JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Adaptive Network Traffic Reduction on Fly with Programmable Data Planes

Mrs. JALA SRAVANTHI - Assistant Professor, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)
Mr. PAMALA SANJAY KUMAR - Student, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)
Ms. PALLE KEERTHANA - Student, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)
Mr. GOKARAPU JASHWANTH - Student, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)

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.

1. 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. More and more 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 similar to 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 number 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 ProfiNet [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 devices whose internal state is unchanged. Each programmable switch is capable of storing the device states, and also 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 smartNICs) 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. The proposed solution provides a real-time reaction to link and device failures but still does not introduce significant computational and memory overhead. The reduction in network load achieved by eliminating redundant traffic opens up new uses for wireless communications that were previously not possible due to the limitations described above, without the need to implement new communication protocols. Our solution makes it possible to convert existing systems using different real-time communication to wireless without the difficulties of deploying new protocols. This paper extends the work of [2] with the following additional features making the method more efficient in non-static environments: 1) We introduce an adaptive mode of operation, where the traffic reduction module dynamically switches on and off for each device depending on the number of state changes. 2) We propose a missing event detection method to recognize both device and link failures quickly. Our pipeline implements a mechanism that exchanges information about the availability of IO devices and also about the state of the radio link. 3) We present the integrated pipeline implemented in P4 and evaluate it on an Intel/Barefoot Tofino-based hardware switch. 4) We have also extended the evaluation with several new scenarios, various network settings, and further aspects.

2. LITERATURE SURVEY

To provide computer networks with a high degree of flexibility and scalability Software Defined Networking (SDN) 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 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 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. In particular, research has focused on improving network performance, reducing latency, increasing the reliability, and security of industrial networks. 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. Kim et al. demonstrate a prototype implementation for in-band 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. 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. 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. Laki et al. 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. However several studies showed, those in-network devices can effectively be used to execute simple control algorithms for example controlling an inverted pendulum. Cesen et al. offer an emergency action execution platform to overcome the latency problems caused by the possible connection problems between the devices and the controller. The authors produce emergency packets, with stop commands, directly from the data plane. The emergency action can be triggered if the switch detects a packet with a specific payload. In Industrial IoT, the infrastructure needs to handle the data generated by millions of sensors, and support the control of production processes in real time. Mai et al. proposed an edge- and in-network computing architecture for industrial IoT, where they use complex event processing to transform the application-specific functions into operation units and offload them to the network devices. Security is also a serious concern in industrial IoT. Wusteney et al. in their paper, analyze the arising problems and quantify the delay and jitter impacts caused by using firewalls and packet filters in TSN networks.

3. OVERVIEW OF THE SYSTEM

3.1 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.

3.1.1 Disadvantages of Existing System

Assumption of Offloading to Edge Servers: Many existing methods assume tasks are offloaded to edge servers, which may not always be the most efficient approach, considering the limitations of edge servers in computing power and memory.

Limited Flexibility: Some methods rely on predetermined scheduling techniques without considering dynamic factors like varying network conditions or real-time resource availability.

Potential Latency Issues: Offloading tasks to edge servers or remote resources may introduce latency, especially if the communication distance is significant. This latency can impact application responsiveness and user experience negatively.

Complexity in Implementation: Certain scheduling techniques proposed in existing methods may involve complex algorithms or overhead, making their implementation and deployment challenging, especially in real-world vehicular environments.

3.2 Proposed System

In this setup, a P4-switch handles incoming data packets from local IO devices. It stores the content carried by these packets and responds on behalf of the PLC controller located on the other end of the radio network.

Importantly, data isn't transmitted over the wireless link, ensuring efficient operation. The switch keeps track of the most recent packet data to generate the expected responses.

If the data in a packet from a specific device change, the packet is forwarded to the switch on the other end to update the stored content.

3.2.1 Advantages of Proposed System

Minimized Time Interval: This approach aims to reduce the time gap between task offloading and execution, ensuring tasks are swiftly executed without unnecessary delays.

Consideration of Distance: By factoring in the distance between client vehicles (CVs) and worker vehicles (WVs) during the selection of WVs for task offloading, it guarantees tasks are assigned to WVs within communication range. This consideration enhances network stability.

Energy Efficiency: The method takes energy costs into account, which is particularly advantageous for electric vehicles, given its potential impact on their driving range. By optimizing task scheduling to minimize energy consumption, it contributes to prolonging the operation of electric vehicles.

Sequential Offloading: Tasks are offloaded sequentially rather than all at once, which minimizes the time discrepancy between scheduling and task execution. This sequential approach aids in efficiently managing computational load.

3.3 Proposed System Design

In this project work, there are four modules and each module has specific functions, they are:

1. Admin Module
2. PLC Module
3. Server Module

3.3.1 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 a entity in the cloud assisted network who will add plc and servers.

3.3.2 PLC Module

Each PLC continuously queries the state of the IO devices and check status of servers who are busy and waiting to accept request. 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 application and check various servers who are busy and waiting to accept request based on their status plc will upload data and sent to servers and get response form severs. PLC can cancel request and upload new task and check response from sever

3.3.3 Server Module

Using this module task manager will login with 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.

3.4 Architecture

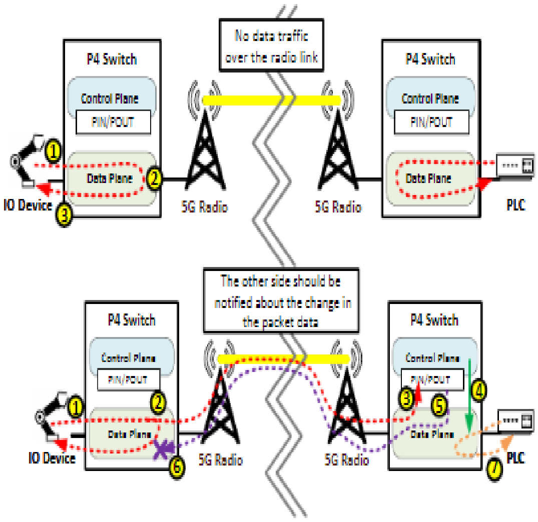
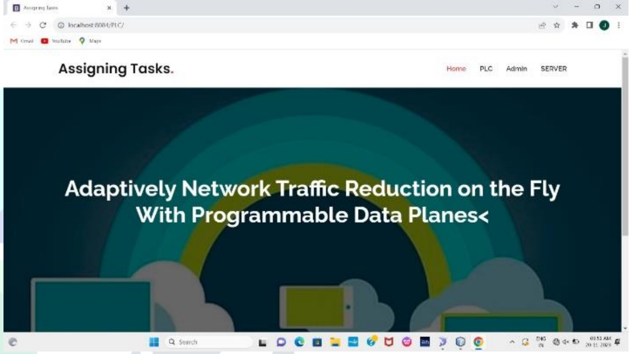
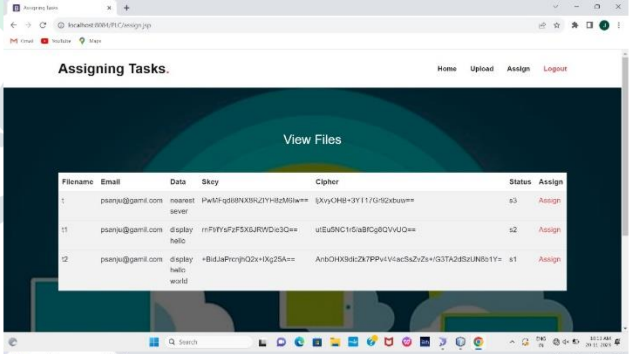


Fig 1: System Architecture

4. RESULT SCREEN SHOTS

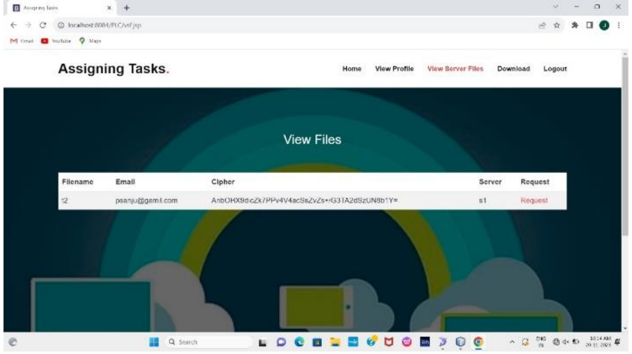


Adaptively Network Traffic Reduction on the Fly With Programmable Data Planes<



View Files

Filename	Email	Data	Skey	Cipher	Status	Assign
t	psanju@gamil.com	nearest sever	PwMFqd88NX8RZIYH8zM6lw==	IjXvyOHB+3YT17G/92xbuw==	s3	Assign
t1	psanju@gamil.com	display hello	rnFt/fYsFzF5X6JRWDie3Q==	utEu5NC1r5/aBfCg8QVvUQ==	s2	Assign
t2	psanju@gamil.com	display hello world	+BidJaPrcnjhQ2x+IXg25A==	AnbOHX9dicZk7PPv4V4acSsZvZs+/G3TA2dSzUN8b1Y=	s1	Assign



View Files

Filename	Email	Cipher	Server	Request
t2	psanju@gamil.com	AnbOHX9dicZk7PPv4V4acSsZvZs+/G3TA2dSzUN8b1Y=	s1	Request

5. 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. In proposed system we are using three techniques to encrypt data for security purpose which is shown on cloud environment. As in future scope a multi-dimensional application can be developed where every time when user uploads data user can select what type of encryption technique, he can use like 2 or three methods based on that each file will have new way of technique.

6. REFERENCES

[1] R. Pigan and M. Metter, Automating with PROFINET: Industrial communication based on Industrial Ethernet. John Wiley & Sons, 2008.
[2] C. Györgyi, K. Kecskeméti, P. Vörös, G. Szabó, and S. Laki, “In-network solution for network traffic reduction in industrial data communication,” Netsoft, 2021.
[3] M. Karakus and A. Durresi, “A survey: Control plane scalability issues and approaches in software-defined networking (sdn),” Computer Networks, vol. 112, pp. 279–293, 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, no. 3, p. 87–95, Jul. 2014. [Online]. Available: https://doi.org/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 2007 IEEE Int. Conference on Microelectronic Systems Education (MSE’07), 2007, pp. 160–161.
[6] T. Kobzan, I. Blöcher, M. Hendel, S. Althoff, A. Gerhard, S. Schriegel, and J. Jasperneite, “Configuration solution for tsn-based industrial networks utilizing sdn and opc ua,” in 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1. IEEE, 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 2016 IEEE World Conference on Factory Communication Systems (WFCS). IEEE, 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 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS). IEEE, 2017, pp. 1–7.
[9] M. Ehrlich, D. Krummacker, C. Fischer, R. Guillaume, S. S. P. Olaya, A. Frimpong, H. de Meer, M. Wollschlaeger, H. D. Schotten, and J. Jasperneite, “Software-defined networking as an enabler for future industrial network management,” in 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1. IEEE, 2018, pp. 1109–1112.
[10] C. Kim, A. Sivaraman, N. Katta, A. Bas, A. Dixit, and L. J. Wobker, “In-band network telemetry via programmable dataplanes,” in ACM SIGCOMM, 2015.
[11] X. Jin, X. Li, H. Zhang, R. Soulé, J. Lee, N. Foster, C. Kim, and I. Stoica, “Netcache: Balancing key-value stores with fast in-network caching,” in Proceedings of the 26th Symposium on Operating Systems Principles, 2017, pp. 121–136.
[12] A. Bremler-Barr, D. Hay, I. Moyal, and L. Schiff, “Load balancing memcached traffic using software defined networking,” in 2017 IFIP Networking Conference (IFIP Networking) and Workshops. IEEE, 2017, pp. 1–9.
[13] M. Zhang, G. Li, X. Kong, C. Liu, M. Xu, G. Gu, and J. Wu, “Nethcf: Filtering spoofed ip traffic with programmable switches,” IEEE Transactions on Dependable and Secure Computing, 2022.
[14] S. Laki, P. Vörös, and F. Fejes, “Towards an aqm evaluation testbed with p4 and dpdk,” in Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos, 2019, pp. 148–150.
[15] J. Rüth, R. Glebke, K. Wehrle, V. Causevic, and S. Hirche, “Towards innetwork industrial feedback control,” in Proceedings of the 2018 Morning Workshop on In-Network Computing, 2018, pp. 14–19.
[16] F. E. R. Cesen, L. Csikor, C. Recalde, C. E. Rothenberg, and G. Pongrácz, “Towards low latency

industrial robot control in programmable data planes," in 2020 6th IEEE Conference on Network Softwarization (NetSoft). IEEE, 2020, pp. 165–169.

[17] T. Mai, H. Yao, S. Guo, and Y. Liu, "In-network computing powered mobile edge: Toward high performance industrial iot," IEEE Network, 2020.

[18] L. Wüsteney, M. Menth, R. Hummen, and T. Heer, "Impact of packet filtering on time-sensitive networking traffic," in 17th IEEE International Conference on Factory Communication Systems, 2021.