Implementation of Networked Control Systems Using Programmable Controller Based Ethernet Network

Aris Ramadhan, Muhammad Ary Murti, Leanna Vidya Yovita
Faculty of Electrical and Communications
Telkom Institute of Technology (IT Telkom)
Bandung, Indonesia
aris_ramadhan@ieee.org; m.ary.murti@ieee.org; lvy@ittelkom.ac.id;

Abstract—**This paper presents the implementation of networked control systems (NCS) to be applied in industrial automation by using the programmable controller as the platform controller based Ethernet network. The fast growing in ICT technology on recent era brings some essential progress to support the development of NCS becomes more reliable to be implemented especially for critical delay application in industrial automation. The experiment, which is to measure the delay on the control signal that occur, the control execution accuracy and the cycle time of the programmable controller, had been conducted in four phase from identifying component of NCS and its characteristics, designing NCS, taking simulation of process workflow and simulation of the network and implement to physical system. The delay measure by monitoring the status of communication port of programmable controller. The measurement of control execution accuracy took after observation of process of removing the box from conveyor. The programmable controller's cycle time measurement obtained by capturing the cycle time data from its memory. The experiment results shows that the latest programmable controller along with recent network protocols has capability to deal with the delay and traffic. Delay in the network could decrease the control execution accuracy. The best control accuracy performances occur when total delay in the network are less than 1 s. Thus, the critical delay < 1 s. In other hand, the traffic condition does not have any impact related to the cycle time.**

Keywords—Networked Control Systems (NCS); Industrial Automation; Programmable Controller; Ethernet network; delay

I. INTRODUCTION

NCS (Networked Control Systems), the candidate of future control systems as mention in [1], has got many attention recently. The rapid growing of information and communications technology (ICT) nowadays also giving their support in the development of NCS. In industrial automation systems, there is a challenges in integrating the production planning, and control, while improving the quality management, and maintenance to become more efficient, etc. so the automation systems will be better, more efficient and in the end will reduce the cost. NCS with better performances together with better communications are one of the solution. NCS offers many benefits that can't be provided by traditional control and automation systems i.e. easy to install and maintenance, less wiring, flexibility in design and upgrade later, and reducing the cost during installation, operation, and maintenance [2-6].

The platform controller commonly used in industrial automation systems is PLC (Programmable Logic Controller) or nowadays mention as Programmable Controller [13-14] because of its performance has improve significantly since using processor as its brain, not just replacing the mechanical relay, but the short name 'PLC' still in used today. PLC become very popular especially for industrial automation because of its strength and reliability to control many plants, sensors, and actuators. OMRON PLC comes with the latest network communications, e.g. Industrial Ethernet, Wireless ISA 100.11a, Ethernet/IP, DeviceNet, and EtherCAT, provide many options to the user to choose the most suitable network communications for their automation systems. Industrial Ethernet, as explain in [8, 17], are one of a good choice to be implemented in control especially for handling distributed systems in industrial automation better than the old fashioned network communications i.e. CAN, etc. In this paper, the network communications which is used are the Industrial Ethernet provided by OMRON, also called as OMRON FINS (Factory Interface Network Service) Ethernet. Detail info about FINS can be found in [12, 15].

The organization of the paper as follows. Section II describes the simple practical approach step by step in designing NCS with programmable controller. Section III explains the implementation phase of NCS in physical systems also build the network. Section IV shows the measurement results and delay analysis. Section V draws the conclusion and future work.

II. DESIGNING NCS

The basic differences between classic control systems and networked control systems is that control signal in classic control systems like sensor-to-controller and controller-to-actuator are connected directly or wired so that the control signal must be in real time. But, in NCS, the control signal must travelled across the network as shown in Fig. 1. Thus the signal are no longer in real time because network has its own characteristics i.e. delay, jitter, packet dropout, traffic, etc. So, in order to gain the benefit of NCS, the designer must compromises with the behavior of the network, e.g. delay, to build good control systems. Delay are the most common challenges in design NCS which can be divided into critical delay and non-critical delay according to the behavior of plant. Systems with critical delay can be found, like in industrial automation where the automation process must be working accurately, in tele-surgery operation, etc. Example of non-critical delay systems are SCADA.

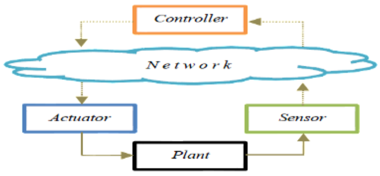

Figure 1. NCS Single Level Controller

In this research, the several steps that had been taken in designing and implementing NCS are shown in Fig. 2. Performance specification stated in this research focuses on the delay that affect the response of the control systems. Component of NCS consist of the platform controller, sensor, actuator, and network. The sensor and actuator used in this research are provided together with the plant.

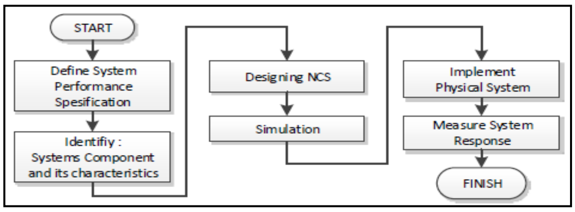

Figure 2. Steps conducted to implement NCS

The plant, as shown in Fig. 3, are miniature of industrial automation which is consist of conveyor, solenoid pneumatic (SO-1), and proximity sensor (PS-1 and PS-2). There are 2 kind of boxes run in the conveyor which has different level, 1 cm and 2 cm. The goal is to remove the box with 2 cm height from the conveyor by using the solenoid pneumatic SO-1 after the box has been detected by sensor PS-2.

Figure 3. Plant using miniature of industrial automation

The most significant causes that affect the overall systems performance coming from the platform controller, as shown in [4]. The platform controller in this paper use PLC OMRON CJ1M-CPU11-ETN. Usually the latest sensor and actuator, e.g. smart sensor and smart actuator, can be connected directly to the network using Ethernet module that come along in their packages. But the sensor and actuator in the miniature of industrial automation above are not provided with capability to be connected directly to the network, so we install another PLC OMRON just only as the interface between sensor and actuator to the network. We can use the remote I/O as another option becoming the interface between sensor-actuator and network. Also PLC OMRON has been chosen as the interface because the network protocol uses OMRON FINS/TCP based TCP/IP in order to make it easier to be implemented. FINS is the protocol that sends messages between PLCs on any of various OMRON Factory Automation network.

To design the control strategy in NCS, select the suitable architecture then define the data flow information. Control signal consist of signal sensor-to-controller and controller-to-actuator. Because of the control signal in NCS travelled across the network, the term 'signal' can be replace by 'data' [4]. The NCS designed and its data flow from each sources and destinations shown in Fig. 4 taking the NCS Single Level Controller as the architecture. This data flow is important if we choose the programmable controller as the platform controller because later we have to design the structure of its memory allocation before coding the program and to avoid the data confuse if the controller must handling many plants, sensors, and actuators moreover if the structure is distributed. More efficient the allocation memory will decrease the computational delay of programmable controller thus will improve the systems response. The NCS data flow designed as follows. First, the signal sensor-to-controller with sources from data sensor PS-2 (S-C_1). Second and third, the signal controller-to-actuator from the controller to actuator which has 2 destinations, controlling the conveyor (C-A_1) and controlling the solenoid pneumatic SO-1 (C-A_2). The PLC 1 and PLC 2 has been added just as interface between sensor and actuator to the network because sensor and actuator that used does not have ability to connect direct to the network. Surely, they can be replaced by other I/O Module which provide connection support to the network and using the same network protocol.

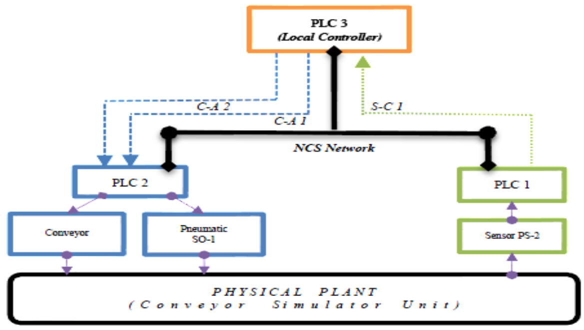

Figure 4. NCS data flow and architecture with single level controller

III. IMPLEMENTATION OF NCS

After NCS has been designed, next step is build the network. The network as shown if Fig. 5 lies on 192.168.250.0. All the programmable controller, which has the unique IP address and subnet, are connected to the industrial switching hub. In this paper, the size of network is Local Area Network (LAN) and possibly can be upgraded to Wide Area Network (WAN) by build more network and linked each network with router. For monitoring and tracing the data, some computer has been added to the network also the server for traffic and some client to send the traffic to server. In this NCS network, the switch has been linked with wireless hub so the client for traffic can be connected using wireless LAN. Before implemented to physical systems, it's better to have done some simulation i.e. simulation of control algorithm program and network simulation. Control algorithm program on programmable controller has been simulated using CX-Simulator to make sure the data flow runs correctly and network simulation also has been conducted using Cisco Packet Tracer.

The tasks doing by the controller can be breakdown into 3 parts, the process for receiving data from sensor, processing the control algorithm, and process sending data to actuator. Because the controller is programmable controller, so the programming language using standard ladder diagram (IEC-6113). Process receiving data sensor and sending to actuator are composed by using FINS Command. FINS Command is the command form to do many operation in FINS network service e.g. write and read the memory of programmable controller, sending message, alarm warning, etc. All of FINS Command will pass the communications port at sources and destinations nodes. OMRON's programmable controller has 8 communications port. This port will be 'open' or has value '1' when the condition is idle and will be 'closed' or has value '0' when it used to sending command, busy or having trouble. When sources node sending any of FINS command, then the destinations node will reply the response code. If the command arrived successfully, the response code will be '0000' hex, in other hand the response code will has special value in hex if the command failed to reach the destinations node. Every value of response code can be used to troubleshoot why the command failed to be sent. Detail about response code available on [15].

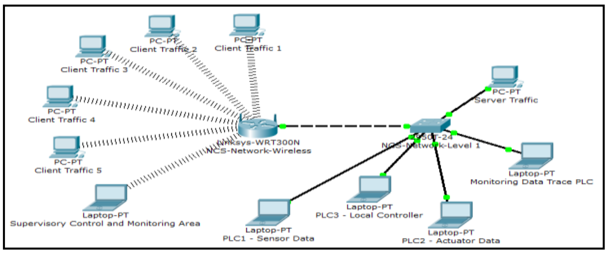

Figure 5. NCS Network with client-server for traffic

IV. MEASUREMENT AND ANALYSIS

A. *Theoritical delay in control network*

In NCS, there are 2 kinds of delay, as described in [8, 11], i.e. device delay and network delay. Delay in each control signal, i.e. delay sensor-to-controller (τ_{sc}) and delay controller-to-actuator (τ_{ca}), there are components of device delay and network delay. The ilustration of delay in NCS shown in Fig. 6.

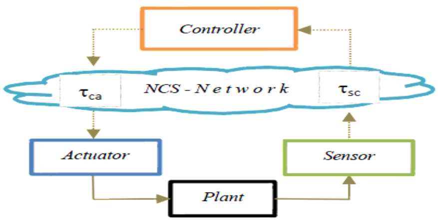

Figure 6. NCS with delay components

Device delay consists of delay at source node and delay at destination node. At the source node, device delay can be breakdown into delay pre-processing (τ_{pre}) and delay waiting (τ_{wait}) also at destination node, the device delay only has delay post-processing (τ_{post}). Delay pre-processing is time required by device at source node to take the data from the systems environment and encode it into suitable format according to the protocol's format data. Delay waiting is waiting time when data are queuing i.e. usually occur when the recent data is still in process to send and in the same time the next data is ready to send. But, in this NCS data flow, the delay waiting can be assumed zero ($\tau_{wait} = 0$) because the program has been set to cancel the sending process if the queuing occur. Also as mention in [8], it is quite hard to measure the delay waiting because the network is uncertainties. Delay post-processing is time required by device at destination node to decode the received data and change it into the suitable format data at the systems environment. So it can be described by:

$$\tau_{device} = \tau_{pre} + \tau_{wait} + \tau_{post} \qquad (1)$$

Because $\tau_{wait} = 0$ then:

$$\tau_{device} = \tau_{pre} + \tau_{post} \qquad (2)$$

τ_{pre} is delay at source node and τ_{post} is delay at destination node, thus:

$$\tau_{pre} = \tau_{source\ node} \qquad (3)$$

$$\tau_{post} = \tau_{destination\ node} \qquad (4)$$

According to information explained in [15], for programmable controller, we have:

$$\tau_{device} = \text{CPU BusUnitService Cycle} + \text{CPU BusUnitService Processing Time}$$

$$= \text{CPU Cycle Time} + (4\% \ \text{x CPU Cycle Time}) \qquad (3)$$

Noted that (3) only for normal processing mode. Cycle time is time that needed by CPU unit of programmable controller to do some tasks, according to [13], as follows. First, Overseeing Processing tasks to check the I/O bus, user program memory, check for battery errors and refreshes the clock. Second, Program Execution tasks to execute the user program and calculate the total time taken for all instructions. Third, Cycle Time Calculation tasks to waits for specified cycle time to elapse when the fixed cycle time has been set for the programmable controller, but in this NCS, we didn't set the fixed cycle time for programmable controller so time for this third tasks will be '0'. Fourth, I/O Refreshing, is process to refresh all the Unit that has been installed in the programmable controller. The last tasks, Peripheral Servicing, is time needed by CPU unit to service the peripheral unit like service communications port, etc. Thus, the cycle time is calculate by summing the time needed by every tasks mention above. So, for calculating the cycle time, we have:

$$\text{Cycle time} = \text{Overseeing Processing Time} + \text{Program Execution Time} + \text{I/O Refreshing Time} + \text{Peripheral Servicing Time} \qquad (4)$$

Network delay is total transmission time of a frame messages sent from source node until received at destination node (τ_{frame}) also include the propagation delay of the network (τ_{prop}). From reference [15], we can conclude that the network delay can be draws as follows.

$$\tau_{network} = \tau_{frame} + \tau_{prop} \qquad (5)$$

where

$$\tau_{frame} = \tau_{transmission\ processing} + \tau_{reception\ processing} \quad (6)$$

$$\tau_{transmission\ processing} = (\text{number of words sent x } 0.002) + 0.550 \text{ ms} \quad (7)$$

$$\tau_{reception\ processing} = (\text{number of words sent x } 0.003) + 0.704 \text{ ms} \quad (8)$$

For propagation delay, we can described by:

$$\tau_{transmission\ delay} = \tau_{prop}$$
$$= (\text{number of words sent x } 0.0013) + 0.0118 \text{ ms} \quad (9)$$

Now we are going to calculate the delay for each of control signal. The control signal were generated by executing the FINS Command issued by the controller. Since the typical of that control signal are divided into time-driven and event-driven, as described in [4], so that different FINS Command was established. Signal sensor-to-controller (S-C_1) were generate using instruction RECV, since it was set to time-driven control signal, the controller took data sensor every periodic of time, i.e. the instructions RECV generated every cycle time of CPU Unit. Signal controller-to-actuator (C-A_1) was the same like S-C_1. In other hand, control signal (C-A_2) was set to event-driven because the controller only issued the FINS Command SEND if the box has been detected by sensor PS-2. Instruction SEND and RECV has different characteristics and probably the RECV instruction will produce more delay than SEND. From reference [15] and the equations above, we can conclude the delay from each control signal into timing diagram of each instructions SEND and RECV shown in Fig. 7 and Fig. 8.

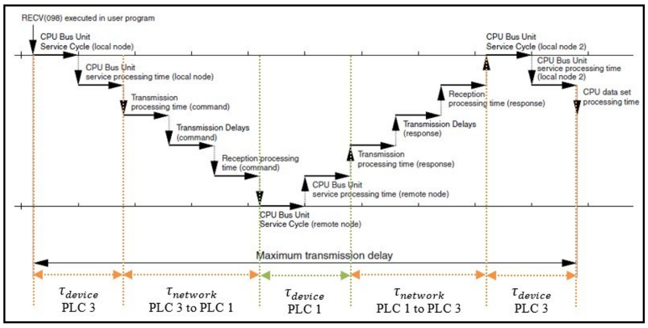

Figure 7. Time diagram for delay sensor-to-controller

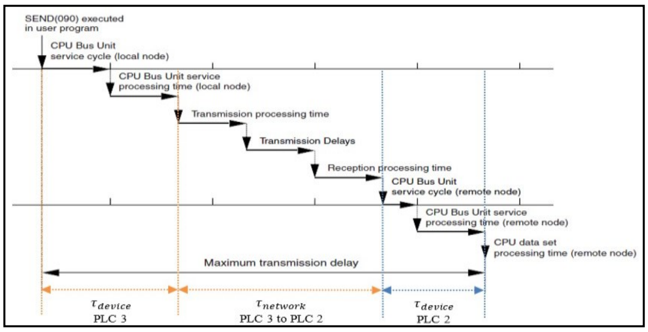

Figure 8. Time diagram for delay controller-to-actuator

Thus, we have:

$$\tau_{sc} = \tau_{device\ PLC3} + \tau_{network\ PLC3\text{-}to\text{-}PLC1} + \tau_{device\ PLC1} + \tau_{network\ PLC1\text{-}to\text{-}PLC3} + \tau_{device\ PLC1} \quad (10)$$

$$\tau_{ca} = \tau_{device\ PLC3} + \tau_{network\ PLC3\text{-}to\text{-}PLC2} + \tau_{device\ PLC2} \quad (11)$$

By calculating the delay using (10) and (11) with the frame message size maximum is 2000 bytes, so we have the theoritical delay are 0,00744 seconds for delay sensor-to-controller and 0,0042 seconds for delay controller-to-actuator.

B. *Experiment Results and Analysis*

In measuring the systems response, i.e. delay, several scenarios/cases has been conducted. The delay which has been measured in this experiment consists of delay sensor-to-controller (S-C) and delay controller-to-actuator (Conveyor) (C-A1) and delay controller-to-actuator (Pneumatic) (C-A2). Also the control execution accuracy, i.e. removing the box from the conveyor, has been added to the experiment in order to find out the impact of delay that occur in NCS related to the control performances. There are 5 rating given to each condition of execution in detail shown in Table 1. Case 0 is the ideal condition using control without network bor just to compare the accuracy control between control using network (NCS) and traditional control (without network). The result of Case 0 shown in Fig 9. Case 1 is scenario using NCS without added any traffic into the network. Case 2 is scenario using NCS with added traffic generated by 50 clients sending the messages 100000 bytes per seconds to the server and server echoed the messages back to each of the clients. Case 3 is similar to Case 2 but the total of clients has been increased to 200 clients. The results of measurement in Case 1, Case 2, Case 3 are shown as follows in Fig 10-19.

TABLE I. CONTROL EXECUTION ACCURACY RATING

Accuracy	Passed Green Line	Passed Orange Line	Bounces the Wall	Pushed-away by Pneumatic
100 % “Perfect”	Yes	Yes	No	Yes
75 % “Fair”	Yes	Yes	Yes	Yes
50 % “Poor”	No	Yes	Yes	Yes
25 % “Bad”	No	No	Yes	Yes
0 % “Very Bad”	No	No	No	No

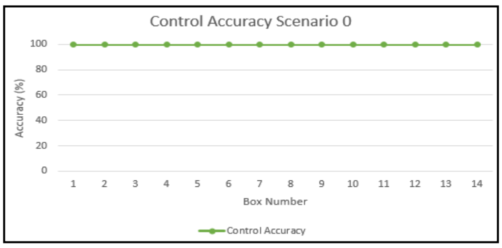

Figure 9. Control execution accuracy in Scenario/Case 0

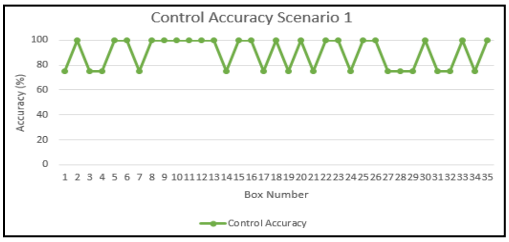

Figure 10. Control execution accuracy in Scenario/Case 1

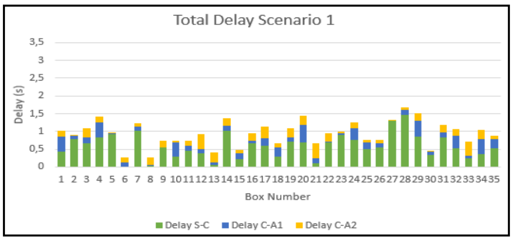

Figure 11. Total delay in Scenario/Case 1

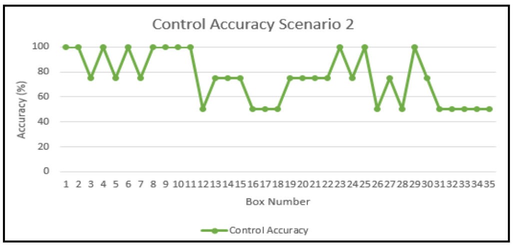

Figure 12. Control execution accuracy in Scenario/Case 2

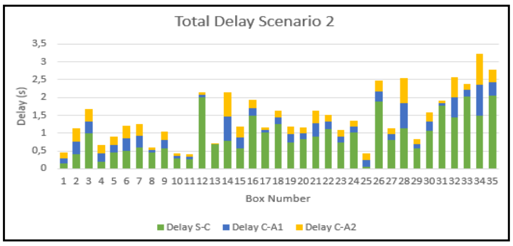

Figure 13. Total delay in Scenario/Case 2

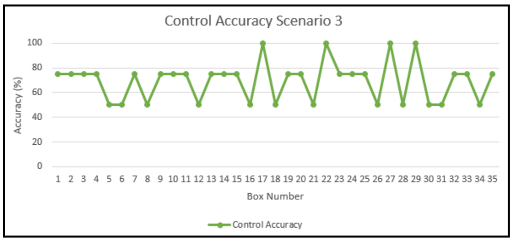

Figure 14. Control execution accuracy in Scenario/Case 3

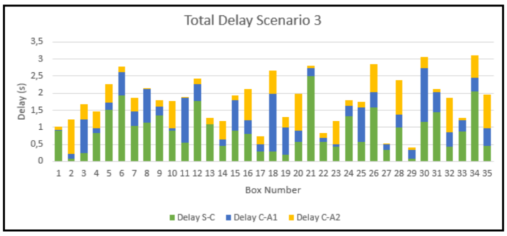

Figure 15. Total delay in Scenario/Case 3

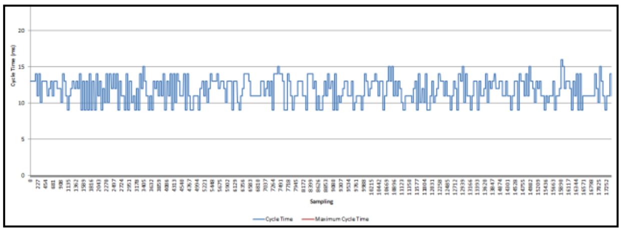

Figure 16. Cycle Time of Programmable Controller in Scenario/Case 0

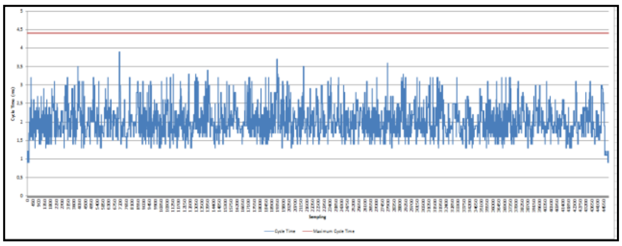

Figure 17. Cycle Time of Programmable Controller in Scenario/Case 1

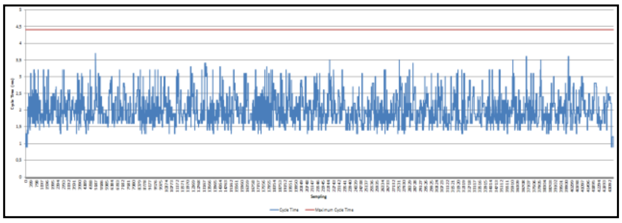

Figure 18. Cycle Time of Programmable Controller in Scenario/Case 2

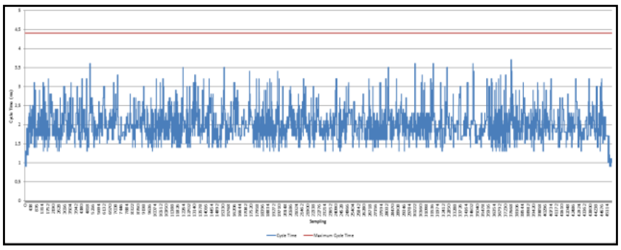

Figure 19. Cycle Time of Programmable Controller in Scenario/Case 3

The delay measurement conducted by monitoring the status of programmable controller's communications port while capturing the cycle time data from programmable controller's memory. The measurement of control execution accuracy took after observation of the process of removing the box from the conveyor. Sampling resolution to retrieve data from memory is 10 ms which is the smallest settings.

From the results of the experiment above, it is clear to see that when using the network into the control systems, the control accuracy will decrease from 100% in Case 0 using traditional control without network to 89.825% in Case 1 when using NCS with dedicated network or without any other traffic except the control systems itself. This control accuracy become 75% in Case 2 and descend to 69,285% in Case 3 or maximum traffic condition that conducted. The delay in each types of control signal will vary but the control execution depends on the total delay that occur. Because the industrial automation is one of the example of critical delay, thus the total delay that occur should not more than critical delay limit. The critical delay is the delay limit when the systems can maintain and meet the specification performances. So that control performances must always "Perfect" so that process workflow and the industrial process will runs well. From the experiment results in measuring the control accuracy and the total delay in Case 1, Case 2, and Case 3 shows that control accuracy 100% or "Perfect" happen if the total delay less than 1 s. The control accuracy 75% or "Fair" occur if the total delay around 1s until 1.5 s. Also the control accuracy will decrease become 50% or "Poor" when the total delay is more than 1.5 s. From the experiment, the critical delay limit is less than 1 s (critical delay < 1 s).

Comparing the results of cycle time measurement between Case 0 which is traditional control scheme without networks and Case 1, Case, 2, Case 3 which is NCS shows that the average value of cycle time increase from 1.179 ms to near 2 ms. This happen because the programmable controller executed more instruction and more complex in NCS scheme. But the cycle time value from experiment in Case 1, Case 2, and Case 3 are close to near 2 ms. It means that no matter traffic condition in the network, the performances and cycle time of programmable controller still the same. The average value of cycle time in each cases are 1.179 ms for Case 0; 2.014 ms for Case 1; 2.0177 ms for Case 2; and 2.0274 for Case 3. The maximum cycle time is about near 4.4 ms.

V. CONCLUSION AND FUTURE WORK

The paper presented the practical approach and simple steps in implementing the NCS using programmable controller based Ethernet network. The network used in the experiment is vary from dedicated network to the high traffic network. Also several experiment has been tested to see the impact of the network in control systems. The results from experiment may differ with the theoretical calculation because the network behavior are really uncertainty. Delay in the network can decrease the control execution accuracy from control accuracy 100% or "Perfect" descend to 69,285% in worst case conducted in this research. The results of the comparison between the total delay control accuracy in Case 1, Case 2, and Case 3 obtained that control accuracy 100% "Perfect" occurs when the total delay less than 1 s. While 75% control accuracy or "Fair" is reached when the total delay between 1 s to 1.5 s. As if the total delay more than 1.5 s the level of control accuracy will be reduced to categories control accuracy 50% or "Poor". In other hand, the traffic condition does not have any impact related to the cycle time of programmable controller.

Further research is needed with other methods or software that can capture the data on the programmable controller but with 1 ms sampling resolution or smaller so that every event that occurs can be see and predicted. Also better control strategy are needed to maintain the control performances along with to deal with the network behavior. Another option, reducing the traffic for example, is also one of the way to maintain the control performances.

REFERENCES

[1] R. M. Murray, K. J. Åström, S. P. Boyd, R. W. Brockett, and G. Stein. "Control in an Information Rich World". IEEE Control Systems Magazine. vol. 23, No.2, pp. 20-33, April 2003.

[2] R. M. Murray. (2006). *an Introduction to Networked Control Systems.* California, USA: California Institute of Technology. Available on: http://www.cds.caltech.edu/~murray/wiki/images/3/34/Ncsbook-introduction.pdf . [November 20, 2011].

[3] Zhang, Wei, M. S. Branicky, and S. M. Phillips. "Stability of Networked Control Systems". IEEE Control Systems Magazine. IEEE Control System Society. pp. 84-99, February 2001.

[4] Yang, Tai Cheng. "Networked Control Systems: Current Research and Future". Panel Presentation for the 10th WSEAS International Conference, Athens, Greek. July 2006.

[5] Sutarto, Herman Y. "Networked Control Systems: Modeling and Control of Road Traffic". The 2012 IEEE Conference on Control, Systems & Industrial Informatics (ICCSII 2012), Bandung, Indonesia. September 2012.

[6] Antsaklis, Panos and John Baillieul. "Special Issue on Technology of Networked Control Systems". Proceedings of The IEEE. Vol. 95, No.1, pp. 5-8, January 2007.

[7] Burns, Roland S. Advanced Control Engineering. Oxford, UK: Butterworth-Heinemann, 2001.

[8] Lian, F.L., J.R. Moyne, and D.M. Tilbury. "Network Protocols for Networked Control Systems". Handbook of Networked and Embedded Control Systems. New York, USA: Birkhäuser Boston, 2005.

[9] Ramadhan, Aris, M. Ary Murti, and Leanna V.Y. "Design and Implementation of Networked Control Systems (NCS) Using Programmable Logic Controller (PLC) OMRON Based Ethernet Network Communications". Final Project, Faculty of Electrical and Communication, Telkom Institute of Technoloy (IT Telkom), Bandung, Indonesia, 2013.

[10] Hespanha, João P, Payam Naghshtabrizi, and Yonggang Xu. "A Survey of Recent Results in Networked Control Systems". Proceedings of The IEEE. vol. 95, No.1, pp. 138-162, January 2007.

[11] Yang, Liman, Yunhua Li, and Guilin Yang. "Analysis of Delay and Traffic Load in Networked Control Systems". Proceedings of The 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Monterey, California, USA, pp. 1425-1430, July 2005.

[12] OMRON. W342-E1-15, SYSMAC CS/CJ/CP Series, SYSMAC NSJ ONE Series, Communications Commands Reference Manual. Kyoto: OMRON Corporation, 2010.

[13] OMRON. W393-E1-14, SYSMAC CJ Series Programmable Controllers, Operation Manual. Kyoto: OMRON Corporation, 2008.

[14] OMRON. W394-E1-14, SYSMAC CS/CJ Series, SYSMAC ONE NSJ Series Programmable Controllers, Programming Manual. Kyoto: OMRON Corporation, 2009.

[15] OMRON. W420-E1-06, SYSMAC CS and CJ Series, CS1W-ETN21 (100Base-Tx), CJ1W-ETN21 (100Base-Tx) Ethernet Units, Construction of Networks Operation Manual. Kyoto: OMRON Corporation, 2009.

[16] OMRON. W474-E2-08A, SYSMAC CS Series, SYSMAC CJ Series, SYSMAC ONE NSJ Series Programmable Controllers, Instruction Reference Manual. Kyoto: OMRON Corporation, 2010.

[17] Johansson, K.H., M. Törngren, and L. Nielsen. "Vehicle Applications of Controller Area Network". Handbook of Networked and Embedded Control Systems. New York, USA: Birkhäuser Boston, 2005.