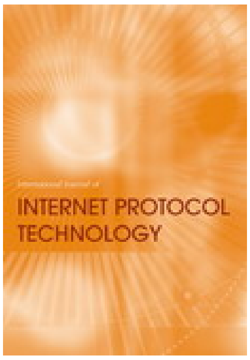


International Journal of Internet Protocol Technology

ISSN online: 1743-8217 - ISSN print: 1743-8209
https://www.inderscience.com/ijipt

Improved network performance in CPS communication with distributed IPC mechanisms of recursive internetworking architecture (RINA)

Bhushana Samyuel Neelam, Benjamin A. Shimray

DOI: 10.1504/IJIPT.2023.10045826

Article History:

Received:	28 November 2020
Last revised:	13 October 2021
Accepted:	16 October 2021
Published online:	23 March 2023

Copyright © 2023 Inderscience Enterprises Ltd.

Improved network performance in CPS communication with distributed IPC mechanisms of recursive internetworking architecture (RINA)

Bhushana Samyuel Neelam* and
Benjamin A. Shimray

Department of Electrical Engineering,
National Institute of Technology (Manipur),
Imphal, Manipur, India
Email: samsneelam@gmail.com
Email: benjaminshimray@gmail.com
*Corresponding author

Abstract: The network convenience provided by TCP/IP networks in Cyber-Physical Systems (CPS) communication redefined them as intelligent real-time systems. Along with convenience and intelligence, it also brought various network concerns into CPS communication. Some of the concerns include increased response times due to communication overheads, unnecessary delays in packet forwarding due to computational overheads, unwanted cyber threats in critical CPS infrastructures like energy, oil, gas, etc. As future networks are focused on achieving an optimum networking solution, Recursive Internetworking Architecture (RINA) based on distributed IPC mechanism seems to be a promising solution. The proposed work discussed extending distributed IPC mechanisms to the client-server communication model of CPS to extract improved and consistent response times compared to TCP/IP. A CPS client-server model is developed on RINA and TCP/IP networks with connection-oriented protocols. Response times are measured for controlling the actuators, sensing the sensors' data. The comparative analysis demonstrates that RINA reduced response times by almost 50% to TCP/IP and provided more consistency than the TCP/IP.

Keywords: future network architectures; distributed IPC; CPS; latency; RINA; response times; communication overheads; shared-state transfer; encapsulation; encapsulation overheads; decapsulation; process control; data acquisition.

Reference to this paper should be made as follows: Neelam, B.S. and Shimray, B.A. (2023) 'Improved network performance in CPS communication with distributed IPC mechanisms of recursive internetworking architecture (RINA)', *Int. J. Internet Protocol Technology*, Vol. 16, No. 1, pp.68–74.

Biographical notes: Bhushana Samyuel Neelam received his MTech degree in Electrical Power Engineering from JNTU College of Engineering, Hyderabad, India. He worked in the Design and Development of a Custom Network Stack for Mission-Critical Applications. Currently, He is pursuing PhD degree in the Department of Electrical Engineering at the National Institute of Technology, Manipur, India. His research areas include network programming, RINA in real-time applications, industrial utilities and cyber resiliency.

Benjamin A. Shimray received his PhD and MTech degree from the National Institute of Technology Manipur (2018), Imphal and The National Institute of Engineering, Mysore (2010), respectively. Since 2011, he has been working as a Faculty Member in the Department of Electrical Engineering, National Institute of Technology Manipur. He is also a Member of Various Professional Organisation such as IEEE, IEI and IAENG. His area of interest is application of soft computing techniques to renewable energy planning and management, control system applications, and in recent years in cyber vulnerabilities and system resilience.

1 Introduction

A Cyber-Physical System (CPS) is a combination of computing and communication facilities with physical systems. These physical systems include real-time applications like industrial utilities, health care systems, energy systems, military systems, etc. (Rajkumar et al., 2017). As technology is evolving further, the concept of CPS is extended to many technologies like Internet of Things (IoT), Wireless Sensor Networks (WSN), Systems of Systems (SoS), Machine to Machine (M2M) and cloud, etc., which associate with the physical world (Gunes et al., 2014). The present-day CPS is of

Copyright © 2023 Inderscience Enterprises Ltd.

the fourth generation and is termed Industry 4.0. Most of the CPS is integrated with the communication network to harness the advantage of connectivity. Various communication enabling technologies integrated Industry 4.0 type CPS with the communication networks (Yadav and Paul, 2020). Porting CPS onto TCP/IP networks brought advantages and many limitations to CPS utilities. Many new security protocols were designed and deployed to mitigate the limitations. As packet forwarding in TCP/IP is based on data encapsulation (Perkins, 1996; Kent, 1998), these add-on protocols caused communication overheads, resulting in increased response times and delays in process control and data acquisition (He et al., 2016).

In this context, a novel network architecture named Recursive Internetworking Architecture (RINA) based on IPC mechanism, proposed by Day (2007) found to have achieved promising features. It simplified data transfer with the help of distributed IPC with security as a by-product (Boddapati et al., 2012). The fundamental component of the RINA model is the *distributed IPC facility* (DIF) which contains all distributed IPC mechanisms responsible for data transfer, data transfer control, data confidentiality, data integrity, and routing. DIF shall form the layer in the RINA model and is generic in its functionality. In RINA, there are no specific layers like TCP/IP but generic layers that are customisable. These layers, i.e., DIF, shall be repeated depending on the scope of the network. Packet forwarding in this architecture is purely based on distributed IPC mechanism and avoids data encapsulation mechanisms of TCP/IP.

As mentioned by Jiang et al. (2000), Krug and O'Nils (2019) and Vázquez et al. (2006), data encapsulation due to the layer-wise protocol headers causes communication overheads in the IoT/CPS communication. But RINA restricted the number of protocols by designing a generalised layer called DIF, which offers distributed IPC mechanisms to transfer data. The novelty of the proposed work lies in investigating the impact of distributed IPC mechanisms on communication overheads in CPS utility. It verifies whether RINA could make a difference in CPS communication with its clean state design and shared state IPC mechanisms in CPS communication. As per our knowledge, this method is novel as it explores the impact of shared state IPC mechanisms in data transfer. This work is limited to CPS utilities backed up by TCP/IP or Internet technology in the closed network configuration, and other versions may include other network architectures also.

The proposed work compares the response times for process control and data acquisition in both the networks, i.e., RINA and TCP/IP, to verify the impact of distributed IPC mechanisms in reducing communication overheads. The proposed model develops a small-scale hardware CPS prototype that operates like the client-server model in both networks, i.e., RINA and TCP/IP. The hardware prototype consists of Arduino processors and a couple of actuators and sensors. The software utility for the proposed model is developed for both the networks, i.e., RINA and TCP/IP. The CPS server application measures the response times for each functionality of the process control and data acquisition. A comparative analysis is provided to verify the optimum RTT among both networks. Results confirm that RINA offers improved RTT when compared to its predecessor. The motivation behind this work is to explore any possibilities of overcoming delays caused by encapsulation overheads, as mentioned in Perkins (1996). It also explores the advantages of distributed IPC mechanisms of RINA in CPS communication for consistent and optimum response times.

2 Related work

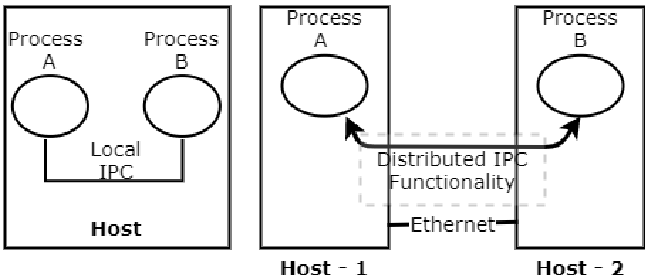
There has been a lot of work undergone for reducing the encapsulation overheads. Whenever a layer on the sending side receives data in the form of a payload from its upper layer, it adds its header as meta-data and sends it to the next layer. This process is called encapsulation (Perkins, 1996). Whenever a layer on the receiving side receives data from its lower layer, it removes the corresponding header and transfers the data part to its upper layer. This process is called decapsulation. Perkins found that this is causing communication overheads and proposed a method to minimise the IP header to reduce the communication overhead (Perkins, 1996). Woodard (n.d.) investigated the impact of encapsulation with various protocols using authenticated tools and confirmed that they are causing communication overheads and thereby reducing the network performance. Jiang et al. (2000) presented a flexible IP encapsulation method for IP over ATM to reduce communication overheads caused by multiple encapsulations in short and moderate data, which refers to the CPS communication. Vazquez et al. (2006) mentioned that it is crucial to reduce the encapsulation overheads even in MPLS networks. Various methods and protocols were employed to mitigate the effects of encapsulation in data communication, particularly for modern CPS utilities (Guenender et al., 2015; Pereira et al., 2014). Guenender et al. (2015) proposed a software method based on network virtualisation to mitigate the encapsulation overheads. It employs Software-Defined Networks (SDN) model to overcome encapsulation overheads. Even though SDN offers programmability, it suffers ossification and programmability constraints (Azzouni et al., 2017; Leon et al., 2015). The proposed method employs fully programmable network architecture RINA to investigate the impact of its network on the response times of process control and data acquisition.

3 Recursive internetworking architecture (RINA)

Recursive Internetworking Architecture (RINA) is a clean state network architecture designed and developed on the contributions of network researchers from a couple of decades. It originates from the principle that 'networking is distributed Inter-Process Communication (IPC) and IPC alone' (Day, 2007; Day et al., 2008). It employs the local IPC concept to develop distributed IPC services across the remote applications, as shown in Figure 1.

Distributed IPC services are implemented in a component called IPC Process (IPCP), and it forms the basic building block of RINA Communication. Applications make use of IPCP mechanisms to transfer the data across the hosts. The mechanism of the IPC process is internal, and the application is connected to it through a local port. Each IPCP is capable of data transfer, error flow mechanism and routing functionality. The combination of IPCPs that perform data transfer between remote applications shall form a layer called Distributed IPC Facility (DIF) (Grasa et al., 2017). A DIF is similar to the layer in TCP/IP. But every DIF offers the same functionality, which can be customised according to the network requirement. In contrast, TCP/IP offers layer-specific functionality, and it performs a fixed set of mechanisms per layer.

Figure 1 Distributed IPC mechanism



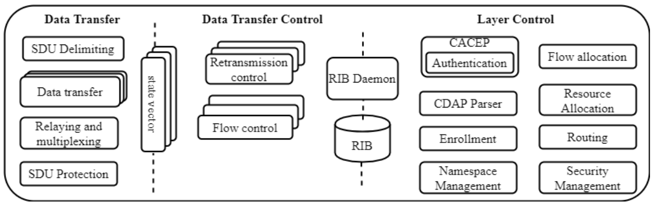
Applications must join the DIF through an IPCP to communicate with each other, and IPCP allots a port number on-demand, which is internal to the IPCP/DIF and acts as an interface between the application and IPCP. This is in contrast to the public and known ports of TCP/IP architecture. Moreover, ports are overloaded as connection endpoints in TCP/IP and have become part of connection management. But by making ports internal and of local significance, RINA disconnected the ports from connection management and thus enhanced the security (Boddapati et al., 2012; Ramezanifarkhani and Teymoori, 2018; Samyuel and Shimray, 2020).

3.1 *Functionality of IPCP*

The functionality of IPCP includes data transfer, error flow mechanism, and layer management, as shown in Figure 2. These functionalities are driven by two major protocols called Error Flow Control Protocol (EFCP) and Common Distributed Application Protocol (CDAP). EFCP is meant for enabling and controlling the transfer of data. It is further divided into two minor protocols, i.e., Data Transfer Protocol (DTP) and Data Transfer Control Protocol (DTCP). DTP takes care of the data transfer module, and DTCP takes care of the data transfer control module. CDAP is meant for layer management functionalities like connection management, routing management, maintenance of shared state components with Resource Information Base (RIB). It coordinates the services of IPCP across the layers of DIF (Grasa et al., 2016; Vrijders et al., 2014).

The functionality of the DIF remains the same irrespective of its position. But the functionality can be programmed to suit the scope of the DIF. This feature enables the network designer to configure mechanisms like authentication and protection of data, routing control and data transfer control using QoS according to the requirement.

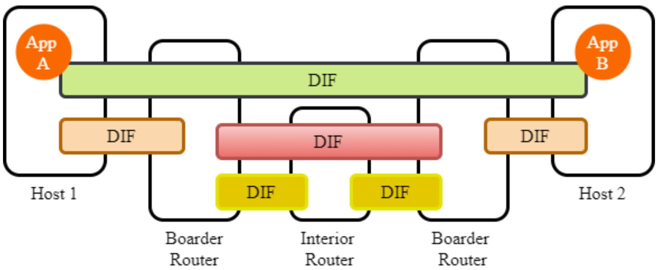
Figure 2 Functionality of IPCP



3.2 *Programmable DIF*

As the scope of application increases, DIF shall be repeated in recursion to extend its services across the applications (Vrijders et al., 2014) as shown in Figure 3. To customise the functionality of DIF across different scopes, RINA adopted the scheme of separation of mechanism from policy (Grasa et al., 2016). The separation mechanism differentiates the fixed components and variable components of the IPCP. The fixed components represent the service of the IPCP and are called mechanisms. At the same time, the variable component represents the timing of service and is called policy. This differentiation resulted in generic protocols with a wide variety of functionalities that can be customised through policies. These policies remain the same and can be used by DIF across its rank and scope of application. This claims RINA as a reliable network architecture with fewer protocols and several policies to suit the requirements of network (Grasa et al., 2016; Vrijders et al., 2014).

Figure 3 Recursion of DIF in RINA



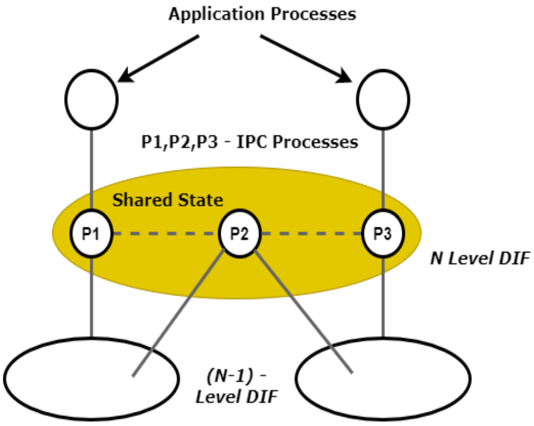
3.3 *Communication flow – TCP/IP vs. RINA*

Besides encapsulation and decapsulation, TCP/IP uses three-way handshaking for connection-oriented flow using TCP. Three-way handshaking is based on hybrid state design as explained by Braden (1994) and Gursun et al. (2010). In RINA, data communication is not layer-wise but a shared state mechanism which is nothing but a distributed IPC mechanism, as shown in Figure 4.

RINA eliminated three-way handshaking for its connection-oriented flows. Instead, it used Watson's delta-t protocol (Fletcher and Watson, 1978), which is based on soft-state design to maintain connection reliability (Gursun et al., 2010; Tarzan et al., 2019). According to Watson, the bounding of three timers is necessary and appropriate to establish a reliable connection. It simplified the network implementation by eliminating SYN and FIN as in TCP. RINA also decoupled port allocation from data synchronisation and avoided overloading the semantics of connection end-point-id (Boddapati et al., 2012).

Proposed work aimed to extract the advantages of RINA's shared state/distributed IPC-based data forwarding compared to IP encapsulation in TCP/IP in connection-oriented communication.

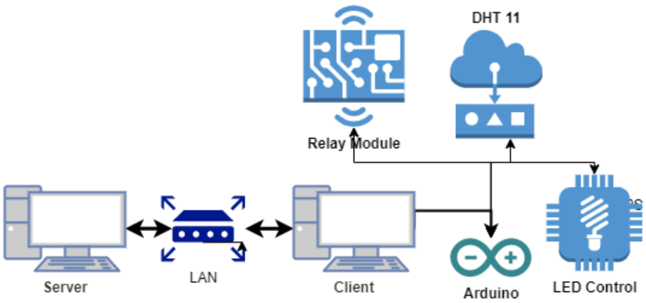
Figure 4 Data transfer in RINA



4 Proposed CPS model

The proposed model developed a small-scale CPS hardware model in the client-server mode for both RINA and TCP/IP networks in a closed (LAN) environment, as shown in Figure 5. The functionality of the proposed CPS model is to control actuators like relays and to acquire data from sensors.

Figure 5 Proposed CPS client-server model



The hardware implementation of the CPS client is shown in Figure 6. CPS Client consists of an Arduino board, actuators like a 4-module relay, a led, and a Digital Humidity Temperature (DHT11) sensor. The server application provides the interface for process control, monitoring acquired data from the client and RTT of process control and data acquisition. This HMI is developed using Qt IDE. The client's functionality includes responding to the process control requests from the server by controlling actuators through Arduino and senses data from the DHT11 sensor on Arduino to send it back to the server.

4.1 Connection-oriented communication

In TCP-based client-server applications, connection-oriented communication is achieved with the help of POSIX socket API in three-way handshake mode. In RINA, it should be programmed in the DIF configuration file, as shown in Figure 7. RINA uses its socket API developed by Vrijders et al. (2014) to develop its CPS applications.

Figure 6 Hardware implementation of the proposed CPS client

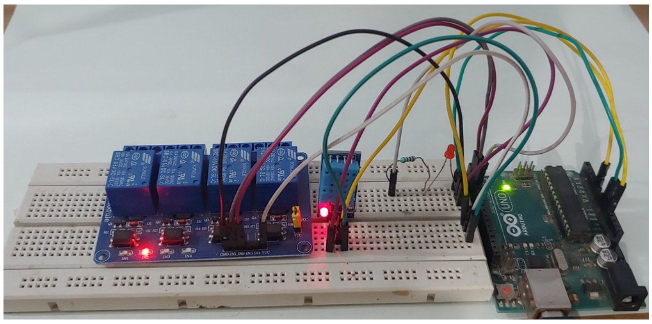


Figure 7 Configuration of the connection-oriented flow

```
},
"id": 2,
"maxAllowableGap": 0,
"name": "reliablewithflowcontrol",
"orderedDelivery": true,
"partialDelivery": false
```

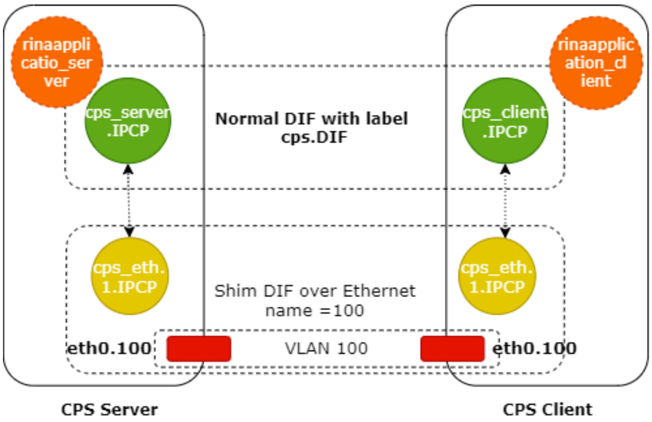
The sequence of establishing a communication flow in RINA can be understood from the following steps, as explained in Samyuel and Shimray (2020):

- *Step 1*: Configuration of the network devices
- *Step 2*: Initiation of IPC Manager with a configuration file
    a) Creation of IPCP's
    b) Creation of instance of DIF according to the configured templates
    c) Attaches IPCP to the instance of DIF
    d) Registers normal DIF to shim DIF
- *Step 3*: Enrolment and creation of distributed IPC facility between client and server.
- *Step 4*: Performing client-server communication with RINA-based socket API

The distributed IPC mechanism of the proposed model is shown in Figure 8.

It shows a DIF named *cps.DIF* comprising of IPC processes *cps_server.IPCP* and *cps_client.IPCP* on server and client, respectively. This *cps.DIF* is responsible for achieving client-server communication between CPS nodes. This is connected to the lower level DIF, which is called shim DIF. A shim DIF is the collection of shim IPCP's. Shim IPCP is the lowest level of IPCP which acts as a wrapper around legacy Ethernet to transfer data. A VLAN device called *eth0.100* is created in each node and is assigned to VLAN tag *100*. RINA uses this VLAN device *eth0.100* as an interface for transferring data. Applications will perform data transfer by connecting to the local IPCP that belongs to (N)-DIF, i.e., *cps.DIF* in each node. As explained earlier, the CPS server must be communicating with the client to control the actuators and acquire data from the client on *cps.DIF*.

Figure 8 Network model of RINA-CPS



5 Results

CPS application is executed for a couple of hours in each network, i.e., TCP/IP and RINA. The following performance indices are observed

- Round Trip Times (RTT)
- Standard deviation in RTT / network jitter

Process control operations of led and relay and data acquisition are observed around 500 times each. Then RTT for each function is calculated and displayed on the HMI of the server. A comparative analysis of RTT in control and data acquisition events is provided to evaluate distributed IPC's advantage. The standard deviation in RTT offers the consistency of RTT, and the lesser the standard deviation, the higher the consistency of RTT. A comparison of standard deviation between RINA and TCP/IP is provided.

5.1 *RTT in process control*

Figure 9 shows the comparison of RTT in process control operations of led and relay between TCP/IP and RINA.

Figure 9 Comparison of RTT in process control

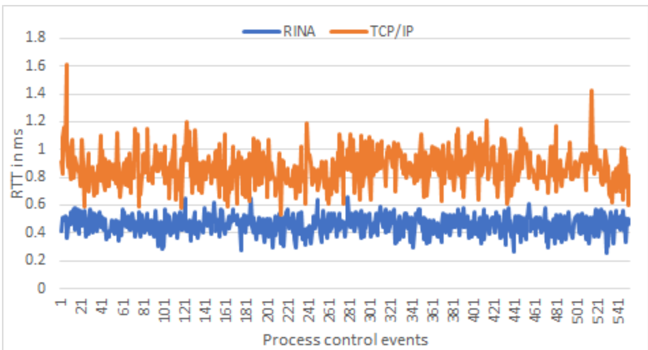
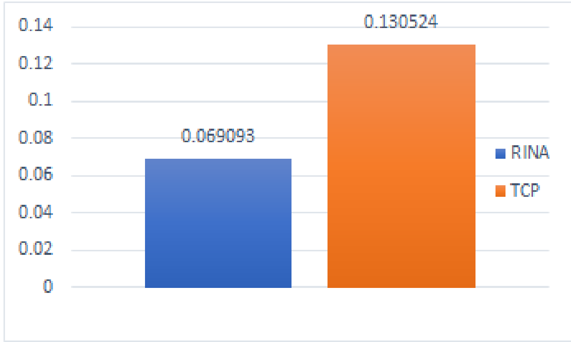


Figure 10 Standard deviation of RTT in process control



It is observed from Figure 9 that RTT is reduced a lot in RINA when compared to the TCP/IP. Figure 10 shows that the standard deviation is reduced to 0.069093 to 0.130524 in TCP/IP. That means the standard deviation is improved by 47% in RINA when compared to TCP/IP. It proves enhanced response times and consistency in process control communication of CPS in the RINA network model.

5.2 *RTT data acquisition*

Figures 11 and 12 show a comparative analysis of RTT in data acquisition processes for temperature and humidity. It is observed from Figure 11 that RINA's RTT in data acquisition is slightly improved than the TCP/IP. But Figure 12 demonstrates that the standard deviation of RTT reduced to 0.064228 in RINA from 0.143218 of TCP/IP. That means the standard deviation is improved by 55% in RINA when compared to TCP/IP. It proves that response time in RINA is more consistent than TCP/IP, and this proves that network performance is improved with RINA in data acquisition also.

Figure 11 Comparison of RTT in data acquisition

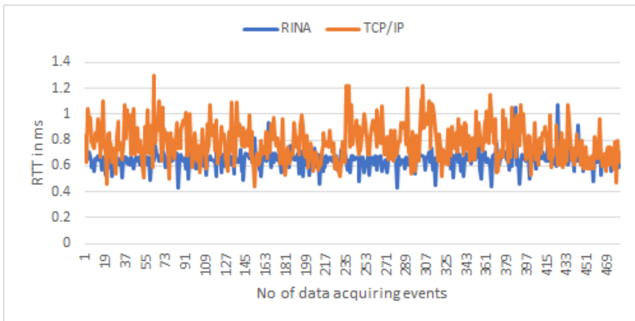
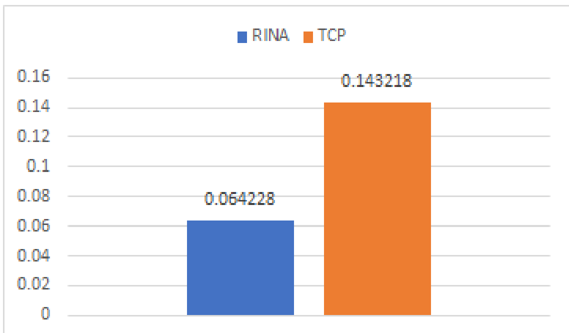


Figure 12 Standard deviation of RTT in data acquisition



Results affirm that network improvement is possible with RINA in CPS process control and data acquisition communication. This improvement is attributed to the shared state/distributed IPC mechanism in RINA. Thus, RINA offers improved RTT in process control and data acquisition than TCP/IP.

6 Conclusion

The proposed model extracted improved network performance of CPS communication in RINA when compared to TCP/IP. A CPS model in client-server mode is developed in RINA as well as TCP/IP. The process control and data acquisition functionality of CPS is tested in the client-server model. Then, the CPS server application measured Round Trip Time (RTT) for each controlling or data acquisition event. The standard deviation of RTT is calculated and compared between both networks. Results emphasised that the RINA model has improved response times with 47% and 55% consistency in process control and data acquisition, respectively, compared to TCP/IP. This work proved the enhanced network performance of RINA with its distributed IPC mechanisms. We are currently investigating RTT variations in the presence of security algorithms like authentication and encryption. The limitations include the availability of its UNIX-based network stack and a unified testing facility where its performance can be tested with authenticity.

References

Azzouni, A., Trang, N.T.M., Boutaba, R. and Pujolle, G. (2017) 'Limitations of openflow topology discovery protocol', *Proceedings of the 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, IEEE, Budva, pp.9–11. Doi: 10.1109/MedHocNet.2017.8001642.

Boddapati, G., Day, J., Matta, I. and Chitkushev, L. (2012) 'Assessing the security of a clean-slate Internet architecture: Security as byproduct of decoupling different concerns', *Proceedings of the International Conference on Network Protocols*. Doi: 10.1109/ICNP.2012.6459947.

Braden, R. (1994) T/TCP – *TCP Extensions for Transactions Functional Specification*, Network Working Group.

Day, J. (2007) *Patterns in Network Architecture*, Pearson Education India.

Day, J., Matta, I., Mattar, K. and College, M. (2008) 'Networking is IPC: a guiding principle to a better internet', *Proceedings of the ACM Conference on Emerging Network Experiment and Technology*, Spain. Doi: 10.1145/1544012.

Fletcher, J.G. and Watson, R.W. (1978) 'Mechanisms for a reliable timer-based protocol', *Computer Networks*, Vol. 2, pp.271–290. Doi: 10.1016/0376-5075(78)90006-5.

Grasa, E., Bergesio, L., Tarzan, M., Trouva, E., Gaston, B., Salvestrini, F., Maffione, V., Carrozzo, G., Staessens, D., Vrijders, S., Colle, D., Chappell, A., Day, J. and Chitkushev, L. (2017) 'Recursive internetwork architecture, investigating RINA as an alternative to TCP/IP (IRATI)', *Recer. (Dipòsit la Recer. Catalunya)*, pp.1–794. Doi: 10.13052/RP-9788793519114.

Grasa, E., Rysavy, O., Lichtner, O., Asgari, H., Day, J. and Chitkushev, L. (2016) 'From protecting protocols to layers: designing, implementing and experimenting with security policies in RINA', *Proceedings of the IEEE International Conference on Communication*, pp.1–7. Doi: 10.1109/ICC.2016.7510780.

Guenender, S., Barabash, K., Ben-Itzhak, Y., Levin, A., Raichstein, E. and Schour, L. (2015) 'NoEncap: overlay network virtualization with no encapsulation overheads', *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, pp.1–7. Doi: 10.1145/2774993.2775003.

Gunes, V., Peter, S., Givargis, T. and Vahid, F. (2014) 'A survey on concepts, applications, and challenges in cyber-physical systems', *KSII Transactions on Internet and Information Systems*, Vol. 8, pp.4242–4268. Doi: 10.3837/tiis.2014.12.001.

Gursun, G., Matta, I. and Mattar, K. (2010) 'Revisiting a soft-state approach to managing reliable transport connections', *Proceedings of the 8th International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports*.

He, H., Maple, C., Watson, T., Tiwari, A., Mehnen, J., Jin, Y. and Gabrys, B. (2016) 'The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence', *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp.1015–1021. Doi: 10.1109/CEC.2016.7743900.

Jiang, S., Ding, Q. and Jin, M. (2000) 'Flexible IP encapsulation for IP over ATM with ATM shortcuts', *Proceedings of the IEEE International Conference on Networks*, pp.238–242. Doi: 10.1109/ICON.2000.875795.

Kent, S. (1998) *IP Encapsulating Security Payload (ESP)*, Network Working Group.

Krug, S. and O'Nils, M. (2019) 'Modeling and comparison of delay and energy cost of IoT data transfers', *IEEE Access*, Vol. 7, pp.58654–58675. Doi: 10.1109/ACCESS.2019.2913703.

Leon, S., Perello, J., Grasa, E., Careglio, D. and Spadaro, S. (2015) 'On the benefits of RINA over programmable optical networks for dynamic and smart resource management', *Proceedings of the International Conference on Transparent Optical Networks*, pp.2–5. Doi: 10.1109/ICTON.2015.7193400.

Pereira, P.P., Eliasson, J. and Delsing, J. (2014) 'An authentication and access control framework for CoAP-based internet of things', *Proceedings of the 40th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, USA, pp.5293–5299. Doi: 10.1109/IECON.2014.7049308.

Perkins, C. (1996) *Minimal Encapsulation within IP*, RFC Editor, USA. Doi: 10.17487/RFC2004.

Rajkumar, R., Lee, I., Sha, L. and Stankovic, J. (2017) 'Cyber-physical systems: the next computing revolution', *Cybernetics and systems analysis*, Vol. 53, pp.821–834. Doi: 10.1007/s10559-017-9984-9.

Ramezanifarkhani, T. and Teymoori, P. (2018) 'Securing the internet of things with recursive internetwork architecture (RINA)', Proceedings of the *International Conference on Computing and Network Communications*, pp.188–194. Doi: 10.1109/ICCNC.2018.8390263.

Samyuel, N.B. and Shimray, B.A. (2020) 'Securing IoT device communication against network flow attacks with recursive internetworking architecture (RINA)', *ICT Express*, pp.1–5. Doi: 10.1016/j.icte.2020.08.001.

Tarzan, M., Bergesio, L. and Grasa, E. (2019) 'Error and flow control protocol (EFCP) design and implementation: a data transfer protocol for the recursive inter network architecture', *Proceedings of the 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pp.66–71. Doi: 10.1109/ICIN.2019.8685905.

Vázquez, E., Álvarez-Campana, M., García, A.B. and Hernández, A. (2006) 'Efficiency and quality of service issues in MPLS transport for the UMTS access network', *Computer on Communication*, Vol. 29, pp.820–826. Doi: 10.1016/J.COMCOM.2005.08.004.

Vrijders, S., Staessens, D., Colle, D., Salvestrini, F., Grasa, E., Tarzan, M. and Bergesio, L. (2014) 'Prototyping the recursive internet architecture: the IRATI project approach', *IEEE Network*, Vol. 28, pp.20–25. Doi: 10.1109/MNET.2014.6786609.

Woodard, J. (n.d.) *Overhead – Bane of Layered Network Design? Steven Institute of Technology*, Hoboken, NJ.

Yadav, G. and Paul, K. (2020) 'Architecture and security of Scada systems: a review', *arXiv:2001.02925*.