

A Process Control Network Analysis for Industry

Student:

Wilfredo Tovar Hidalgo / WilfredoTovarHidalgo@cmail.carleton.ca

Ramy Maarouf / [Ramymaarouf@cmail.carleton.ca](mailto:Ramymaarouf@cmail.carleton.ca)

Lukeman Hakkim / Lukemanhakkimsheikal@cmail.carleton.ca

Carleton University

School of Information Technology & Dep. of Systems and Computer Engineering

Ottawa, Ontario, Canada.

Course: ITEC5100W / SYSC5407W Plan and Design Networks.

Professor: Dr. Jun Huang

March 23rd, 2020.

**TABLE OF CONTENTS**

1. INTRODUCTION

1.1 Purpose

1.2 Project Scope

1.3 Background

2. MOTIVATION AND PROBLEM STATEMENT

3. RELATED WORK.

4. PROPOSED SOLUTION

4.1. Tools and Setup

5. IMPLEMENTATION OF NETWORK ARCHITECTURE

5.1 Methodology of Industrial Network Architecture

5.2 Model Topology

5.3 Routing Protocol

5.4 Operation Layer System Analysis

5.5 Scalability

5.6 Securing the Network.

6. IMPLEMENTATION OF NETWORK AUTOMATION

6.1 Types of Network Automation

6.2 Proposed Network Automation Solution

6.3 Few other use cases

7. GITHUB

8. RESULTS AND DISCUSSION

8.1 Network Performance Analysis

9. FUTURE SCOPE

10. CONCLUSIONS

REFERENCES.

**A PROCESS CONTROL NETWORK ANALYSIS FOR INDUSTRY**

Wilfredo Tovar Hidalgo Lukeman Hakkim Ramy Maarouf

School of Information Technology Dep. of Systems and Computer Engineering School of Information Technology

Carleton University Carleton University Carleton University

Ottawa, Canada Ottawa, Canada Ottawa, Canada wilfredotovarhidalgo@cmail.carleton.ca lukemanhakkimsheikal@cmail.carleton.ca ramymaarouf@cmail.carleton.ca

**ABSTRACT**

An interface linking a remote communications network and a process control system involves a storage device, a communication software stack and a user software layer. The user software layer empowers interfacing between the remote communications network and the process control system by addressing the communication software stack to operate in the process control system employing a process communication protocol, by monitoring the message traffic on the communication software stack, and by copying requested message traffic to the storage device.

This paper presents an industrial control system review to the development of a monitoring and infrastructure solutions with existing technology and infrastructure to aid the reliability and maintainability of large Process Control Networks (PCNs), to reduce production downtime attributed to intricate industrial network designs.

CCS CONCEPTS

• Process Control Networks (PCNs) • Remote Communications Network.

KEYWORDS

Process Control Networks (PCNs), Network Automation, Industrial Network Design.

1. INTRODUCTION

1.1 Purpose

Large Corporations contain and rely on many mechanical and electronic devices for industry production use. The original design of these older plants allowed the devices to work together without any remote-control capabilities, with monitoring and adjustment performed locally by operators. The introduction of, electronic circuits, Programmable Logic Controllers (PLC), and Programmable Automation Controllers (PAC) allowed operators to monitor and control mechanical devices or even personal performance remotely using point-to-point communication. As a result, defined communication protocols were developed for each vendor's requirements, leading to an introduction of a variety of standards and protocols for industrial communications. Examples of such industrial protocols are Common Industrial Protocol (CIP), Ethernet Industrial Protocol (EtherNet/IP), DeviceNet, ControlNet and Modbus. Transmission Control Protocol (TCP) emerged with the introduction of Ethernet to industrial sites and has become widely popular throughout the industry as it continues the simple, robust application layer protocol encapsulated in the industrial Ethernet standard. Ethernet-based has inherited handshake and timeouts from the serial communication protocols and utilize port 502 of the TCP/IP protocol.

**1.2 Project Scope**

The scope of this project consists of three parts:

A. The development of an Architecture solution for a real time complex industrial network to improve reliability through redundancy and the addition of monitoring and configuration tools to reduce downtime associated with communication issues reported by delay accounting software.

B. The development of automation script that is capable of doing certain task to improve the maintenance involved in the larger networks. It helps to configure, provision and analyse the network in an efficient way by running the script.

We have carefully examined the stochastic stabilization and disturbance attenuation issues for a discrete-time system at the device layer subject to random failures and network induced delays. By modelling the different random processes, an output feedback controller was designed to track the given decomposed set points. For brevity, we first consider the case that a single subsystem exists at the device layer, and then the results can be extended to the multiple subsystems case in a similar way adopted in [11]. The simulation was created to be configurable for expansion for future use and data collected in the current process control network.

**1.3 Background**

This project is focused on a TCP based Process Control Network at a large industrial processing facility and presents an industrial control system review to the development of a monitoring and infrastructure solutions with existing technology and infrastructure to aid the reliability and maintainability of large Process Control Networks (PCNs), to reduce production downtime attributed to intricate industrial network designs.

The layer design adopted in this paper is shown in Fig. 1, which possesses an upper operation layer and a local device layer with different objectives to be achieved. The below network architecture acts as a base layer for our project.

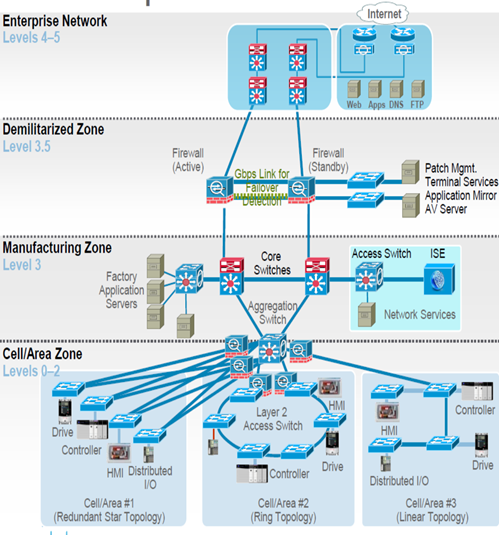


Fig 1. Scheme for industrial process control.

**2. MOTIVATION AND PROBLEM STATEMENT**

The rapid development of communication technologies has significantly accelerated the application of networked control systems (NCSs) in industrial processes [1] due to their economic productivity and flexibility in modularization. Elements of industrial processes like sensors, controllers, and other components are often connected over industrial Ethernet and communicated by exchanging packet-based messages. Because of the extensive data exchange over the Ethernet, there is a strong possibility that random packet dropouts and network-induced delays happen [2][3]. Thus, it is necessary to incorporate such effects into the overall consideration in the industrial processes. So here in this project we are mitigating the issues faced because of these delays in the traditional networks by implementing TCP based Process Control Network.

**3. RELATED WORK**

At the device layer, the controlled plants are NCSs with random failures, which are also known as the so-called networked fault-tolerant control (FTC) systems. FTC systems are of great practical importance and have attracted a lot of interest for many years, see, e.g., [4][5] and the references therein. M. Mariton, in [6], proposed a jump linear quadratic control scheme in a random environment. P. Shi, Et. al. in [7] investigated the robust disturbance attenuation problem for discrete-time systems. Z. Mao, Et al. in [8] studied the FTC problem for a class of nonlinear sampled-data systems via Euler approximate observer, while J. F. Martins, Et al. [9] and S. Tong, Et al. [10] are for three-phase induction motor systems and stochastic nonlinear systems by using advanced neural networks and fuzzy techniques.

Most of the collected results for FTC dilemmas can be categorized into two classes, in which the passive one is easier to design, and the active one incorporates fault detection and isolation technique. In this paper, due to the behaviour of Ethernet and OSPF at the device layer, linear discrete-time systems with random breakdowns and network-induced delays are considered. The stochastic breakdowns and delays are modelled in GNS-3. This problem is solved with an own developed Python Script by designing output feedback FTC controllers.

**4.** **PROPOSED SOLUTION**

A network simulation tool was developed to replicate the process control network for implementation as a test environment for process control network (PCN) architecture changes, the introduction of new protocols, and as a proof of concept for the purchase of commercially available simulation software. Simulation of the PCN was a necessary element in the project as an offline tool where network environment changes could be tested without incurring downtime in the processing plant. The software chosen to build the process control network simulator was the GNS-3 due it is free software, publicly available for research, development, and are compatible with a commonly used site network tools “Wireshark,” and “Solarwinds”, Additionally, there is extensive configuration documentation and example available online, a Python script code was used for Network automation, Python Modules used were Netmiko, NAPALM for Configuring VLANS in multiple switches, Back up Network configurations, and Reading device credentials.

**4.1. Tools and Setup**

|  |  |
| --- | --- |
| Platform | Windows, Linux |
| Network Simulation | GNS3 |
| Tools | Pycharm, Cisco IOS, python libraries like Netmiko, Napalm, simplecrypt, telnetlib |
| Language | Python |
| *Coding*  standards | PEP8, Pylint |

**5. IMPLEMENTATION OF NETWORK ARCHITECTURE**

**5.1 Methodology of Industrial Network Architecture**

As industrial production continues growing quickly due to the many benefits that automation provides. While modern corporations are exploring techniques to recognize and evaluate the difficulties of their practices, the increasing demand to optimize connectivity and enhance alliance between IT and operational technology (OT) to support maximize performance, profitability, and maintain competitiveness.

There are several steps involved in the development of an autonomous network topology which we followed to develop our industrial architecture, they are:

1. Assessment by classifying and prioritizing network concerns and uncertainties by carrying an evaluation of the data to be collected; the sub-steps in the process include: surveying technicians and personnel, inspecting infrastructure and documenting the outcomes. This appraisal will assist businesses in recognizing where gaps are in the network structure.

2. The second step is compounded by the Design. It should consider performance, security and activity of every novel design to obtain maximum value across the entire period of the venture.

3. The third step is Implementation; it includes configuration, connection, and experimentation.

4. The fourth step is Validation by verifying that the design was performed efficiently and properly to address the technological and economic criteria required.

5. The fifth step five is Monitor and Maintain, by applying a right mix of talent and ongoing monitoring tools (like NetScout Network Monitoring, SolarWinds Network Performance Monitor, that can help prevent the network and production from having unwanted downtime, and also monitoring the key parameters of the plant and implement secure remote access to the solving of problems before they influence production.

5.2. Model Topology (Fig 2).

The base simulation that was implemented first model a mixed topology using TCP and UDP applications, with the capability to rearrange the model to suit the test. The model created simulates a Dumbbell style topology, this model being chosen due to the ability to use the model for a variety of simulation tasks.

Our network topology was made according to a real industrial network architecture found in figure 2. We used around 40 Cisco nodes (due to the simulation tool limitation) that were divided into Cisco 7200 Routers, Cisco 3700 Switches, Cisco Catalyst 6500 Multilayer Switches, Cisco 5500 ASA Firewalls, and several types of servers.

5.3 Routing Protocol

For the implementation, we have used OSPF, which is a link-state routing protocol, which indicates that the routers switch topology data among their most adjacent neighbors.

***Why use OSPF and not BGP protocol ?***

During the project development proces, we faced different diatribes, including the selection of OSPF or GBP protocols, after performing an extensive survey of its application and feasibility for our approach the use of OSPF raise as the selected protocol for various reasons such as it is easily-configured and moldable. At the same time, BGP configuration contains a higher degree of complexity, OSPF can achieve convergence (the time a router takes to share and update the latest routing information) rate faster than that in the BGP. Also, OSPF is a type of hierarchical network topology while BGP is a type of mesh topology which is not needed in our project; finally, OSPF can be applied primarily on smaller-scale network which could be administered centrally such as our approach while BGP is frequently on large scale networks such as the internet.

***Advantage of OSPF in the industrial setup:***

The topology data is transmitted completely by the Autonomous System (AS) so that each router inside the AS holds a comprehensive understanding of the topology. In a link-state routing protocol, the next-hop address to which data is forwarded is determined by choosing the best end-to-end path to the eventual destination(Dijkstra algorithm). Since we have several locations with different departments in our topology, each OSPF router shares data concerning its local state (usable interfaces and reachable neighbors, and the cost of using each interface) to another router utilizing a Link State Advertisement (LSA) message. Every router handles the collected information to produce an equal database that represents the topology of the AS. From this database, each router computes its individual routing table utilizing a Shortest Path First (SPF) or Dijkstra algorithm. This routing table includes the addresses the routing protocol knows about, linked with a next-hop IP address and outgoing interface.

5.4 Operation Layer System Analysis

As shown in Fig. 1, the outputs and inputs of local factories at the device layer are forwarded to the operation layer through Ethernet to form the index review or prediction of performance within a sampling period. Thus, the packet dropout compensation and performance tracking problems will be investigated by using NMPC, and the impact of the disturbance at the device layer can be well mitigated by the governor at the device layer, so the disturbance will not be considered at operation layer.

5.5 Scalability

The Industrial Network Model developed to be a scaled simplified site model as:

1. The test site “A” systems contain multiples network rings in isolation, with each containing the same types of network devices and applications; therefore, responses of the individual network rings will be identical.

2. Proof of concept of a developed model could be proven on a smaller scale, and a simple, scalable model design would be more useful to Site Engineers in initial network design testing, with the ability to simply expand the model as needed for full-scale simulations.

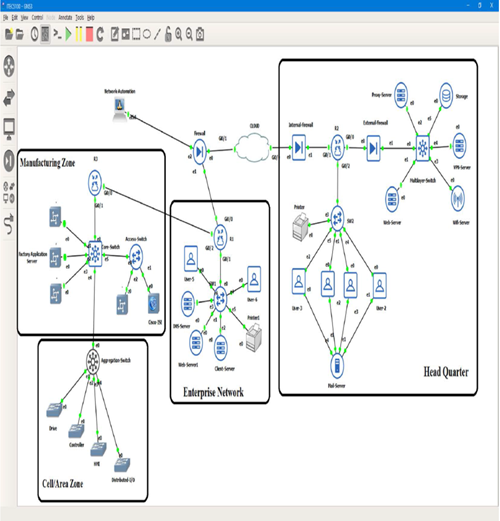


Fig 2. Network Topology

**5.6 Securing the Network.**

Today, cyber-attacks are increasing rapidly and becoming more sophisticated, so we strongly focused on securing our network internally and externally using Cisco ASA firewalls. To securely connect with different branches of the company around the globe, we configured Cloud VPN network to connect different company locations with one secure network. A site-to-site VPN network provides an authenticated, encrypted connection that hides information in about the sites you are visiting from attackers. The routers were configured from scratch and implemented with IPsec tunnel mode that acts as an additional security feature. The main advantage of IPSec is that, it can provide security for individual users if needed and moreover its useful for offsite workers, and also for setting up a secure virtual sub network within a company. The security strategy implemented in these routers are the IKEV2 tunnel security with ISAKMP policy 2, AES 256 for encryption, and pre-shared key group 5 for authentication. An access list is configured to identify the network traffic by filtering the packets to control its flow through a network and also restricting the access of users and devices to our network, thus providing an additional measure of security.

**6. IMPLEMENTATION OF NETWORK AUTOMATION**

Network automation[16] is a process or a methodology in which the piece of software developed will automatically configures, provisions, manages and tests network device on its own periodically. It is widely used by enterprises and service providers[17] to improve efficiency and reduce human error and operating expenses.

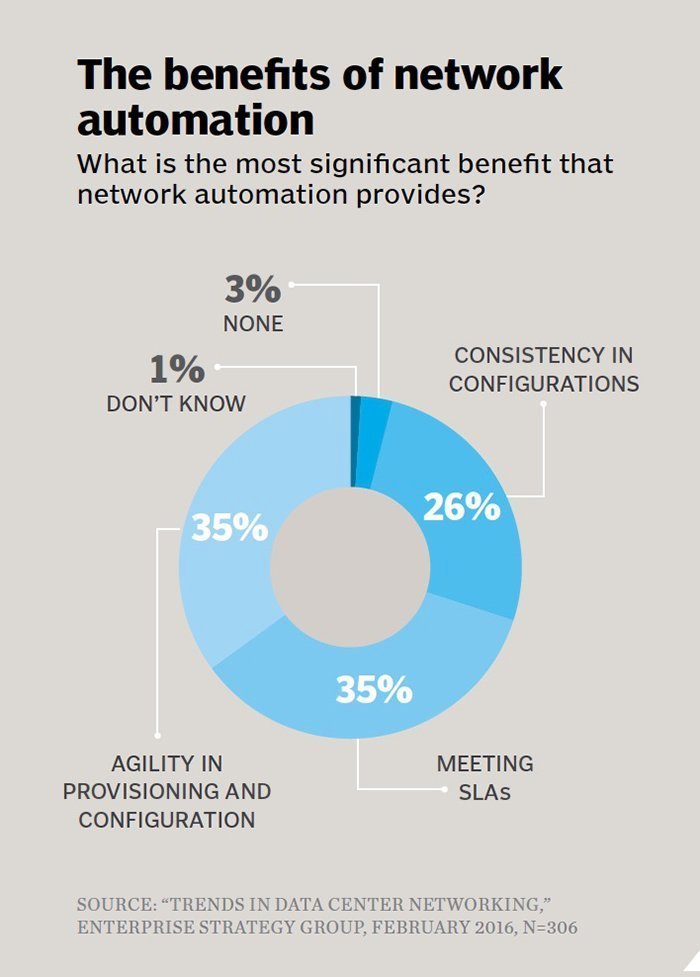
****

Fig.3 The above figure shows the benefits involved in network automation.

**6.1 Types of Network Automation**

Automation can be developed and deployed in any type of network, including local area networks (LANs), wide area network (WANs), data center networks, cloud networks and wireless networks. Different types of network resources which can be automated are given below:

* Command-line interface ([CLI](https://searchwindowsserver.techtarget.com/definition/command-line-interface-CLI))
* Application programming interface (API)
* Script-driven network automation

In this project we are using script driven network automation using open source programming language - Python.

**6.2 Proposed Network Automation Solution**

In this project we are focusing on three scenarios. They are:

1. **Configuring VLANS in multiple switches:**

In a very big network, in order to configure the VLAN in muliple switches, one have to ssh to that particular switch and do the configurations. But in our solution we have developed a python script which will automatically configure the VLAN in n number of switches in the network.

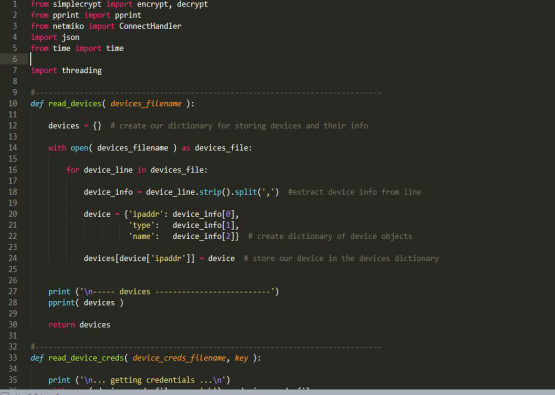
1. **Back up Network configurations​**:

In case of unexpected network shutdown or physical damage to the nodes, it is impossible to backup the configuration, to address this issue we have developed a network backup script to save all the config data from the nodes, which act as a network management software.

1. **Reading device credentials:**

In an existing architecture, in order to make changes and maintain the network one needs to know what configurations have already been provisioned to mitigate this we have developed a script to read the credentials so that it will ease the understanding of architecture for a network admin.

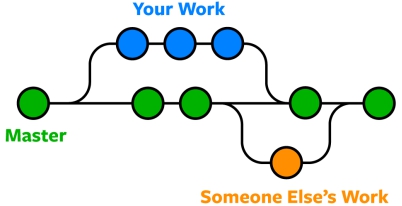
Note: The python script have been pushed in Git hub repo. Sample snap of Core script is mentioned below:

Fig.4 Shows the core sample python script

**6.3** **Few other use cases**

* It can be used to discover topologies.
* Managing bandwidth and finding fast reroutes to implement the best computing paths.
* Performing root cause analysis
* Setting performance benchmarks
* Updating software
* Implementing security and compliance

**7. GITHUB**

​

​

Fig 5. Shows the Git hub workflow

​

**7.1 Why Git?​**

Git keep these revisions straight, storing the modifications in a central repository. This allows the contributors to easily collaborate, as they will be able to download a new version of the committed software, make changes, and upload the newest commit. Every developer added as a collaborator will be able to make changes and get notified if they is a change in the existing project repository.

We have breakdown our project into different issues and worked in parallel to complete the project.

[Click here to check the project tracking in GitHub](https://github.com/users/lukemanhakkim/projects/2)

The link to our repository is given below:​

[Click here to check the code in GitHub](https://github.com/lukemanhakkim/A-PROCESS-CONTROL-NETWORK-ANALYSIS-FOR-INDUSTRY-)

**8. RESULTS AND DISCUSSION:**

As we are done with our real time industrial network architecture using GNS3 and developed Python script for Network Automation, its time to test the stability of our model. The stability of the network can be analysed by various metrics which is explained below:

**8.1 Network Performance Analysis**

We started searching for an open-source network monitoring tool to test our architecture after entirely building it. It uses a 30 days free trial for SolarWinds Tool to uses it as a Network Performance Monitor (NPM). It possesses several features that can be employed to metric real-time network performance.

NPM is able to perform automatic scheduling of device discovery to find nodes in the topology and map it to the monitoring database, allowing the creation of a dynamic network maps to visually track their performance analysis. NMP catches data sent from the node to generate network alerts, reports, and interactive charts so the technical staf can readily check the network performance.

The data obtained in Figures 6 and 7 illustrate that the novel link connected to the firewall executed as expected, enabling network movement flow to the firewall when the organization network switch is in rest. The Redline specified uplink breakdown, in Figure 6, presents the time of failure of the PCN uplink to the core/broker switch; in Figure 7, the condition of the crash is observed on the left-hand in the chart as a loss of energy in the switch. During the previously mentioned breakdown, in the left-hand Figure 7, it reveals that through this season, there were not identified downtimes on the PCN sharing switch.

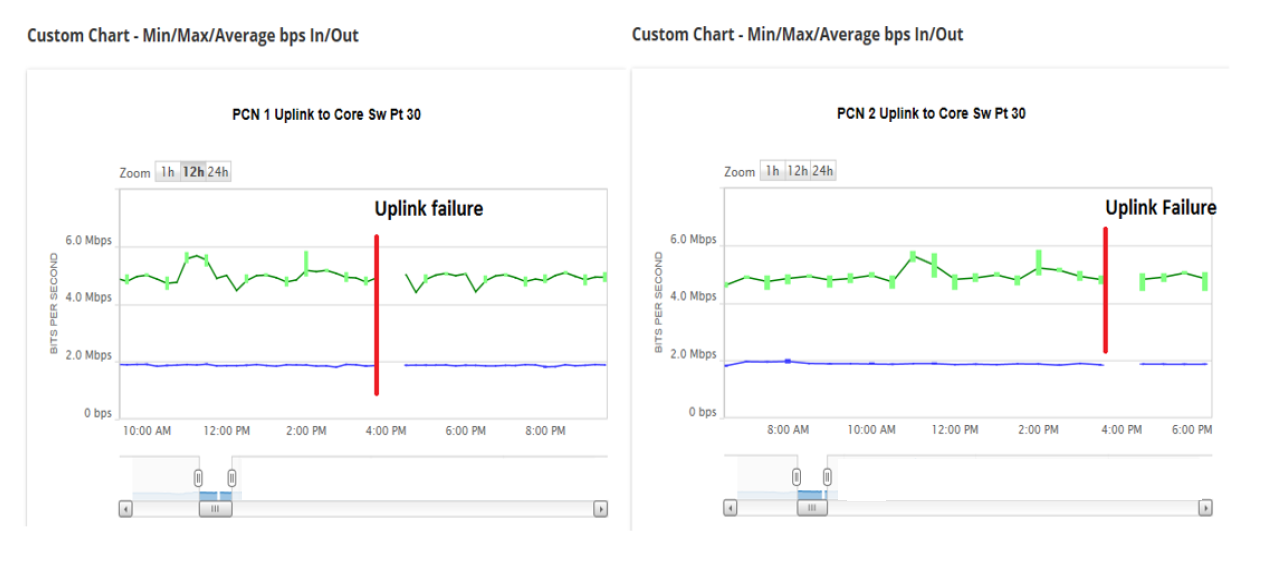


Fig 6 . Solarwinds Metrics - Monitoring Data

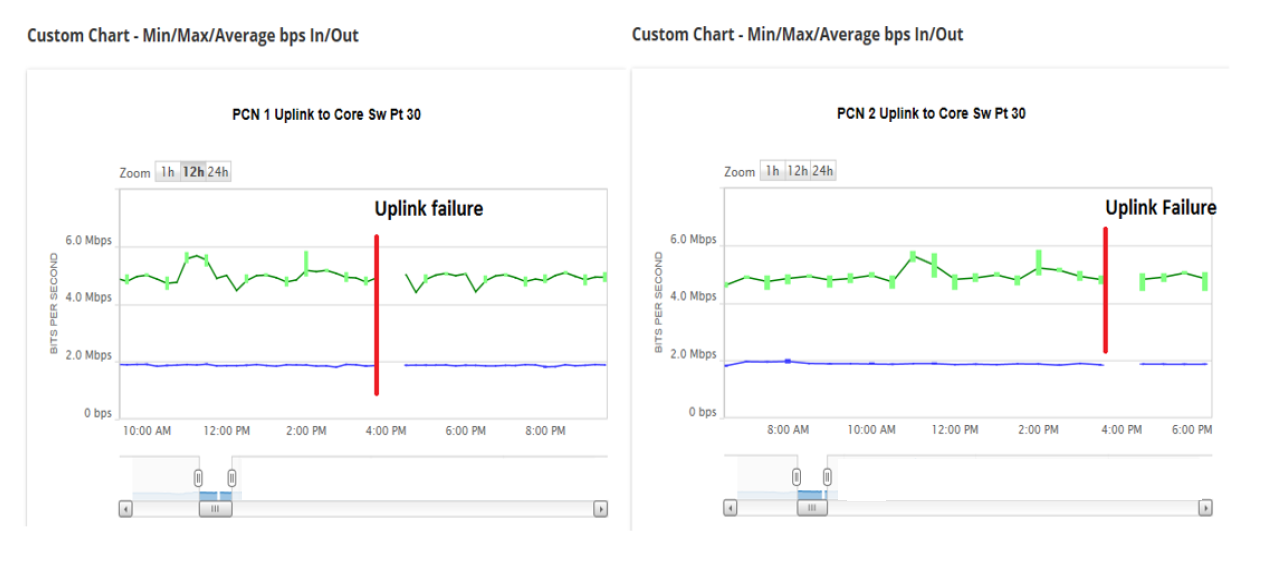


Fig 7 . Solarwinds Metrics - Monitoring Data.

The Figure 8. exhibits a tiny decrease in the downtime correlated with a system function and both; i) the process control network structure variations and ii) the included capability to monitor.

****

Fig 8. The Downtime associated to PCN Network Failure obtained from Delay Accounting software.

**9. FUTURE SCOPE**

What’s the answer to managing today’s networks? ​ ​

**Automation​**

The skill of Humans and manual work needed can no longer keep pace with network innovation, complexity, innovation and the change. ​

The Future would be “self-driving networks,” “self-healing networks,” “intent-based networking,” which uses efficient algorithms (artificial intelligence (AI), machine learning (ML)), and provides support to support modern network operations.

**10. CONCLUSIONS**

In this paper, an efficient process control network automation design was developed and simulated for industrial processes. At the device layer, the tracking problem is examined for a system with stochastic crashes and network-induced delays. At the operation layer, the optimization obstacle is resolved by employing the NMPC approach. Finally, the introduced technique has been validated by a simulation of a network-based process in GNS3. The fault exposure and filtering intricacies will be examined in research. The network automation script using python increased the network configuration management, which needs further improved by adding more use cases to it, which can later be developed as a network management software that would be a prospective future work.

**REFERENCES.**

[1] S. Yin, X. Li, H. Gao, and O. Kaynak, “Data-based techniques focused on modern industry: An overview,” IEEE Trans. Ind. Electron., vol. 62, no. 1, pp. 657–667, Jan. 2015.

[2] L. Zhang, H. Gao, and O. Kaynak, “Network-induced constraints in networked control systems—A survey,” IEEE Trans. Ind. Informat., vol. 9, no. 1, pp. 403–416, Feb. 2013.

[3] Y. Tang, H. Gao, W. Zhang, and J. Kurths, “Leader-following consensus of a class of stochastic delayed multi-agent systems with partial mixed impulses,” Automatica, vol. 53, pp. 346–354, 2015.

[4] S. Yin and Z. Huang, “Performance monitoring for vehicle suspension system via fuzzy positivistic C-means clustering based on accelerometer measurements,” IEEE/ASME Trans. Mechatronics, vol. 20, no. 5, pp. 2613–2620, Oct. 2015.

[5] S. Yin, X. Zhu, and O. Kaynak, “Improved PLS focused on key-performance-indicator-related fault diagnosis,” IEEE Trans. Ind. Electron., vol. 62, no. 3, pp. 1651–1658, Mar. 2015.

[6] M. Mariton, “Jump linear quadratic control with random state discontinuities,” Automatica, vol. 23, no. 2, pp. 237–240, 1987.

[7] P. Shi, E. K. Boukas, S. K. Nguang, and X. Guo, “Robust disturbance attenuation for discrete-time active fault tolerant control systems with uncertainties,” Optim. Control Appl. Methods, vol. 24, no. 2, pp. 85–101, 2003.

[8] Z. Mao, B. Jiang, and P. Shi, “Fault-tolerant control for a class of nonlinear sampled-data systems via a Euler approximate observer,” Automatica, vol. 46, no. 11, pp. 1852–1859, 2010.

[9] J. F. Martins, V. F. Pires, and A. Pires, “Unsupervised neural-network-based algorithm for an online diagnosis of three-phase induction motor stator fault,” IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 259–264, Feb. 2007.

[10] S. Tong, T. Wang, and Y. Li, “Fuzzy adaptive actuator failure compensation control of uncertain stochastic nonlinear systems with unmodeled dynamics,” IEEE Trans. Fuzzy Syst., vol. 22, no. 3, pp. 563–574, Jun. 2014.

[11] T. Chai, L. Zhao, J. Qiu, F. Liu, and J. Fan, “Integrated network-based model predictive control for setpoints compensation in industrial processes,” IEEE Trans. Ind. Informat., vol. 9, no. 1, pp. 417–426, Feb. 2013.

[12]<https://www.rockwellautomation.com/en_NZ/news/automationtoday/detail.page?pagetitle=Five-Steps-to-Building-Secure-Industrial> Networks&content\_type=magazine&docid=195c8e67e92ccbc5e884902019e46abc

[13][https://www.metaswitch.com/knowledge-center/reference/what-is-open-shortest path-first-ospf](https://www.metaswitch.com/knowledge-center/reference/what-is-open-shortest%20path-first-ospf)

[14]https://www.metaswitch.com/knowledge-center/reference/what-is-open-shortest path-first-ospf

[15] <https://www.vmware.com/topics/glossary/content/internal-firewall>

[16]https://www.networkcomputing.com/networking/ansible-network-automation-tutorial

[17] <https://searchnetworking.techtarget.com/definition/network-automation>

[18] <https://www.juniper.net/us/en/products-services/what-is/network-automation/>