order to guarantee the consistency in time, at CMS level (data consumer), of the merging process of the data provided by the two application partners, RCS and Vision System as the data producers.

1 THE DESCRIPTION OF THE ROBOTIC APPLICATION

The system described in the paper is dedicated for the robot-based automation of the unloading and packing stages in the flat glass industry. In Figure 1 it is presented the architecture of the proposed automation system. This architecture is often utilized in industrial applications (in palletizing of moving objects systems).

1.1 The Structural System Architecture

▪*Active Elements:*
Control Management System (CMS), Routing Control System (RCS), Vision System and Robotic Cells.

▪*Passive Elements:*
Conveyor, glass plates.

▪*Infrastructure:*
Communicational Link between Vision System and CMS;
Communicational Link between CMS and Robots Controllers;
Communicational Link between CMS and RCS.

▪*General assumptions:*
The plates are connected to the conveyor (the same speed and direction).

1.2 The Functional System Architecture

The Routing Control System has to provide for CMS the Routing Data - a description of the possible destinations (one or more of the robotic cells) of each plate in the moment the plate is passing the Decision Point of the Vision System. The role of the Vision System is to inspect the cutting accuracy and the shape parameters of every plate. The Vision

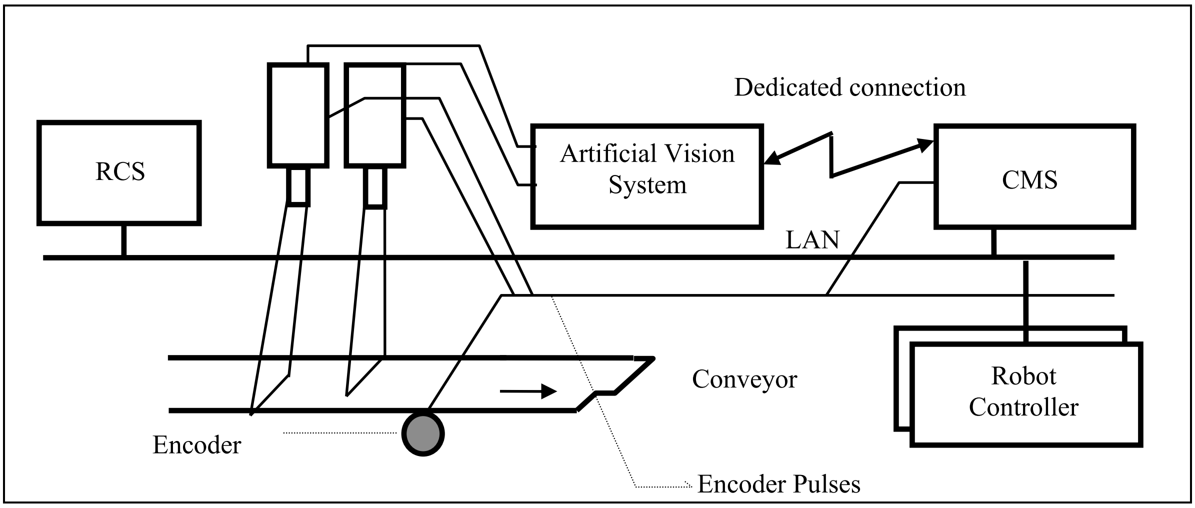

Figure 1: The robot-based automation system for handling glass plates from a moving conveyor.

System is analyzing the information provided by a Line Scan Acquisition System (a dual line scan camera system) in conjunction with the information provided by an encoder connected to the transport conveyor. The Vision Data, containing the data resulted from the inspection process, together with the data describing the location of the plate, are transmitted to CMS in the moment the Vision System processing time ended.

The moment (time-based) is called Vision Decision Point. Both sets of data (Routing Data and Vision Data) are merged by CMS. CMS will take the decision to send the pick the plate command to a certain robotic cell only if Vision Data describe the plate having cutting accuracy and shape parameters inside the accepted tolerances for a certain packing destination AND the plate is routed to that certain destination.

2 THE DESCRIPTION OF THE INFORMATIONAL SYSTEM ARCHITECTURE

The communication between Vision System and CMS is a dedicated connection. Some of the reasons for choosing this type of communication is the geographical neighborhood of these two systems and the fact the structure of the data transferred between Vision System and CMS could be very tight specified (no need for using a general type of protocol). Another reason is the concept that Vision System is an intelligent sensor of the Robotic Application, which means the Vision System will be not "visible" on the higher automation level, but only on the Robotic Automation Level (the Vision System is "visible" only for CMS). This communication channel has a serial support. This type of connection is providing a deterministic character of the Vision System – CMS communication. For the communication between CMS and RCS it was adopted an Ethernet-based topology. The main reason of choosing this type of topology is the fact CMS is an automation entity visible on the high automation level (CMS includes all the automated stacking capabilities of the whole production line). Industrial Protocol on Ethernet is a very good and reliable support for modern configurations on industrial automations, but because it's non-deterministic character, it is poor in data time-synchronization (Marshall, et al., 2004). The time-synchronization between the Routing Data (data coming from RCS) and the Vision Data (data coming from Vision) in the merging process in CMS it is a key factor of achieving the requirements of the automation application.

3 EXPERIMENTAL RESULTS

In Figure 2 are presented the experimental results of recording Ethernet Delays over around 10 minutes on the analyzed network. The Ethernet Delays are estimated as the differences between the CPU time of receiving Routing Data (sent over a non-deterministic communication channel) and the CPU time of receiving Vision Data (sent over a deterministic communication channel). A lost in

synchronization will occur only if the variation of the Ethernet Delay value from the average value of the Ethernet Delay is greater than the minimum time between two consecutive sets of Routing Data. Analyzing the manufacturing process we can identify what is the minimum time between two consecutive sets of Routing Data.
This is the minimum time between two glass plates coming on the conveyor (this is called “snapping period”). The actual glass manufacturing process has the minimum snapping period of 1.44 seconds. Analyzing the experimental results we could see the necessity of implementing a method for recovering data transfer time-consistency.

4 RECOVERING DATA TRANSFER TIME-CONSISTENCY

A few solutions were analyzed in order to solve this problem (Marshall, et al., 2004).:

- Use a more powerful Ethernet board (instead of using 10MB/s type of board, to use a 100BT Ethernet module)
- Replace the communication software support (RSLinx) with another one with a better response time (a software module dedicated only for a specific protocol would provide a better response time related to a general software package like RSLinx, which is coming with a large CPU overhead).

The above two solutions could improve the Ethernet behavior, but the non-deterministic character of this type of communication is not eliminated.

- Install another dedicated Ethernet module in the RCS and an additional dedicated Ethernet module in the CMS PC. These modules would be connected to a separate isolated Ethernet switch. In this case, most of the delays experienced on the current Ethernet link would be eliminated since the only traffic on the link would be between the routing system and the CMS cell PC.
- Use an ASCII serial (RS-232, RS-485, etc.) connection rather than using Ethernet. This would make the communication time between the Routing and CMS systems deterministic.
- Use a dedicated digital signal from RCS to the CMS in addition of the Ethernet connection in order be used to re-synchronize the Routing Data in CMS. This would be implemented by energizing a digital output that would indicate to the CMS cell in the moment of sending current Routing Data. When the input was seen by the CMS system, the CMS system would capture its own internal time and it will use this time value in the moment the Routing Data is received over Ethernet.

These last three solutions could solve the time-synchronization problem, but any of these solutions wouldn’t be accepted because of a dramatic aggression on the network topology previously agreed on the design time of the application (Mackay, et al., 2003), (Stenerson, et al., 2002).

The solution proposed in the paper is based on inserting a “*timestamp parameter*” in any set of Routing Data transmitted from the Routing System to the Control Management System.

This *timestamp parameter* will be the Routing System CPU time in millisecond representation. This timestamp parameter value will be a wraparound counter representing the least significant two bytes (one Word) of the CPU time (in milliseconds).

This *timestamp parameter* will be used by CMS to estimate with a “good enough” approximation, the offset between the CPU time values of the producer of the message (RCS) and the consumer of the message (CMS). This estimated offset would be used for time synchronization of the current message. It means CMS will add the estimated CPU time values offset to the current timestamp parameter value contained in the current received message.

In order to provide a “good enough” approximation of the offset between the CPU time values of the producer (RCS) and the consumer (CMS) the algorithm has to estimate the minimum value of the statistical population containing all the offsets estimated for a large number of transmitted/received messages.

This minimum is a “moving minimum” (it will be estimated from a statistical population collected on a certain time window) because we expect a slippage between the clocks of the two CPU (this slippage is accumulative and will become significant in time).

For the support of building this statistical population of offset values is used an existing message from RCS to CMS, called “Request Status Message”.

This message is sent by RCS every half a second in order to check the communication with CMS and also to obtain from the CMS the status of the

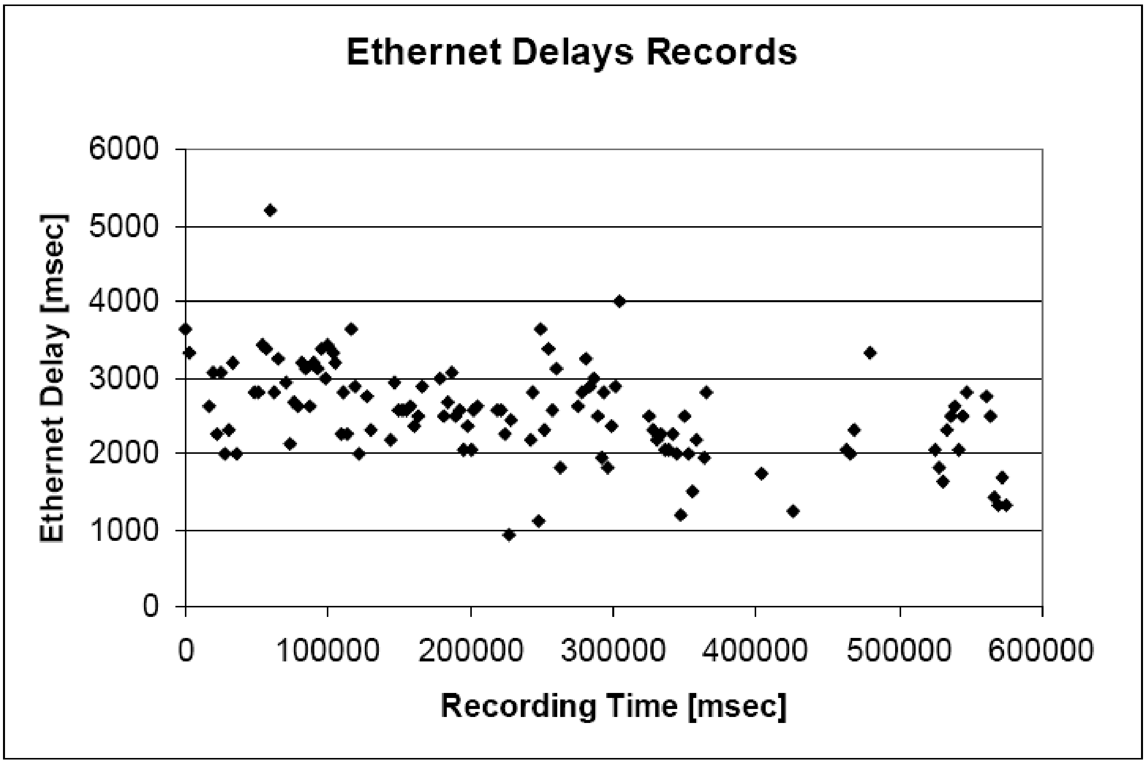

Figure 2: The experimental results of recording Ethernet Delays over around 10 minutes on the analyzed network.

availability of each possible plate destination stations (of each Robotic Cell).

The proposed solution requires the Routing System to move the CMS cell data hand-off point (the point where it is transmitting the Routing Data to CMS) more upstream the conveyor.

The CMS has to receive the Routing Data message of a plate before the Vision Systems ends the process of analyzing that plate (even on the highest Ethernet traffic). But the method is not anymore affected by receiving Routing Data in advance. CMS will build a buffer of all the Routing Data received from the RCS describing plates will come in time, and it will be able to recover the consistency in time of these data on the merging process with the Vision Data.

5 CONCLUSIONS

The paper presented an algorithm developed at protocol level, in order to guarantee the recovering of data transfer consistency in time, at CMS level, for merging of the data provided from RCS over an Ethernet communication channel with the data provided by the Vision System over a deterministic communication channel.

This solution proposed in this paper is based on the following assumptions:

- *The actual slippage of the clocks on either the RCS or the CMS processors would be very minimal*. This assumption is not a restrictive one, being normal to accumulate a significant slippage value around one second in much more that days or even weeks.
- *In the time the slippage value of the clocks would become significant, the network connection would have a period of relative low traffic*. This assumption is also not restrictive one, because in those days or weeks that the slippage value of the clocks is becoming significant, it is more likely a relative calm moment to occur in the net traffic.
- *The number of collected messages transmitted by the producer/received by the consumer (till the slippage of the CPU clocks will accumulate a significant value) will be large enough to build a statistical population*. This assumption is also not a restrictive one because the statistical population main support is the “Request Status Message” which is set to

be transmitted every half second (most of the statistical population members are coming from collecting the estimated offsets for this type of message).

REFERENCES

Mackay S., Wright E., Reynders D., Park J., 2003, *Practical Industrial Data Networks*, Elsevier.

Marshall P.S., Rinaldi J.S., 2004, *Industrial Ethernet (2nd Edition),* ISA – The Instrumentation, Systems, and Automation Society.

Stenerson J., 2002, *Industrial Automation and Process Control*, Prentice Hal.