

# Data Communications Network for Real-Time Industrial Control Systems

Nkolika Ogechukwu Nwazor<sup>1\*</sup>, Eliezar Elisha Audu<sup>2</sup>

<sup>1</sup>Department of Electrical/Electronic Engineering, University of Port Harcourt, Nigeria

<sup>2</sup>Center for Information and Telecommunication Engineering, University of Port Harcourt, Nigeria.



**ABSTRACT:** The advancements in network technologies and the evolution of the Internet of Things (IoT) have made supporting industrial control systems over probabilistic data networks promising. However, control systems' communication over the traditional data networks is faced with problems of instability in feedback control and poor quality of performance due to time-varying data propagation delay. This paper presents two approaches that can enable real-time industrial control over non-deterministic computer networks allowing control system designers to take advantage of the existing communication infrastructures. The first approach is based on system-level interaction over two wires network called the collaboration network. The second approach is based on the implementation of the virtual local area network (VLAN). This method allows real-time control of industrial equipment or systems over IP-based networks while other computers are connected. Nodes providing real-time control services have the same PortID on the VLAN switch. This approach minimizes data traffic and reduces time-varying delay in system control over IP networks. The first approach was modeled and simulated using Proteus ISIS software. Two PIC16F877A microcontrollers were used to represent two nodes. CISCO packet tracer was used in the second approach to model and simulate IP-based control system communications over the traditional data network. Results indicate that the use of a two-wire collaborative network approach to a real-time control system is effective but requires an additional network alongside the main data traffic channel. VLAN, therefore, presents a more flexible approach that relies on the same infrastructures.

**KEYWORDS:** Real-time systems, Control systems, Local area network, Internet protocol, Data communication, CISCO.

[Received Aug. 18, 2021; Revised Nov. 25, 2021; Accepted Jan. 29, 2022]

Print ISSN: 0189-9546 | Online ISSN: 2437-2110

## I. INTRODUCTION

The innovative transformation from classical to network-based control systems (NCSs) has presented new possibilities in system integration and created new possibilities that have been applied in many spheres. NCSs are widely used in transportation systems, industrial production systems, manufacturing, electrical transmission networks, and automobile and aircraft systems (Gupta and Chow, 2008; Zampieri, 2008). They offer many benefits such as ease of maintenance, greater flexibility, low installation costs, and the ability to scale the nodes (Cloosterman et al., 2009; Heemels et al., 2010). In NCS implementation, a controller may depend on more than one sensor node for decision making, which makes speed and latency important factors in NCS design. Non-linear time-varying delay, packet dropout, and data congestions negatively affect nodes in NCS causing poor control performance, instability, and loss of quality control (Cloosterman et al., 2009; Heemels et al., 2010; Jetto and Orsini, 2009; Wang and Lemmon, 2010; Pan et al., 2011).

In real-time control systems, network information is time-constrained, and failure to meet deadlines can impact negatively on the system performance and quality of service. Different methods have been developed to deal with the

problem of time-varying delay and packet dropout due to distance, routing, and traffic congestion. From an analysis and control system point of view, the Lyapunov function has been used to address the problem of nonlinear quality of service (QoS) (Peng et al., 2008). In (Cloosterman et al., 2009), a new approach was presented based on the discrete-time representation of the NCS using the Lyapunov-based stability criterion expressed in terms of linear matrix inequalities (LMI). In the same fashion, perturbed frozen time (PFT) and point-wise stability approaches have been used to investigate the problem of stability in NCS with the assumption of a linear time-varying system (Jetto and Orsini, 2009). Besides stability analysis of NCS, methods have been devised based on the already established stability analysis to minimize the effect of time-varying delay and packet dropout on networked control systems.

The solution to the problem of stochastic delay (or dead time) was first offered by Smith in 1957 by eliminating the time-delay variable from the control loop characteristic equation (Smith, 1957; Shokri et al., 2010). The Smith predictor method of compensation and robust control stability theory has been used to improve the quality of performance (QoP) of NCS on shared communication networks (Vatanski et al., 2006). It is largely applied to either structurally or

\*Corresponding author: nkolika.nwazor@uniport.edu.ng

parametrically optimize the performance of a classical control system over networks with unpredictable time delay. Similarly, a Smith compensator based on the internal control module and dynamic matrix controller was used to deal with the problem of stochastic network delay in real-time and internet-based NCS (O'Dwyer, 2005; Yang et al., 2005). One of the benefits of this method is that it requires minimal knowledge of plants under control (Massimiliano, 2003). The development of the Smith Predictor model opened a Pandora box in developing a technique that can minimize time delay. However, the model mismatch can result in closed-loop instability and degrade the quality of control of the networked control system (Shokri et al., 2010).

Fuzzy logic has been successfully applied in NCS to provide tuning parameters and non-linear mapping for Proportional-Integral-Derivative (PID)-based controllers. The traditional PID controllers tuning technique such as the Ziegler-Nichol method provides unsatisfactory performance of NCS where the time delay exceeds the critical value, thus resulting in loss of control efficiency (Vardhan and Kumar, 2011; Hang, 2009). The PID-based fuzzy controller allows both the input variables and the control action to be defined in terms of linguistic rules (Al-Odienat and Al-Lawana, 2008). Gaddam and Akula (2006) proposed the use of the fuzzy set weighted controller in dealing with the problem of stochastic time delay in a control network.

In this paper, we present two methods for implementing industrial control systems over IP-based data communication networks. The first approach is based on an intelligent two-wire communication network called the collaborative network. The second approach relies on VLAN implementation on switched to create group collision domains within the network.

## II. LITERATURE REVIEW

Network control systems play a crucial role in industries and manufacturing where processes are automated and controlled easily and efficiently (Walsh et al, 2002; Zhao et al., 2009). The network supports different vendors and requires stringent time constraints. Over the years, different specifications have been developed by different control systems service providers to ensure reliability, ease of troubleshooting, scalability, and efficiency (Zhao et al., 2009).

### A. Modbus and Fieldbus (Profibus)

Modbus is one of the oldest and most widely used industrial communication protocols for instrumentation and automation. It was designed to connect programmable logic controllers (PLC) and smart equipment using the master-slave concept (Powel, 2013). Modbus is a serial data communication protocol for interconnecting and integrating devices in industrial process control and monitoring systems (Zhang et al., 2011). The network is structured into Master-Slave architecture, where the device requesting for data from the network is termed the Modbus master, and devices replying to requests are called the Modbus slave. Transaction of information in the Modbus network is regarded as stateless, and a Master controller is required to

verify the completeness of the information (Swales, 1999). Devices connected using Modbus protocols are equipped with communications capability and processing ability to effectively take advantage of collective intelligence by decoding and executing commands from received data frames. Only the Modbus master can send multicast messages to the slave devices. The robustness and simplicity of the Modbus communication protocol have made the network one of the preferred industrial communication networks.

Similar to Modbus industrial communication protocol is Fieldbus which is an industrial communication protocol that was developed in the 1990s as an alternative solution to Modbus (Powel, 2013). It is specified under IEC 61158 with several variants such as the French Factory Instrumentation Protocol (FIP) and the German Process Fieldbus (PROFIBUS) (Thomesse, 2005). Fieldbus is a variety of solutions and techniques or a collection of protocols that allows different vendors and process equipment manufacturers to implement their own real-time industrial control system network (Choi and Kim, 2008; Thomesse, 2005). The development of Fieldbus technologies allow industrial devices to communicate over a shared communication network and aggregate information in real-time. Several topologies such as ring, bus, tree, branch and star can be implemented using Fieldbus protocols (Choi and Kim, 2008). To guarantee real-time communications, a timed token protocol is used as a prescription for the timely delivery of information between two or more nodes. The large number of sensors functioning as nodes in Fieldbus architecture can be monitored from a central computer.

Modbus and Fieldbus are completely different communication protocols and they utilize the same master-slave concept to manage and control industrial processes (Zhang et al., 2011). Modbus is easy to implement and deploy than Profibus but creates variation in applications that requires multiple vendors (Zhang et al., 2011; Powel, 2013). However, Profibus is a more robust protocol suitable for multi-vendor applications and has detailed diagnostics (Powel, 2013). The lack of compatibility between Modbus and Profibus protocols has been a subject of research activities in network control systems and data communications. Gateway for connecting Profibus and Modbus industrial communication protocols was proposed by Zhang et al (2011).

### B. Controller Area Network

The development of a controller area network (CAN) by BOSCH as a multi-master, multicast-based advanced serial communication protocol is aimed at managing communications in vehicles and automobiles in real-time (Di Natale, 2008; Johansson et al., 2005; Corrigan, 2002). CAN is a broadcast bus that can operate at a network speed of 1 Mbps designed to meet the requirement of complexity in automobile electronics technology (Tindell et al., 1994; Tindell et al., 1995). An essential aspect of CAN is the use of an identifier that assigns a priority to a message and enables receivers to filter (Tindell et al., 1994; Tindell et al., 1995). CAN protocol is widely used in industrial automation and other areas of networked embedded applications in diverse products (Johansson et al, 2005). Electro-mechanical devices can be

networked and controlled using CAN serial communications protocol as shown in Figure 1.

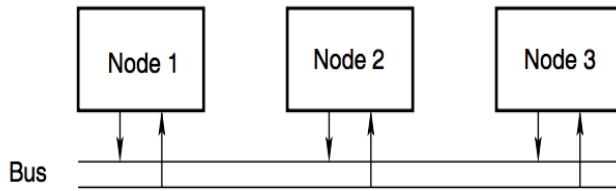


Figure 1: CAN bus topology for connecting devices as nodes (Johansson et al., 2005).

This communication network protocol was introduced to deal with the problem of cost and scalability from the growing point-to-point wiring of electronic modules in vehicles (Zhen et al., 2017; Lokman et al., 2019). Only the content and priority of messages are broadcast on the network. Unlike Modbus, communications in CAN are by short messages and no master is required to arbitrate data delivery from one node to another (Corrigan, 2002). Nodes can be added or removed from the network without affecting the topology of the network. These advantages make CAN a protocol of choice in device integration and systems management in the automobile and aerospace industries.

The CAN communication architecture utilizes the data link and physical layers, which conform to the standard open system interconnection (OSI) layers as shown in Figure 2.

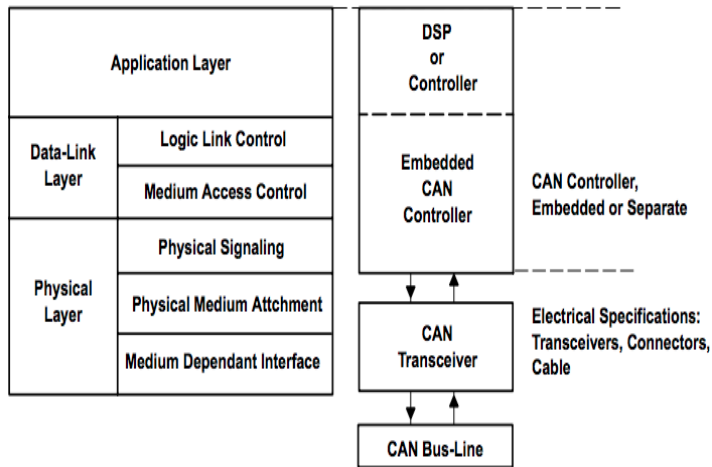


Figure 2: CAN ISO 11898 standard architecture (Corrigan, 2002).

Devices over CAN protocol access shared medium by sensing the channel to avoid collision using carrier sense multiple access (CSMA) with collision detection. One of the characteristics of CAN communication protocol is the deterministic resolution of the contention on its network (Di Natale, 2008). Using this protocol, several control systems can be connected to the serial bus system to share information.

### C. IoT and Data Communications

Advances in sensor technologies, microcontroller-based open-source development board communication networks and smart devices have led to the emergence of the Internet of Things (IoT), which connects physical intelligent objects or computing systems through some medium (Ojo et al., 2018; Jabbar et al., 2018; Niranja et al., 2017). It expanded the frontier of communication technology and redefined how devices interact over the internet by creating a digital highway of information that cohabits different devices and sensors over the same network. In automation and control, IoT provided a prescription to the problem of connectivity, coverage, and capacity of short-range communication technologies (Niranja et al., 2017). The global connectivity of IoT means that industrial automation and control systems are becoming more distributed and flexible than ever (Ojo et al., 2018; Jabbar et al., 2018). Today, industrial automation has evolved to be an integrated technology comprising distributed control systems, energy management systems, instrumentation and control, safety instrumented systems, and supervisory control and data acquisition that manage complex industrial and manufacturing processes (Fadali and Visioli, 2013; Mallikarjun et al., 2017).

Several platforms have been developed to support sensor nodes for application in IoTs for automation such as are Raspberry pi (Nehete and Bhide, 2017; Patil et al., 2017) and advanced RISC Machine (Kunal et al., 2020). The use of FPGA as a platform for early validation of platform design is increased in recent years (Wang and Jang, 2019; Cho et al., 2009). In industrial automation, a high-speed interface that can connect the computing system or IoTs to control external loads, motors, actuators, or sensors is gaining significant research attention. Interface systems allow low voltage, low current Arduino, or FPGA boards to control high current equipment or load to effect control actions on a process.

To achieve convergence of data networks and industrial automation of the future, a framework referred to as industrial revolution 4.0 has been developed which provides guidelines for integrating networks and cyber-physical systems (Rojko, 2017). This network of independent but interconnected smart sensors provides seamless interaction between the digital and the real world. Digital information and data are constantly being exchanged between the two worlds through internet protocol (Berger, 2014). According to Vuksanovic et al (2016), Industry 4.0 is a smart ecosystem where man and machines are equal partners, and networked intelligent systems provide connectivity for local decentralized information processing. Industry 4.0 combines artificial intelligence, data science, and pervasive digital transformation to deliver flexibility, efficient products, product customization, and real-time business intelligence (Huang et al., 2019; Berger, 2014 and Vukanovic et al., 2016).

### III. MATERIALS AND METHODS

In this section, we present in detail the materials and methods used to achieve the results. Proteus ISIS schematics were used to model and simulate the communications between nodes (microcontrollers). CISCO packet tracer was used to

model, simulate and validate IP-based communications using the virtual local area (VLAN) network.

A. Collaborative Network using Two-wire Communication

The EIA-485 network for prioritizing communication in real-time control is implemented in Proteus ISIS IDE. Two microcontrollers, PIC16F877A, acting as two transceivers are used to demonstrate how nodes can collaborate and prevent packet loss and ensure real-time control of systems over conventional data networks. The PIC16F877A microcontroller has an in-built ADC module and digital input/output (I/O) pins. The module converts the analog signal into a 10-bit digital representation. Figure 3 shows the pinout designation of PIC16F877A microcontrollers.

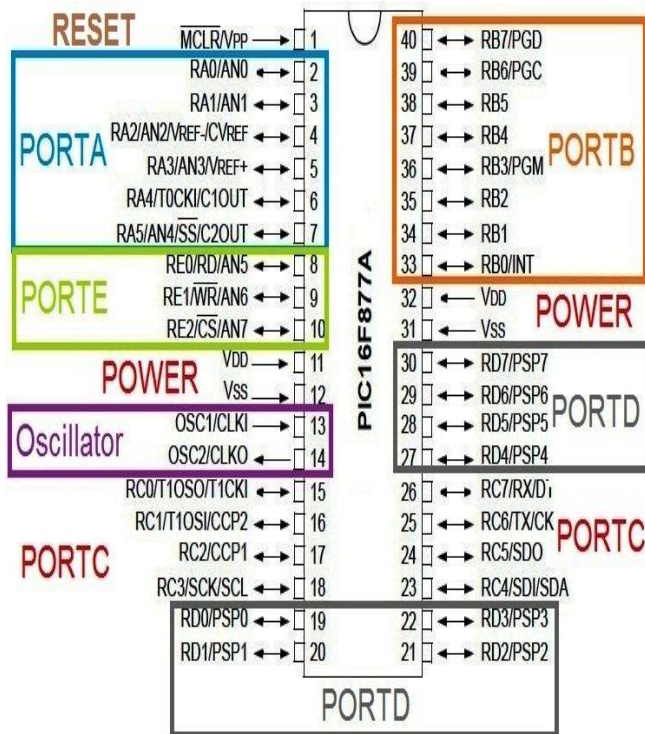


Figure 3: PIC16F877A pinout designation.

The 8-bit port B is configured to write and read data from the liquid crystal display (LCD). The reset, read/write control and LCD control of the LCD are connected to Port D5, Port D6, and Port D7 pins respectively. Pin 6 and 7 of port C are used to transmit and receive data on the collaborative network. Pin 3 and 4 of port D are used to provide control for the node (microcontroller) to send/receive and to sense/acquire control line respectively. Figure 4 shows the overall node block diagram with major components. The bi-directional communication over two wires communication network is achieved using sequential logic using 74HCT245 octal bus driver and 74HCT14 Schmitt trigger. The signal direction can be changed by toggling the signal at pin 1 of 74HCT245 from 0 to 1 or from 1 to 0 depending on the PIC16f877A microcontroller is sensing the bi-directional bus driver with logic inverter implemented the two wires multifunction communication network as shown in Figure 5. Port D, pin 3 is designated by CTRL1 which controls RX/TX bi-

directional bus driver, and Port D, pin 4 is designated by CTRL2 controls Request/Line busy (control line) functionality. Observed that both TX and RX lines of the PIC16F877A are connected to a single RX/TX line through their respective octal bus driver. The use of the 74HCT14 Schmitt trigger in the logic only allows the controller to transmit or receive at a time. On idle, the control line or Network Access Control Line (NACL) is pulled LOW by an internal circuit resistor. To access the network, a node senses the NACL by making the CTRL2 active LOW. If the NACL is LOW, then the collaborative network is idle and no device is transmitting. Once the Line Busy is LOW, the transmitting node acquires the network by setting CTRL2 pin to active HIGH. The ID of the transmitting node is sent to other nodes connected to the network. The transmitting node becomes the scheduled node and all other nodes are non-scheduling nodes.

The bus drivers (74HCT245) are enabled by an active LOW signal so that the node can read TX/RX into the network or sense the network for transmission. The logic inverter expands the ability of the control lines (CTRL1 and CTRL2) to switch between two functionally different bus drivers. Using a collaborative network, no network control node will attempt to communicate over the data network without authorization by the collaborative network. Each node on the collaborative network is assigned an ID that can be prioritized for critical control nodes.

Each node on the network is assigned an identification number (ID) for communicating over the two wired networks and an IP address for communicating over the data network. When no device is transmitting, the control line of the collaborative network is active LOW or at logic 0 and all devices on the network are listening devices. To send data, a node must acquire the collaborative network by pulling the control line HIGH. Once the control line is active HIGH, the ID of the transmitting node is read and placed on a priority table by other nodes on the network. This process is called the node discovery process. Access to a collaborative network is determined by which node is ready to transmit and has acquired the collaboration. This is quite slow.

Each node follows a simple algorithm to monitor and transmit the information as shown in the flowchart in Figure6. Once a node is connected to the network, it initializes and listens to the collaborative network for communications.

B. Virtual Local Area Network (VLAN)

This approach proposes the integration of a real-time control system into the traditional local area network data communication networks using VLAN. This technology allows two different subnetworks or subgroups of computers to be logically isolated within the same main network and improve the overall performance of the LAN. Logical partition of the network into several (small) broadcast domains within a layer-2 network, called the virtual local network (VLAN) reduces traffic and speeds up data transactions. Thus, solving the problem of a large collision domain created by a switch that consumes a significant

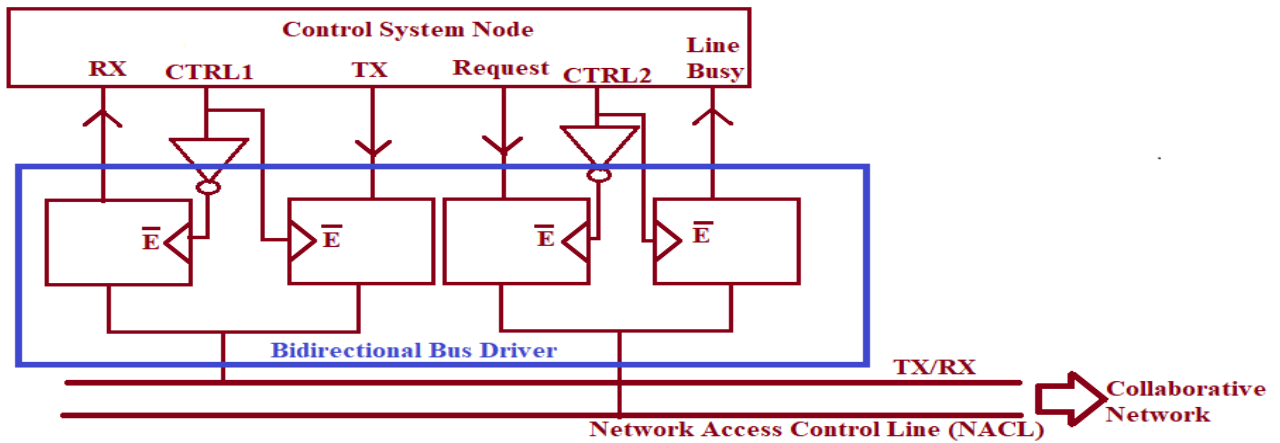


Figure 4: Simplified block diagram of PIC16F877A as control system node.

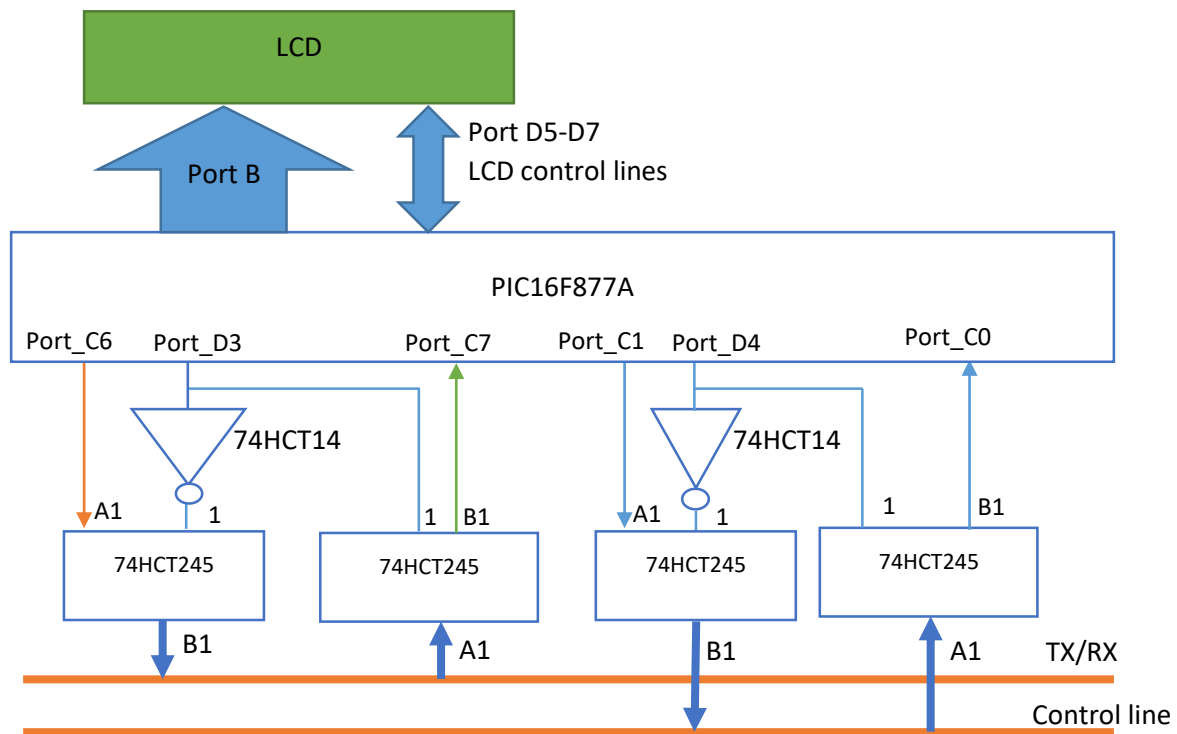


Figure 5: Design of nodes using PIC16F877A and two wires communication network.

amount of bandwidth and CPU cycle (Kalavathi et al., 2017). Links between switches are called the trunk, which carries different network traffic. In VLAN, data exchange between computers can occur if they belong to the same virtual subnet (Saenko and Kotenko, 2014). With data exchange being reduced to a group of devices, industrial control networks can be hosted on the same physical network as the information technology systems but logically separated using VLAN (Yamasaki et al., 2011; Sun et al., 2010; Garimella et al., 2007). Industrial control devices on Ethernet can be associated with the same VLAN or access port ID so that only IP-enabled sensors, actuators, and controllers can communicate on that VLAN. The packet sent over industrial control VLAN cannot be broadcasted to information

technology systems that are connected to a different VLAN subnet. In this design, there are three (3) remote terminal units (RTU) 1, 2, and 3. Each RMU has two NCSN linked through a switch as shown in Figure 7.

Each node is associated with an IP address and connected to an access port on the switch. The connection between switches is configured as a trunk to carry all the access ports traffic. Node 11 and Node 12 have the same VLAN ID, Node 21 and Node 22 have the same VLAN ID and Node 31 and Node 32 have the same VLAN ID. By this configuration, Only Node 11 in RTU-1 can communicate with Node 12 in RTU-3 since they have the same VLAN ID. Traffic from Node 11 will only be sent to the trunk port and block traffic to all other access ports with different VLAN IDs by SW1. The

connection between SW1 and SW2 carries VLAN 10, 20, and VLAN 30 traffic, and the connection between SW2 and SW3 carries VLAN 10, 20, and 30 traffic.

The naming of the node ID has the last byte of its IP address. Table 1 shows the list of Nodes ID, IP addresses, VLAN access port ID, and the remote terminal unit number. All Nodes are on the same subnets, 255, 255, 255, 0, and network, 192.168.1.xx.

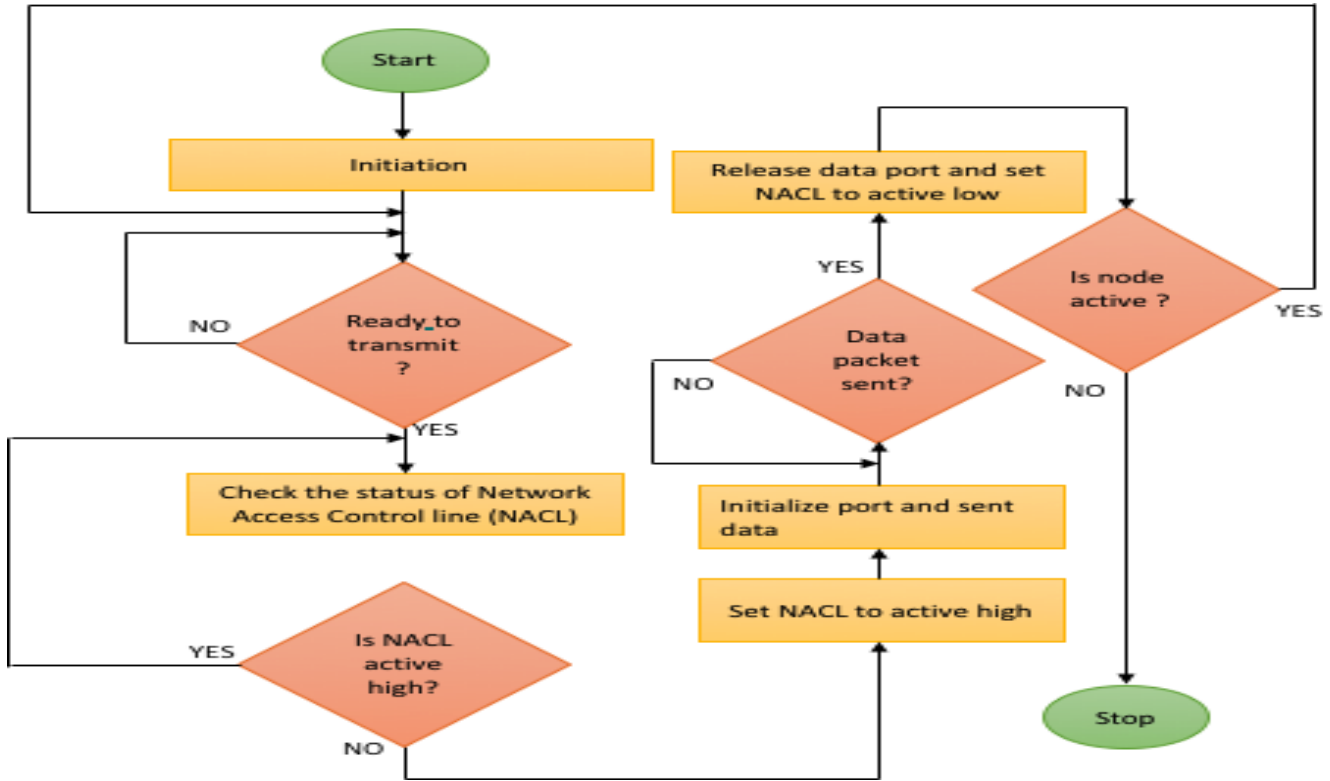


Figure 6: Flowchart of the microcontroller (node) communication over two wires network.

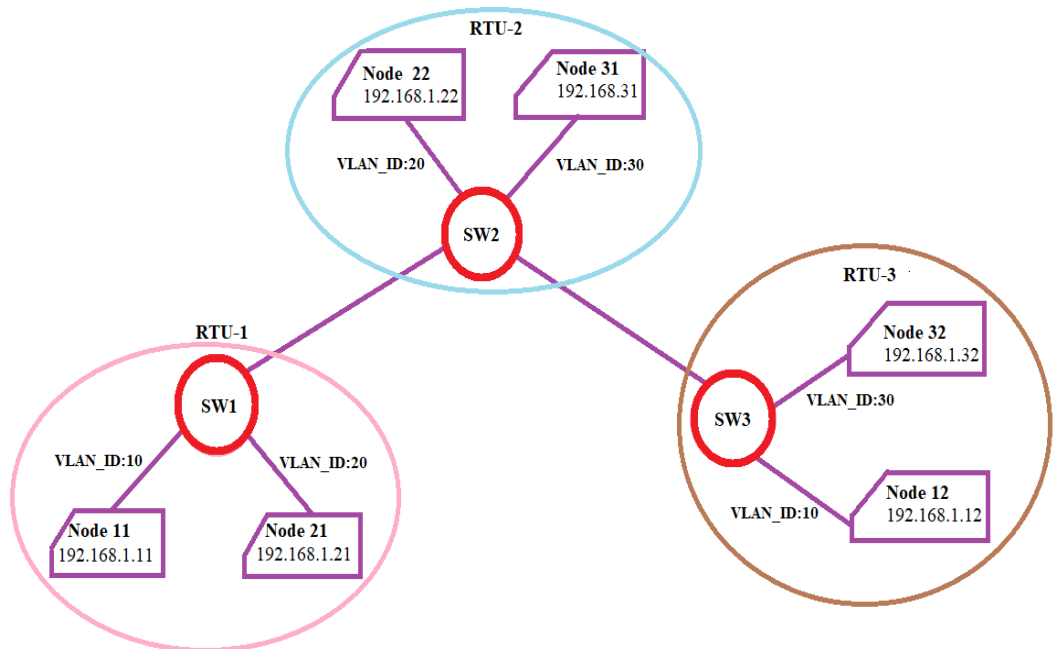


Figure 7: VLAN connected Remote Terminal Units for Network Control System Nodes.

#### IV. RESULTS AND DISCUSSION

In each of the two approaches to real-time control over data communication networks presented, we modelled and simulated prototypes, and the results are demonstrated in this section.

##### A. Two Wires Communication Network

Two microcontrollers are connected over two wires network to simulated scheduling through intelligent communication over a network. Figures 8 and 9 show how communication between two nodes is initiated and modulated over the two wires network (collaborative network).

The communication protocol was based on serial communications where one line was used for TX/RX data and the other line was used for Request/Line Busy (control line). Each PIC16F877A represented a control system node, which performs communications, information display, and control functions. When a node is in transmitting mode, all other nodes listen to TX/RX line. In Figure 8, when node 1 acquired the NACL and began the transmission, node 2 detected the communication and initialize itself by listening to both RX/TX line and NACL. The collaborative, two-wire network allows both node 1 and node 2 to communicate and prioritize access to the data communication network.

Data from Node 1 or any transmitting node is sent to the collaborative network by a broadcast method where all other nodes listen. Once a node successfully acquired NACL and is detected by other listening nodes, communication links are established otherwise the communication is timed out. Time out occurs when a node acquired access to the two-wire network but other nodes did not receive a message after T-seconds.

A transmitting node is known as the scheduled node and it sends information such as node ID over the two wires network as shown in Figure 9. Since each node has the knowledge of each other on the collaborative network, node 1 has an ID of 5, and its position in the node table is given by the function Node Position (Node\_ID) =Node Position. Through communication, each node knows the number of all the nodes on the collaborative network and their Node ID.

##### B. VLAN Implementation

To model and simulate how industrial control networks can be hosted and communicate in real-time over IP-based networks, we used three (3) switches namely, SW1\_RTU1, SW2\_RTU2, and SW3\_RTU3 in each of the three remote terminal units (RTU), RTU1, RTU2, and RTU3. These remote terminal units represent clusters of network control system nodes. Figure 10 shows the design of VLAN of three remote terminal units for IP-based networked control systems in hybrid topology using CISCO packet tracer. Each control system node is represented by a server, which in reality, is a node in the Internet of Things (IoT) with an associated logical address, the IP address for communicating over the network. The IP addresses and VLAN access ports ID of each RTU is presented in Table 1.

Each switch interface was configured to a specific VLAN\_ID. The default VLAN\_ID for all the interfaces on a switch is 1. Three VLAN groups were created on each of the three switches. The idea of creating VLAN groups is to create a different broadcasting domain and increase performance in the control system application. In SW1\_RTU1, interface fa0/2 and fa0/3 were configured to group 1 (VLAN 10) and group 2 (VLAN 20) respectively. By connecting these switch interfaces to different VLANs, traffic from group 1 cannot be received by group 2 and group 3. Interfaces fa0/2 and fa0/3 are called access ports and Figure 11 shows the VLAN configuration of SW1\_RTU1.

**Table 1: Nodes and associated IP addresses and VLAN access ports ID.**

Node ID	IP	VLAN ACCESS PORT ID	RTU
Node-11	192.168.1.11	10	RTU-1
Node-12	192.168.1.12	10	RTU-3
Node-21	192.168.1.21	20	RTU-1
Node-22	192.168.1.22	20	RTU-2
Node-31	192.168.1.31	30	RTU-2
Node-32	192.168.1.32	30	RTU-3

Similarly, fa03 and fa04 interfaces (ports) of switch SW2\_RTU2 were configured to group 2 (VLAN 20) and group 3 (VLAN 30) respectively. The SW3\_RTU3 switch interfaces are fa0/2 and fa0/3 on VLAN group 1 (VLAN 10) and VLAN group 3 (VLAN 30) were configured. All the switch interfaces have default VLAN port ID as 1. Traffic from the trunk port is only routed to the designated VLAN port ID. The connecting links between the switches are configured as trunks to carry VLAN 1, 10, 20, and 30. Interfaces Fa0/1 on SW1\_RTU1 and SW2\_RTU2 were configured as a trunk port while fa0/2 on SW2\_RTU2 and fa0/1 on SW3\_RTU3 were configured as trunk ports.

Despite the same subnet IP addresses, communication between nodes on the network is modulated by a switch VLAN port ID. Only nodes with the same VLAN ID or group can share information. Nodes under VLAN 10 cannot share traffic with VLAN 20 and VLAN 30. This technique (VLAN) eliminates unwanted traffic and minimizes latency in the use of a data network for instrumentation and control applications. Figure 11 shows the result of implementing VLAN at layer-2. Packets from 192.168.11 can only be successfully routed to the node on 192.168.1.12. Similarly, packets from 192.168.32 can only be received by a node with IP address 192.168.31 which belongs to the same VLAN group as the source IP. The implementation of VLAN provides security, makes communication more efficient, and limits the broadcast domain. This makes it possible to integrate instrumentation and control networks into data networks in IoTs. Figure 12 shows communication between nodes with the same VLAN ID and nodes with a different VLAN ID.

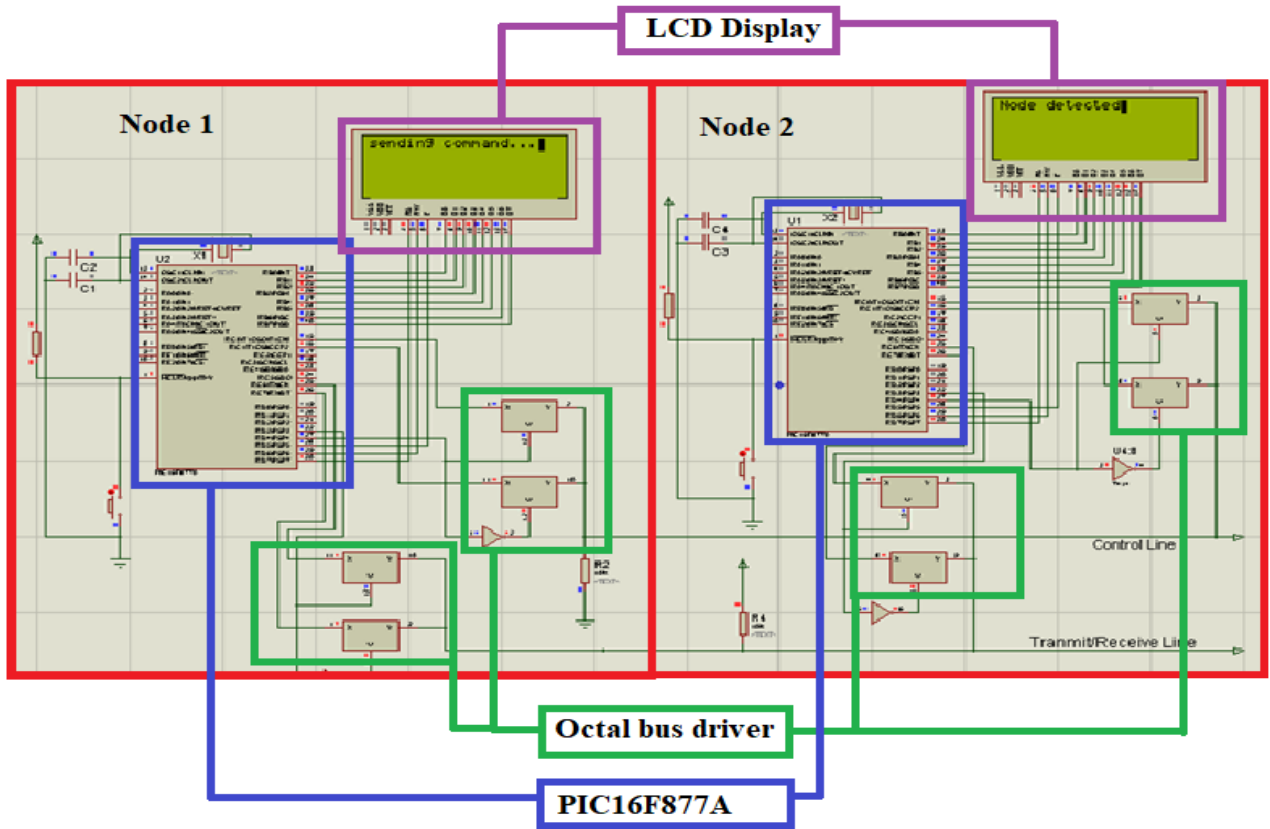


Figure 8: Two nodes communicating over collaborative network EIA-485.

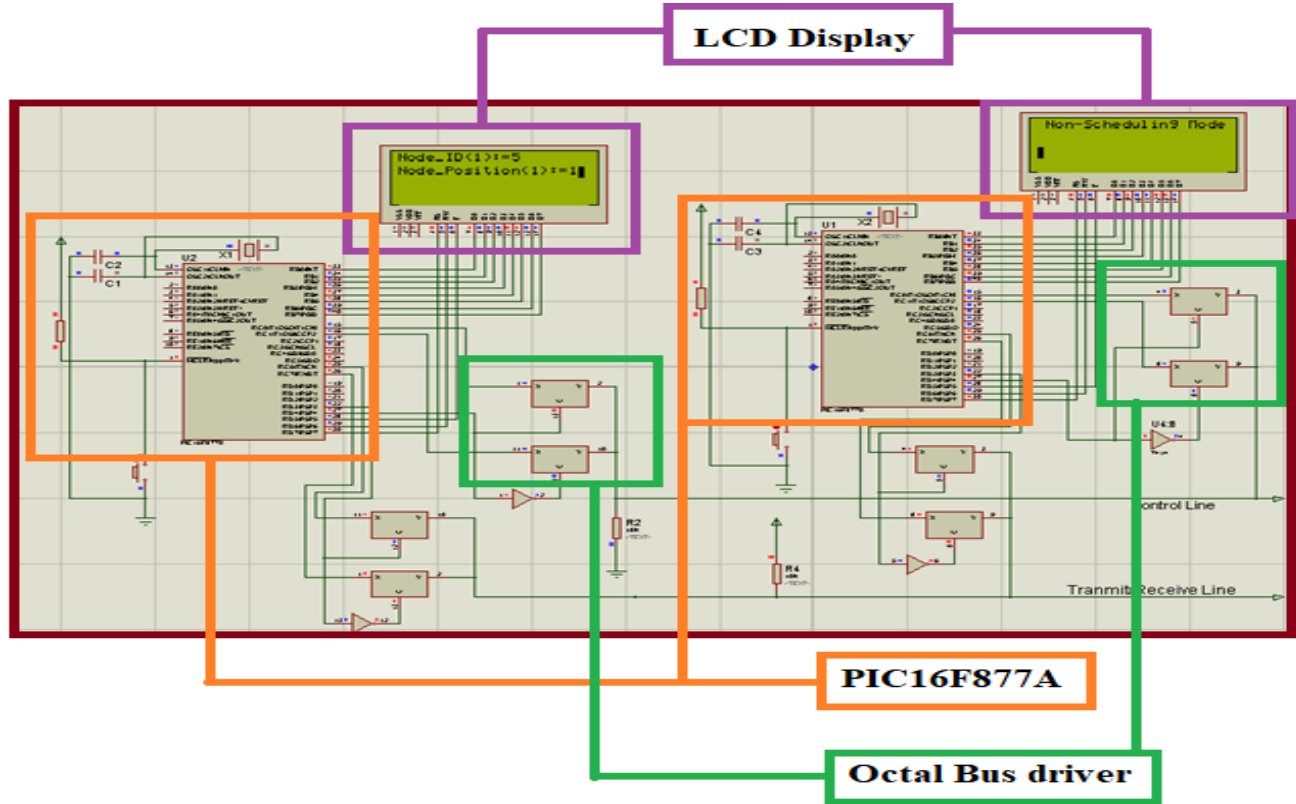


Figure 9: Two nodes exchanging data over two wires, collaborative network.



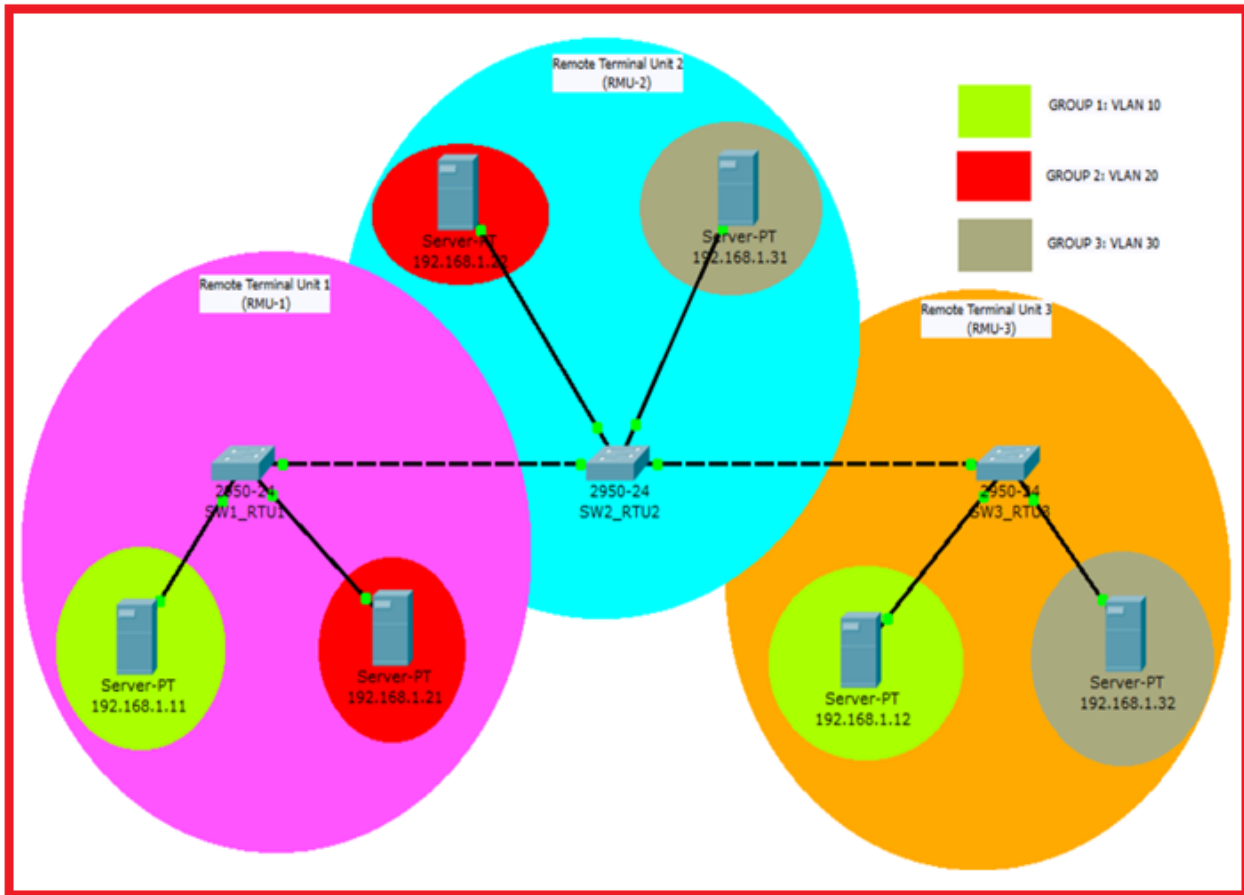


Figure 10: VLAN implementation in CISCO packet tracer.

```
Switch>show vlan
```

VLAN Name	Status	Ports
1 default	active	Fa0/4, Fa0/5, Fa0/6, Fa0/7 Fa0/8, Fa0/9, Fa0/10, Fa0/11 Fa0/12, Fa0/13, Fa0/14, Fa0/15 Fa0/16, Fa0/17, Fa0/18, Fa0/19 Fa0/20, Fa0/21, Fa0/22, Fa0/23 Fa0/24
10 GROUP1	active	Fa0/2
20 GROUP2	active	Fa0/3
30 GROUP3	active	
1002 fddi-default	act/unsup	
1003 token-ring-default	act/unsup	
1004 fddinet-default	act/unsup	
1005 trnet-default	act/unsup	

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
1	enet	100001	1500	-	-	-	-	-	0	0
10	enet	100010	1500	-	-	-	-	-	0	0
20	enet	100020	1500	-	-	-	-	-	0	0

Figure 11: Configure VLAN switch (SW1).

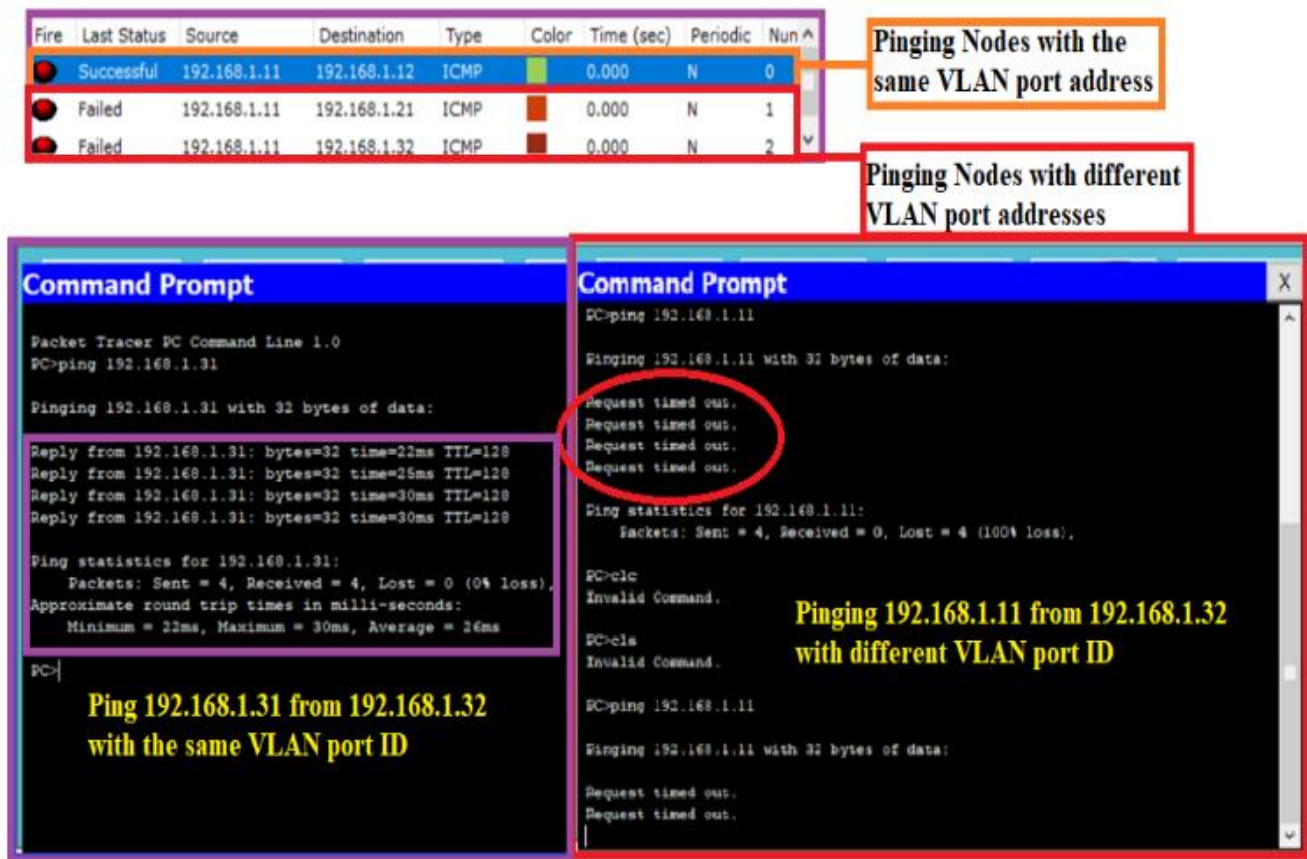


Figure 12: Inter and Intra VLAN group communications.

V. CONCLUSION

This paper presents two methods of implementing industrial control systems over existing IP-based networks. The use of two wires network to intelligently schedule nodes for data communication which was modeled and simulated using Proteus ISIS software can make control system nodes within the same RTU reliably communicate over IP-based networks as demonstrated in the result. The only challenge is the fact that you need to provide the communication line on which the nodes are connected leading to increased cost, unlike the VLAN where all that is needed is the configuration of the switches without the connection of any external device. The data speed is also reduced.

This approach makes communication over IP-network more deterministic and less probabilistic between nodes. In the second approach, the use of VLAN demonstrated that by creating different collision domains within the same network, different remote terminal units of industrial control systems can reliably communicate over IP-network without interfering with other computer networks. Only nodes of the same VLAN ID can communicate with each other. Therefore for enhanced security, more efficient communication, and limited broadcast domain, VLAN is used. This makes it possible to integrate instrumentation and control networks into data networks in IoTs. Summarily, the method used depends on the existing communication infrastructures.

REFERENCES

Al-Odienat, A.I. and Al-Lawana, A.A. (2008). The advantages of PID fuzzy controllers over the conventional Types. American Journal of Applied Science, 5(6): 653-658. DOI: <https://doi.org/10.3844/ajassp.2008.653.658>

Berger, A.S. (2002). Embedded System Design: An introduction to processes, tools, and techniques. CMP Books: Kansas, USA.

Choi, D.H and Kim, D.S. (2008). Wireless Fieldbus for networked control systems using LR-WPAN. International Journal of Control, Automation, and Systems, 6(1): 119-125.

Cho, J.U.; Q.N. Le and J. W. Jeon. (2009). An FPGA-based Multiple Axis Motion Control Chip. IEEE Transactions on Industrial Electronics, 56(3): 856-870.

Chow, M.Y. and Tipsuwan, Y. (2001). Networked-based control systems: A tutorial. Paper presented at the 27<sup>th</sup> Annual Conference of the IEEE industrial electronics society (IECON'01), Denver, Colorado, USA, 1593 – 1602: IEEE.

Cloosterman, M. B., Van de Wouw, N., Heemels, W. P. M. H., & Nijmeijer, H. (2009). Stability of networked control systems with uncertain time-varying delays. IEEE Transactions on Automatic Control, 54(7), 1575-1580.

Corrigan, S. (2002). Introduction to the Controller Area Network. Application Report, TEXAS Instrument.

Damas, M.; A.M. Prados; F. Gomez and G. Olivares. (2001). HidroBus System: Fieldbus for integrated management of extensive area of irrigated land. Microprocessors and Microsystems, 25: 177-184.

- Di Natale, M.; H. Zeng; P. Giusto and A. Ghosal. (2012).** Understanding and using the controller area network communication protocol: Theory and practice. Springer Science & Business Media.
- Durtertre, B. (2007).** Formal modeling and analysis of the Modbus protocol. In: proceeding of the 1<sup>st</sup> Annual IFIP working group 11.10 International Conference on Critical Infrastructure, Protection, Hanover, New Hampshire: 189-204.
- Fadali, M.S. and Visioli, A. (2013).** Digital Control engineering. Analysis and Design (2<sup>nd</sup> Edition). Waltham, USA: Elsevier.
- Gaddam, M. and Akula R. (2006).** Automatic tuning of PID controller using fuzzy logic. 8th International Conference on development and application systems Suceava, Romania: 120-127
- Garimella, P.; Y. W. E. Sung; N. Zhang and S. Rao. (2007).** Characterizing VLAN usage in an operational network. In Proceedings of the 2007 SIGCOMM workshop on Internet network management (INM '07). Association for Computing Machinery, New York, NY, USA, 305–306. DOI:<https://doi.org/10.1145/1321753.1321772>
- Gupta, R. A. and Chow, M. Y. (2008).** Gupta, R. A., & Chow, M. Y. (2008). Overview of networked control systems. In *Networked Control Systems*: 1-23. Springer, London. [https://doi.org/10.1007/978-1-84800-215-9\\_1](https://doi.org/10.1007/978-1-84800-215-9_1)
- Gupta, R. A. and Chow, M. Y. (2009).** Networked control system: Overview and research trends. *IEEE transactions on industrial electronics*, 57(7): 2527-2535.
- Hang, C.C. (2009).** Smith Predictor and its Modification Control System, Robotics, and Automation. *Encyclopedia of Life Support System-Vol. II*.
- Heemels, W. M. H.; A. R. Teel; N. Van de Wouw and D. Nešić. (2010).** Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance. *IEEE Transactions on Automatic Control*, 55(8): 1781-1796.
- Huang, J.; H. Handley; P. Pazos-Lago; A. Pinto; S. Kovacic; A. Collins; C. Keating; A. Sousa-Poza; A. Gheorghe; G. Rabadi; R. Unal; T. Cotter; R. Landaeta; and C. Daniels. (2019).** Digital Systems Engineering: Concepts, Challenges, and Enabling Technologies. Available at <https://www.researchgate.net/publication/336702108>
- Imtiaz, J.; J. Jasperneite; K. Weber; F. J. Goetz and G. Lessmann. (2008).** A novel method for auto-configuration of real-time Ethernet networks. Paper presented at 2008 IEEE International Conference on Emerging Technologies and Factory Automation: 861-868. IEEE.
- Jabbar, A.W.; M.H. Alsibai; N.S. Amran and S.K. Mahayadin. (2018).** Design and Implementation of IoT-Based Automation System for Smart Home. Paper presented at the 2018 International Symposium on Networks, Computers and Communications (ISNCC): 1-6.
- Jetto, L. and Orsini, V. (2009).** Relaxed Conditions for the Exponential Stability of a Class of Linear Time-Varying System. *IEEE Transactions on Automatic Control*, 54(7): 1580-1585
- Johansson, K. H.; M. Törngren and L. Nielsen. (2005).** Vehicle applications of controller area network. In *Handbook of networked and embedded control systems*: 741-765 Birkhäuser Boston
- Kalavathi, R.; A.Y. Reddy and C. Swathi. (2017).** A Comprehensive Analysis of Virtual Local Area Network (VLAN) and Inter-VLAN Routing Strategies, *Star*, 5(12): 14-16.
- Korčok, D. (2016).** Industry 4.0: The future concepts and new visions of factory of the future development. International Scientific Conference on ICT and E-business related research (Sinteza 2016) Singidunum University.: 293-298.
- Kunal, K.; Artifiante, A.Z.; Poetro, J.E.; Waseel, F. and Atmoko, R.A. (2020).** Accelerator meter implementation as feedback on 5 degrees of freedom Arm Robot. *Journal of Robotics and Control*, 1(1): 31-34.
- Lokman, S. F.; A. T. Othman and M. H. Abu-Bakar. (2019).** Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review. *EURASIP Journal on Wireless Communications and Networking*, 2019(1): 1-17.
- Malliarjun, G.H.; R.M. Umayal; K.K. Shiva and H. Raghavendra. (2017).** Programmable Logic Controller (PLC) in Automation. *Advanced Journal of Graduate Research*, 2(1): 37-45.
- Massimiliano, V. (2003).** Performance Improvement of Smith Predictor through Automatic Computation of Dead-Time. *Yokogawa Technical Report, Industrial Automation Department, Italy*. 5:25-30.
- Nehete, R. O. and Bhide, A. S. (2015).** Raspberry PI 3 based control and monitor remote machine automation. *Proc Int J Eng Comput Sci*, 6(1): 20151-20155.
- Niranjan, M., N. Madhukar; A. Ashwini; J. Muddsar and M. Saish. (2017).** IOT Based Industrial Automation. *IOSR Journal of Computer Engineering (IOSR-JCE)*: 36-40.
- O'Dwyer, A. (2005).** A reference guide to Smith Predictor-based methods for the compensation of dead-time processes. *ISSC, Dublin*: 231-238.
- Ojo, M.O.; Giordano, S.; Procissi, G and Seitanidis, I.N. (2018).** A Review of Low-End, Middle-End, and High-End IoT devices. *IEEE Access*, 6: 70528-70554
- Pan, I.; Das, S. and Gupta, A.(2011).** Tuning of Optimal Fuzzy PID Controllers with stochastic algorithm for Networked Control System with random time-delay. *ISA Transactions*, 50(1): 28-36.
- Patil, B.; M.H. Panchal; M.S. Yadav; M.A. Singh and M.D. Patil. (2017).** Plant Monitoring using Image processing, Raspberry Pi and IoT. *International Research Journal of Engineering and Technology (IRJET)*, 4 (10): 1337-1342.
- Powell, J. and Eng, P. (2013).** Profibus and Modbus: a comparison. *Automation.com*: 1-5.
- Peng, C.; Y. C. Tian and M. O. Tade. (2008).** State feedback controller design of networked control systems with interval time-varying delay and nonlinearity. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 18(12): 1285-1301.
- Rojko, A. (2017).** Industry 4.0 Concept: Background and Overview. *International Journal of Interactive Mobile Technologies (IJIM)*, 11(5):77-90.

- Saenko, I and Kottenko, I.V. (2014).** Design of Virtual Local Area Network Scheme based on Genetic Optimization and Visual Analysis. *Journal of Wireless Mobile Networks, Ubiquitous computing, and Dependable Applications (J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.)*, 5(4): 86-102.
- Shokri, S.; M. Shirvani; A.R. Salmani and M. Younesi. (2010).** Improved PI-Controllers Tuning in Time-delay Smith Predictor with Model Mismatch. *International Journal of Chemical Engineering and Applications*, 1(4): 290-293.
- Sun, X.; Y. W. Sung; S. D. Krothapalli and S. G. Rao. (2010).** A systematic approach for evolving VLAN designs. In *2010 Proceedings IEEE INFOCOM*: 1-9. IEEE.
- Smith, O.J. (1957).** Closed control of loop with dead time. *Chemical Engineering Progress*, 53: 217-219
- Swales, A. (1999).** Open Modbus/TCP specifications. *Schneider Electric*. 29: 3-19.
- Tindell, K.; H. Hanssmon and A. J. Wellings. (1994).** Analysing Real-Time Communications: Controller Area Network (CAN). In *RTSS*: 259-263.
- Tindell, K.; A. Burns and A. J. Wellings. (1995).** Calculating controller area network (CAN) message response times. *Control engineering practice*, 3(8): 1163-1169.
- Tipsuwan, Y. and Chow, M.Y. (2003).** Control methodologies in networked control systems. *Control engineering practice*, 11(10): 1099-1111.
- Thomasse, J.P. (2005).** Fieldbus technology in Industrial Automation. *Proceeding of the IEEE*, 93(6): 1073-1101.
- Vardhan, S. and Kumar, R. (2011).** Simulation for Time-delay Compensation in Network Control System. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunication (JSAT)*: 38-43.
- Vatanski, N.; J.P. Georges; C. Aubrun; E. Rondeau and S.L.J. Jounela. (2006).** Control Compensation Based on Upper-bound delay in NCS. Available online at: <http://arxiv.org/ftp/cs/papers/0609/0609151.pdf>. Accessed on 26<sup>th</sup> September 2020.
- Walsh, G. C.; H. Ye and L. G. Bushnell. (2002).** Stability analysis of networked control systems. *IEEE transactions on control systems technology*, 10(3): 438-446.
- Wang, X and Lemmon, M.D. (2010).** Event Triggering in Distributed Networked Control System. *IEEE Transactions on Automatic Control*, 56(3): 586-601.
- Wang, Y and Jang, S. (2019).** A pulse sensor interface design for FPGA based multi-sensor health monitoring platform. *International Journal of Biosensors and Bioelectronics*, 5(1): 23-27.
- Yamasaki, Y.; Y. Miyamoto; J. Yamato; H. Goto and H. Sone. (2011).** Flexible access management system for campus VLAN based on OpenFlow. Paper presented at 2011 IEEE/IPSJ International Symposium on Applications and the Internet: 347-351. IEEE.
- Yang, S.H.; X. Chen; L.S. Tan and L. Yang. (2005).** Time Delay and Data Loss Compensation for Internet-Based Process Control. *Transactions of the Institute of Measurement and Control*, 27(2): 103-118.
- Zampieri, S. (2008).** Trends in networked control systems. *IFAC Proceedings Volumes*, 41(2): 2886-2894.
- Zhang, H.; Y. Li and H. Zhu. (2011).** Development for Protocol of Conversion gateway of Profibus and Modbus. *Procedia Engineering*, 15:767-771.
- Zhang, Y.; M. Chen; N. Guizani; D. Wu and V. C. Leung. (2017).** SOVCAN: Safety-oriented vehicular controller area network. *IEEE Communications Magazine*, 55(8): 94-99.
- Zhao, Y. B.; G. P. Liu and D. Rees. (2009).** Design of a packet-based control framework for networked control systems. *IEEE Transactions on Control Systems Technology*, 17(4): 859-865.
- Zhang, H.; Y. Li and H. Zhu. (2011).** Development for protocol conversion gateway of PROFIBUS and Modbus. *Procedia Engineering*, 15: 767-771.