ENHANCING INDUSTRIAL NETWORK EFFICIENCY WITH ADAPTIVE P4 DATA PLANE SOLUTIONS

J. Sudheer Kumar, Bejjenki Chaithanya, Meghana. Akula, Vavilla Sandeep, Manee Jayanth Kumar

Assistant Professor in department Of IT Teegala Krishna Reddy Engineering College

askphy@gmail.com

UG Scholars In Department of IT Teegala Krishna Reddy Engineering College

chaithanyabejjenki@gmail.com , meghana.a442@gmail.com , vavillasandeep75@gmail.com , jayanthpatel1920@gmail.com

Abstract

Industrial networks rely on standard real-time communication protocols, such as ProfiNet RT. These real-time protocols use cyclic data exchange between IO devices and controllers. Each IO device reports its internal state to the controller at a predefined frequency, even if the state of the device is unchanged. These reports are essential to accurately monitor the health of the devices, but network resources are limited and it is not advisable to overload the network with unnecessary packets. The traffic generated by a single device is insignificant, but in an industrial site with hundreds of such devices, the number of packets to be transmitted adds up. As cloud-based industrial controllers (e.g., cloud-based soft-PLCs) become more prevalent, all generated IO device traffic must be forwarded over the access link to edge computing or private/public cloud infrastructure. Wireless (e.g. 5G radio) transmission of many small packets leads to spectrum efficiency issues and high power consumption. In this paper, we propose an in-network solution to significantly reduce industrial network traffic by cooperating with two P4 programmable network elements deployed on both sides of an access link. Excess traffic is filtered out and new data content is cached at both ends while detecting both link and device failures in real-time. The adaptive mechanism introduced allows the system to automatically optimize its efficiency and performance by dynamically enabling and disabling traffic filtering/caching.

INTRODUCTION

Over the past few decades, industrial controls have evolved from boards full of control relays to modern Programmable logic controllers (PLCs). Other types of specialty controllers such as distributed control system controllers for the process control industry have also developed as industrial controls became more and more automated. Nowadays the whole industrial network is changing quickly. The associate

editor coordinating the review of this manuscript and approving it for publication was Renato Ferrero . cloud-based solutions appear to replace the traditional Industrial controller hardware. With the advent of 5G, a stable, highly reliable network connection can be established over the radio like the wired counterparts. The communication between different parts of an industrial site or other remote locations, such as private and edge cloud nodes became a viable option over the radio. While the idea is functional, connecting programmable logic controllers and IO devices via 5G radio presents a number of new challenges. Sensors and PLCs are currently implemented to send status signals with predetermined frequencies usually between 1 and 1000 Hz. This amount of messages leads to a high overhead affecting both spectral and energy efficiency. Analyzing the contents of these packets, we found that most of the data is redundant, they do not contain any new sensor information. While these packets are redundant, they also take a crucial part in the connection, because both PLCs and IO devices are very sensitive to packet loss and jitter. In our work, we classify these messages and treat redundant packets as simple life-signals, while using the valuable sensor information to operate the system.In this paper, we focus on real-time industrial protocols that implement a communication behavior called cyclic data exchange (for example ProfNet [1]). In such protocols, each IO device and the appropriate PLC cyclically exchange data packets with a predefined frequency. This predefined frequency communication does not allow us to simply change existing wired networks to wireless, simply because hundreds or thousands of such devices will deplete the radio spectrum very quickly. To overcome the radio limitation, we propose a new method, using the concept of in-network computing to substantially reduce the network traffic that we need to transmit over the radio link. By installing a programmable switch before the radio transceiver at both ends of the communication, we can get rid of the unnecessary data transmission from IO evices whose internal state is unchanged. Each programmable switch can store the device states, and is able to detect missing packets. Since both switches have this information, the sender switch only needs to transmit updates and error information over the radio.

The receiver switch, if it does not receive any status updates or error information from the radio, will continuously generate and send out the life signals to the PLC. Our solution adds a layer of logic on top of existing protocols and does not require any modifications to the protocol itself, the PLC, or the IO devices. Seamless deployment of this system is possible by placing two instances of the P4-programmable switches (or smartness) at the two sides of the critical link for example the radio. No further changes in the infrastructure or modifications in the configurations are required.

LITERATURE SURVEY

To provide computer networks with a high degree of flexibility and scalability Software Defined Networking (SDN) [3] introduced a new way of programming abstractions by decoupling the data and the control plane functionality. While the literature on control plane programmability has a rich past, difficulties of programmable and portable data planes have just started gaining attention in recent years. To offer network developers the desired flexibility, specific programming languages have evolved. These languages let experts describe the entire packet processing pipeline in a protocol-independent way from a high-level abstraction. P4 [4] is one of the language propositions, which has achieved the most influential community support, backed by members from both industry and academia. The language has numerous compilers for diverse software and hardware targets, ranging from general-purpose processors, NetFPGAs [5] and SmartNICs, to custom-designed sets of ASICs such as Intel Tofino. Protocol-independent network programming opens up the fields for a new era, in which switches are more than simple packet-forwarding tools. By offloading low-latency processing rules to in-network devices, network hardware can also take part in calculations on the application level during communication.

These newborn paradigms are called in-network computing and edge computing, where server-based computations are offloaded partly or completely to the programmable switches. In contrast with cloud computing where computing servers are located far away from end-users, edge computing and in-network computing offer computation very close to the endpoints. Doing so will minimize the response time,therefore satisfying the low-latency constraints in various domain-specific applications became viable. The

Integration of SDN into industrial networks has proven to be beneficial in terms of increased reliability, scalability, and cost-efficiency A number of scientific papers have been published over the years on the topic, highlighting its advantages as well as potential challenges that need to be addressed [6], [7]. In particular, research has focused on improving network performance, reducing latency, increasing the reliability, and security of industrial networks [8]. Other research topics have included the development of algorithms for better traffic management, optimizing energy consumption in industrial settings, and enabling mobility across various industrial applications [9]. Kim et al. demonstrate a prototype implementation for inband network telemetry with P4 language, using a software switch as the implementation platform. They show how their implementation can be used to diagnose various performance problems. Jin et al. [11] implement NetCache, which is a new rack scale key-value store architecture that leverages in-network caching to provide dynamic load balancing across all storage servers. Bremler-Barr et al. [12] show how a simple L7 load balancer over Software-Defined Networks can work. NETHCF a line-rate in-network system using programmable switches to design a novel defense against spoofed IP traffic is introduced in [13]. Laki et al. [14] present that with the advent of P4, description, validation, and evaluation of AQM algorithms in a generic framework has become possible since the different drop policies applied by these methods can be implemented in ingress and/or egress control blocks of a P4 program. Not originally developed for the execution of complex algorithms, these devices are restrained by their processing capability, and the rules to be executed on them are limited.

EXISTING SYSTEM

Over the past few decades, industrial controls have evolved from boards full of control relays to modern programmable logic controllers (PLCs). Other types of specialty controllers such as distributed control system controllers for the process control industry have also developed as industrial controls became more and more automated.

Disadvantages

In contrast with cloud computing where computing servers are located far away from end-users, edge computing and in-network computing offer computation very close to the endpoints. Doing so will minimize the response time, therefore satisfying the low-latency constraints in various domain-specific applications became viable.

PROPOSED SYSTEM

The proposed pipeline has two main tasks:

1) caching IO data and sending automatic responses and

2) packet filtering and state change detection. Each IO device expects replies to their messages, therefore traditional traffic filtering (dropping packets) is not feasible in our scenario. The P4-switch has to handle the incoming data packets from local IO devices, store their carried contents, and send back a response on behalf of the PLC controller on the other side of the radio network, without transmitting data over the wireless link.

The latest packet data is stored by the switch used to generate the response expected by the other end. If the packet data of a particular device is changed, the packet is forwarded to the switch on the other side to update the stored data content.

Advantages

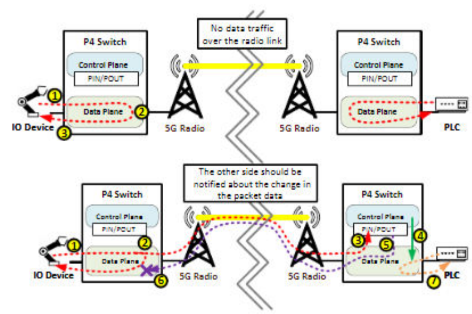
Our solution makes it possible to convert existing systems using different real-time communication to wireless without the difficulties of deploying new protocols.

❖ To avoid the transmission of unnecessary packets over the radio link,

❖ The proposed approach combines three key functionalities: 1) algorithms for storing and updating packet data and reducing data traffic over the radio link, 2) methods for monitoring the status of the radio link and the different IO devices, 3) dynamic enable and disable of the traffic reduction, to adapt to the network characteristics.

❖ We propose an in-network traffic reduction method. Accordingly, we assume two P4-programmable switches at the two sides of the radio link (as shown in Fig. 1). The two programmable switches cooperate to keep track of the latest state (reported packet data) of IO devices, filter out redundant Traffic, recognize whether a device is in an active or a passive phase and turn traffic reduction on and off accordingly, and detect the device and radio link failures.

ARCHITECTURE



IMPLEMENTATION

Admin Module:

Assume a cloud-assisted industrial environment where the data communication between IO devices (sensors and actuators) is deployed in the industrial site and software PLCs running in the cloud (public, private, or edge). Admin module is an entity in the cloud assisted network who will add plc and servers.

PLC Module:

Each PLC continuously queries the state of the IO devices and checks the status of servers who are busy and waiting to accept requests. The typical update period fits into the range of how busy servers are and may vary from server to server. In our system model, we assume three operational phases of each device: 1) Active phase when the reported IO data continuously changes, representing the case when the servers (e.g.,an actuator) performs an industrial task or its environment is not static. PLC can login to the application and check various servers who are busy and waiting to accept requests based on their status, plc will upload data and send to servers and get responses from servers. PLC can cancel request and upload new task and check response from server

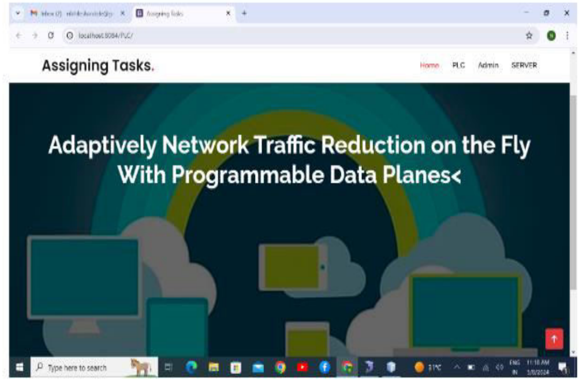
Server Module (s1) :

Using this server module there are three servers, s1 , s2 and s3 servers who are waiting for requests from plc and check status based on how busy they are plc will send data to less busy server and server can check requests cancel task if plc requests to cancel before execution and send decrypted data to plc.

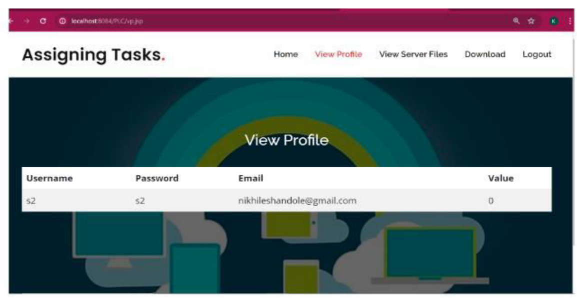
S2 Module:

Using this server module server 2 will check if there is change in task allocation by the plc module from the previous server and delete previous server data

RESULTS



Home page



Server page

CONCLUSION

The communication between IO devices and PLCs produces countless small packets, so it was not feasible to build a radio link between the devices and the cloud-based soft-PLCs. In this paper, we proposed a model in which device states are managed locally to minimize communication over critical links. Using in-network

computing, we have presented our prototype that can integrate into real-time industrial environments without affecting the required reliability and without the need to modify the protocol, PLC, or IO devices. We have also shown that by managing redundant sensor data locally, we can significantly reduce the average load on critical links, such as 5G radios between different parts of an industrial site. Through our evaluations, we have demonstrated the benefits of the proposed method in several aspects. We have quantified the impact of traffic reduction both in the standard scenario and using our adaptive switching. We found that our implementation has no significant impact on information latency while also having relatively low resource requirements.

REFERENCES

[1] R. Pigan and M. Metter, Automating With PROFINET: Industrial Communication Based on Industrial Ethernet. Hoboken, NJ, USA: Wiley, 2008.

[2] C. Györgyi, K. Kecskeméti, P. Vörös, G. Szabo, and S. Laki, ''In-network solution for network traffic reduction in industrial data communication,'' in Proc. IEEE 7th Int. Conf. Netw. Softwarization, Jun. 2021, pp. 191–195.

[3] M. Karakus and A. Durresi, ''A survey: Control plane scalability issues and approaches in software-defined networking (SDN),'' Comput. Netw., vol. 112, pp. 279–293, Jan. 2017.

[4] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco,A. Vahdat, G. Varghese, and D. Walker, ''P4: Programming protocol-independent packet processors,'' SIGCOMM Comput. Commun. Rev., vol. 44, pp. 87–95, Jul. 2014, doi: 10.1145/2656877.2656890.

[5] J. W. Lockwood, N. McKeown, G. Watson, G. Gibb, P. Hartke, J. Naous, R. Raghuraman, and J. Luo, ''NetFPGA—An open platform for gigabitrate network switching and routing,'' in Proc. IEEE Int. Conf. Microelectronic Syst. Educ. (MSE), Jun. 2007, pp. 160–161.

[6] T. Kobzan, I. Blocher, M. Hendel, S. Althoff, A. Gerhard, S. Schriegel, and J. Jasperneite, ''Configuration solution for TSN-based industrial networks utilizing SDN and OPC UA,'' in Proc.25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA), Sep. 2020, pp. 1629–1636.

[7] D. Henneke, L. Wisniewski, and J. Jasperneite, ''Analysis of realizing a future industrial network by means of software-defined networking (SDN),'' in Proc. IEEE World Conf. Factory Commun.Syst. (WFCS), May 2016, pp. 1–4.

[8] M. Cheminod, L. Durante, L. Seno, F. Valenza, A. Valenzano, and C. Zunino, ''Leveraging SDN

to improve security in industrial networks,'' in Proc. IEEE 13th Int. Workshop Factory Commun. Syst. (WFCS), May 2017, pp. 1–7.