Novel Design of Ubiquitous Data-Centric Automation and Control Architecture

Sadiq M. Sait, Center for Communications and Information Technology Research, and Computer Engineering Department, King Fahad University of Petroleum and Minerals, Dhahran, 31261, Saudi Arabia, Sadiq@kfupm.edu.sa

Ghalib A. Al-Hashim, Saudi Aramco, Dhahran, 31311, Saudi Arabia, ghalib.hashim@aramco.com

Keywords: Automation; Programmable Logic Controller; Distributed Control System; Data Distribution Service Middleware; Obsolescence; Utilization; Collaboration.

ABSTRACT

The development of process automation systems has been evolving to raise productivity and enhance plant operation. Their fundamental architecture has advanced from large centralized systems to highly distributed systems; but suffering from three major disadvantages. First is the requirement of extensive premature capital investments for sustaining the systems life cycle. Secondly are the inherent architectural constraints for achieving full utilization of the controllers' resources. Finally is the inefficient distribution of real-time process data for achieving maximum yield of processing facilities. This paper presents a paradigm shift from the traditional distributed architecture to a ubiquitous data-centric architecture. This is based on distributed autonomous process interface systems, fault-tolerant real-time data distribution service middleware, and virtually clustered servers. Results of detailed performance analysis confirm the viability of the new proposal as the basis for designing an evergreen collaborative automation platform that provides flexibility and economy of scale for maximum systems' utilization and heterogeneous scalability and compatibility across multiple vendors.

1. INTRODUCTION

For the past forty years, the development of process automation systems including programmable logic controllers (PLC) and distributed control systems (DCS) has been evolving to ensure effective control of all disparate equipment within processing facilities in a cohesive fashion for improved safety, higher production rates, more efficient use of materials and energy, and consistent product quality [14]. Their fundamental architecture has advanced from large centralized system with all control hardware and input/output (I/O) racks mounted in large cabinets located in the central control room (CCR) to highly distributed systems hosted in process interface buildings (PIBs) closer to the processing facilities in order to minimize the cost of electric signal cabling [4].

Although process automation technology has advanced rapidly since the mid-1980s, the latest systems still follow the traditional hierarchical structure: field control network layer, plant control network layer, and plant information network layer. The main function of process control applications at the lowest network layer is to monitor the operation of each part of the process, identifies unwanted changes and initiates any necessary corrective actions. At this layer, the conventional controllers are based on a monolithic architecture in which functionally distinguishable aspects such as the I/O racks, the main control module, and the control application, are not architecturally separate standard components but are all proprietary and interwoven. Further up the hierarchical structure, the control system at the middle network layer controls the overall production process and makes sure it continues to operate efficiently. Finally at the plant network layer, the advanced control applications ensure optimum performance of the processing units by moving the controllers' setting closer to the operating envelope's constraints, where typically the performance is higher [14].

Subsequently, this network-centric architecture has transformed to an application-centric architecture where the interaction among the heterogeneous process control applications across all network layers are implemented using client-server communication model. This communication model works very well for the conventional system architecture where there is a centralized server in each network layer. At the field control network layer, the process data is centralized within each associated controller. At the plant control network layer, the process data is centralized within a data server providing real-time process data to all proprietary nodes from the same system vendor at this layer. At the plant network layer, the process data is centralized within an object linking and embedding for process control (OPC) data server providing near real-time plant data to other nodes from different manufacturers at this layer [1].

The remainder of this paper is organized as follows. Section 2 establishes the motivation. An overview of related work is discussed in Section 3.The proposed architecture is described in Section 4. The performance analysis is detailed in Section 5. Finally, we conclude in Section 6.

2. MOTIVATION

From users' perspective in process industries, all automation solutions are suffering from three major disadvantages:

1) The application-centric architecture is not effective for integrating highly interacting heterogeneous process control applications across multiple network layers for exploiting processing facilities to achieve maximum yield of processing facilities. The computing power of the main control module is limited and not suitable for computing intensive control strategies. As a result, proprietary application modules are used for implementing advanced regulatory and multivariable control strategies at the plant control network layer. The plant control layer is proprietary and cannot accommodate any third party applications. Hence, the standard rigorous solutions for advanced process control have been implemented at the plant network layer utilizing near real-time process data provided by the OPC data server. Therefore, the information is being generated at multiple nodes and the client-server communication model is inefficient and precludes deterministic communications, since the client does not know when new information is available.
2) There are inherent architectural constraints for achieving full utilization of the controllers' resources. The controller monolithic architecture includes hardened built-in bond between the main control module and the associated I/O racks. The required composite scan time resolution of the controller for each process application depends on the size of logic memory utilized for the control application and the required number of I/O racks. Therefore, the controller's resource utilization is dependent on the process application rather than the available capacity.
3) The current proprietary process automation systems typically have limited useful lives measured by the competitive advantage they deliver. Therefore, there is a requirement for extensive premature capital investments, approximately 75% of the initial cost of ownership, to sustain the systems life cycle. This is due to frequent obsolescence challenges of proprietary controllers and associated I/O racks resulting from the shortage of critical third party subcomponents and/or the change in vendors' marketing strategies for increasing their market share.

3. RELATED WORK

Modern process automation engineering is a relatively new field of study that gained significant attention during the 20^{th} century with the advancement of computing technology including real-time middleware.

Middleware is a collection of technologies and services to enable the integration of subsystems and applications across an overall system. Most of the architectural models used in the development of middleware systems have evolved from point-to-point, client-server, and publish-subscribe communication models. Point-to-point is the simplest tightly coupled form of communication. TCP is a point-to-point network protocol designed in the 1970s. While it provides reliable, high bandwidth communication, TCP is cumbersome for systems with many communicating nodes [3]. To address the scalability issues of the Point-to-Point communication model, developers turned to the client-server communication model for centralized information and distributed applications, and many other paradigms are built upon it. However, if information is being generated at multiple nodes, client-server model is inefficient and precludes deterministic communications, since the client does not know when new information is available [13]. A solution to this limitation is to adopt publish-subscribe communication model. In this model, computer applications subscribe to data they need and publish data they want to share. Messages pass directly between the publisher and the subscribers, rather than moving into and out of a centralized server. This direct and simultaneous communication among a variety of nodes makes publish-subscribe network architecture the best choice for systems with complex time-critical data flows, even in the presence of unreliable delivery mechanisms [9].

One of the most important efforts to standardize publish-subscribe middleware is the development of data distribution service (DDS) specification by Object Management Group, Inc. (OMG). Data-centric publish-subscribe standard is the portion of the OMG DDS specification that addresses the specific needs of real-time data-critical applications and describes the fundamental concept supported by the design of the application programming interface. It focuses on the distribution of data between communicating applications, and provides several mechanisms that allow application developers to control how communication works and how the middleware handles resource limitations and error conditions. The communication is based on passing data of known types in named streams from publishers to subscribers [6,7]. The OMG DDS is the first open international middleware standards suitable for addressing data-centric publish-subscribe communications for real-time data-critical applications and embedded systems that require specific features including efficiency, determinism, flexibility delivery bandwidth, and fault-tolerant operation [5,10].

Efficiency: Automation systems require efficient data collection and delivery. Only minimal delays should be introduced into the critical data-transfer path. Publish-subscribe communication model is more efficient than client-server model in both latency and bandwidth for periodic data exchange [1].

Determinism: Automation applications also care about the determinism of delivering periodic data as well as the latency of delivering event data. Publish-subscribe implementations can provide configurable trade-offs between the deterministic delivery of new data and the reliable delivery of all data [3].

Flexibility Delivery Bandwidth: Typical control systems include both real-time and non-real-time nodes. The bandwidth requirements for these nodes are different. Data-centric publish-subscribe communication model allows subscribers for the same data to set individual limits on how fast data should be delivered to each subscriber [8].

Fault-Tolerant Operation: Automation applications are required to run in the presence of component failures. Publish-subscribe communication model is capable of supporting many-to-many connectivity with redundant publishers and subscribers. This feature is ideal for constructing high availability automation applications with redundant nodes and robust fault detection and handling services [11].

OMG also developed a standard wire protocol that allows DDS implementations from multiple vendors interoperate to avoid any DDS obsolescence challenges. Hence, the information published on a topic using a specific DDS implementation is consumable by one or more subscribers of other DDS versions [12].

4. PROPOSED ARCHITECTURE

In this section, we discuss the evolution of the proposed ubiquitous data-centric automation and control architecture. The three current challenges facing automation users in process industries are centered on a common root cause resulting from the inherent interwoven characteristics of the monolithic controller architecture. Decoupling the I/O racks physically and logically from the main control module provides the essential building block of a data-centric automation and control architecture for addressing all above challenges. The second building block is the utilization of distributed autonomous process interface systems in lieu of centralized I/O racks and virtual fault-tolerant computer servers in lieu of hardened embedded controllers. The third building block is the utilization of real-time OMG DDS middleware to provide an effective and efficient seamless integration layer between the virtual fault-tolerant control servers and the heterogeneous distributed autonomous process interface systems.

Figure 1 illustrates an example of a data-centric automation and control architecture. This architecture consists of four abstracted layers: process interface layer, control network layer, real-time data distribution service middleware layer, and collaborative automation layer.

The first layer includes the distributed process interface systems housed in the field junction boxes close to the associated process equipment. Each process interface system consists of required I/O modules and one DDS-enabled communication adapter with dual Ethernet communication ports. The distributed process interface systems are autonomous because they are physically self-contained and independent of the controllers. There are two types of junction boxes; master junction box and smart junction box. The smart junction box, located very close to the processing equipment requiring control, includes one process interface system with terminal strips to interface with wiring cables connecting the electrical signals to the associated instruments and field devices. The master junction box is similar to the smart junction box with an additional Ethernet switch for connecting the nearby smart junction boxes. The location of the master junction box is selected to be as close as possible to the center of large number of junction boxes in order to minimize communication cabling cost. The Ethernet switch uplink within each master junction box is a control segment extended from the local area control network in the CCR.

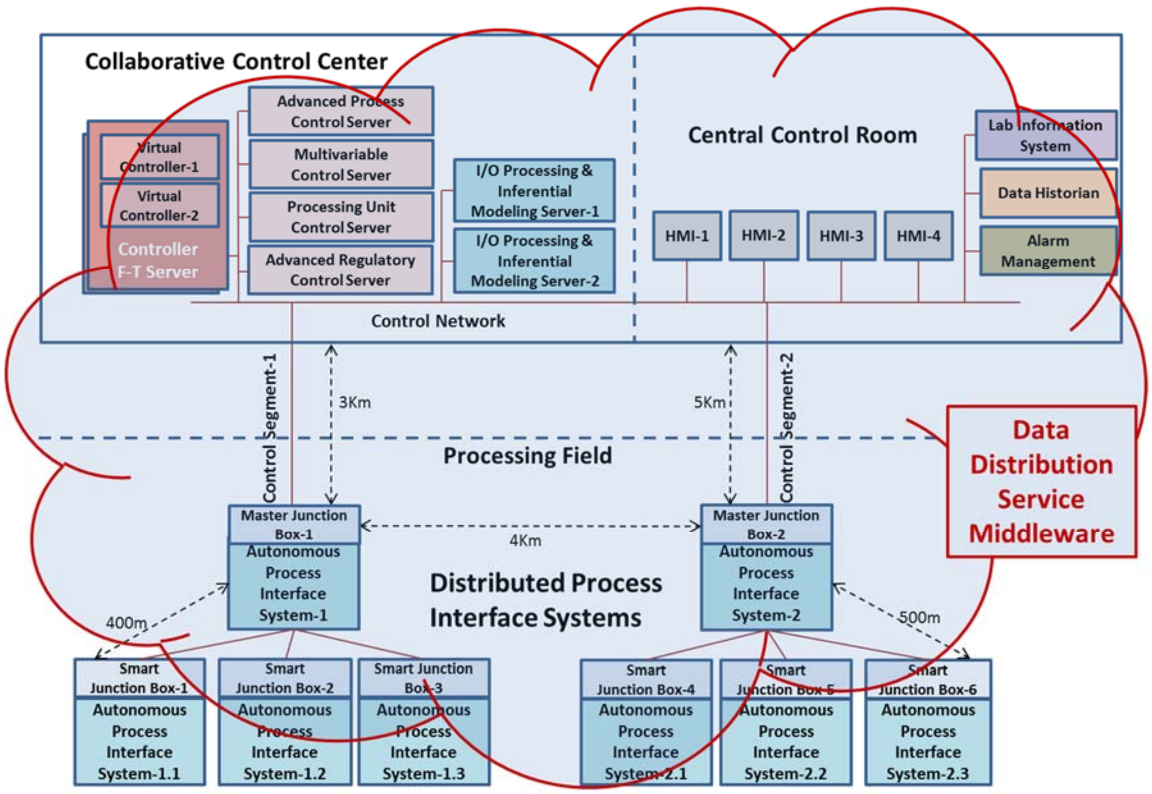

Figure 1 - Data-Centric Automation and Control Architecture

The second layer includes the hardware infrastructure for the Ethernet-based local area control network to interconnect the distributed process interface systems located at the field junction boxes, the computer servers located at the collaborative control center, and the operator interface consoles located at the central control room. Depending on the criticality of the automation application, the design of the local area control network can be simplex Ethernet network, parallel Ethernet network, or fault-tolerant Ethernet network. The simplex Ethernet network provides a single communication path from any source nodes to any destination nodes. The parallel Ethernet network is single fault-tolerant network architecture and provides two disjoint communication paths from any source nodes to any destination nodes. The fault-tolerant Ethernet network is multi fault-tolerant network architecture and provides four redundant communication paths from any source nodes to any destination nodes.

The third layer includes the communication software infrastructure for interconnecting the distributed process interface systems, the computer servers, and the operator interface consoles. It is based on the standard OMG real-time data-centric DDS middleware and distributed among all hardware nodes connected to the Ethernet-based control network infrastructure. The DDS middleware provides capabilities to provide communication among heterogeneous components, to extend its functionalities using open interface, to sustain system performance with higher loads in future, to recover from hardware or software failures, to provide security policies and mechanisms, to guarantee quality of service (QoS) for real-time control applications, and to cope with changes in the applications and/or users requirements.

The last layer includes the optimum collaborative process control environment based on virtually clustered servers for sequential and regulatory control, advanced regulatory control, multivariable control, unit-based process control, and plant-wide advanced process control to achieve maximum yield of all manufacturing facilities. The novel design of virtually clustered automation platform is also to overcome the inherent low utilization constraint, resulting from the traditional approach of installing one physical system per control application, and to maximize the total value of ownership.

The overall data-centric automation and control architecture is a paradigm shift in the design of automation control environment to provide inherent mechanisms for upgrades and/or partial replacement of any obsolete components without obligation for a complete system replacement throughout the expected life span of the processing facilities.

5. PERFORMANCE ANALYSIS

The new data-centric automation and control architecture has been evaluated empirically using software based simulation model to demonstrate its performance sustainability while growing in size based on the number of I/O signals. The focus of this empirical test is to validate the viability of using real-time DDS middleware to exchange required interaction traffic among the controllers and the autonomous process interface systems for safe and reliable operation of the processing facilities. The measuring performance criteria are the average latency and throughput. The communication test between a publisher and a subscriber is as follows. The I/O system is the publishing side and the controller is the subscribing side. The publishing side writes data, a total of 30 million data samples, to the middleware as fast as it can. Every time, after writing 1000 data samples to the middleware, it sends a special sample requesting an echo from the subscribing side. The publishing application publishes throughput data and at the same time it also subscribes to the latency echoes. The subscribing applications subscribe to the throughput data, in which the echo requests are embedded; they also publish the latency echoes. The publisher uses the request for an echo exchange to measure the round-trip latency. The time stamp is logged by the publisher from the start of sending the data sample request until it receives the echo of the data sample back from the subscriber. The communication latency between a publisher and a subscriber is one half of the round-trip latency. The average communication latency between a publisher and a subscriber is the average of the 30 thousand times of latency measurement during one test. The reason for measuring the round-trip latency rather than one-way latency is to overcome the challenge of ensuring accurate clock time synchronization between the publisher and the subscriber. Each test scenario is repeated eight times with different data packet size of 100, 200 400, 800, 1600, 3200, 6400 and 12800 bytes. The change in data size represents the change in the number of I/O signals. For example, 100 bytes is equivalent to a controller with 400 digital I/O and 50 analog I/O and 12,800 bytes is equivalent to a controller with 80,000 digital I/O and 2,800 analog I/O. The normal minimum controller scan time resolution is 100ms and about 35ms is allocated for I/O communication services. Therefore, the average communication latency between the controller and the I/O system through the real-time publish/subscribe DDS middleware shall be within 35ms. The subscriber measures the throughput by counting the number of received data packets per second and the throughput rate of Megabits per second.

5.1. PERFORMANCE TEST SETUP

The model set up, shown in Figure , includes one 2.1GHz Lenovo i7-4600U Thinkpad, three 2GHz Lenovo i7-3667 Thinkpads, and one 16-port 10/100 Mbps fast Ethernet switch. Real-time Connext DDS professional middleware version 5.1.0.14-i86Win32VS2013 from Real-Time Innovations, Inc. is installed in all Lenovo Thinkpad laptops. The four laptops are connected to one 16-port 10/100 Mbps fast Ethernet switch.

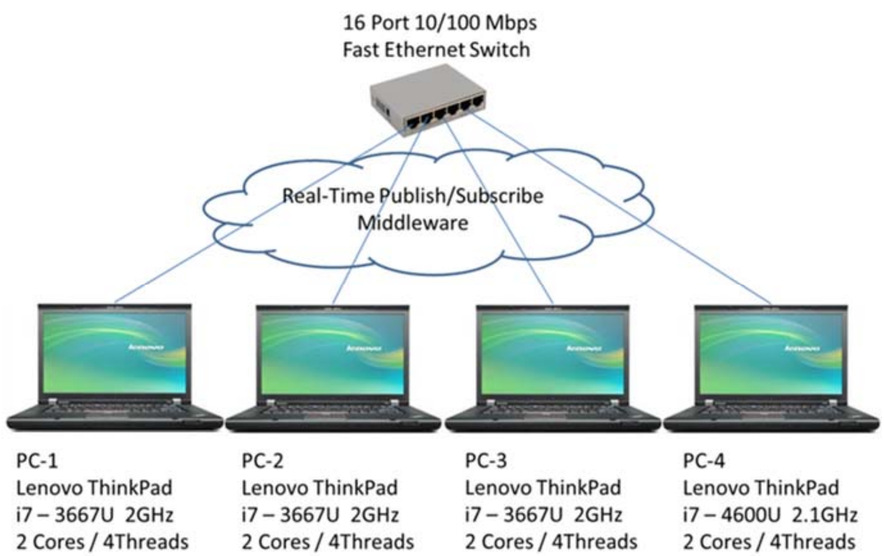

Figure 2 - Performance Test Setup

5.2. DDS QUALITY OF SERVICE POLICES

DDS QoS policies for real-time systems can be used to control and optimize network as well as computing resource to ensure that the right information is delivered to the right subscriber at the right time. The default values are used with the following exceptions to meet the requirement of the automation applications:

- The reliability QoS policy to specify the level of guarantee offered by the DDS in delivering data to subscribers.
- The durability QoS policy to control the data availability with respect to late joining publishers and subscribers.
- The history QoS policy to control whether the DDS should deliver only the most recent value, attempt to deliver all intermediate values, or do something in between.
- The ownership QoS policy to specify whether it is allowed for multiple publishers to write the same instance of the data and if so, how these modifications should be arbitrated.

5.3. BASELINE COMMUNICATION LATENCY ANALYSIS

The baseline performance test is to measure the communication latency of one controller and one I/O system within each laptop. The performance result of the average communication latency between the controller and the I/O system in all laptops is within 1ms as shown in Figure 3, very well below the required scan time resolution of 35ms while varying the controller size from 100 bytes to 12,800 bytes. The data shows that communication latency remains consistently low as message size increases.

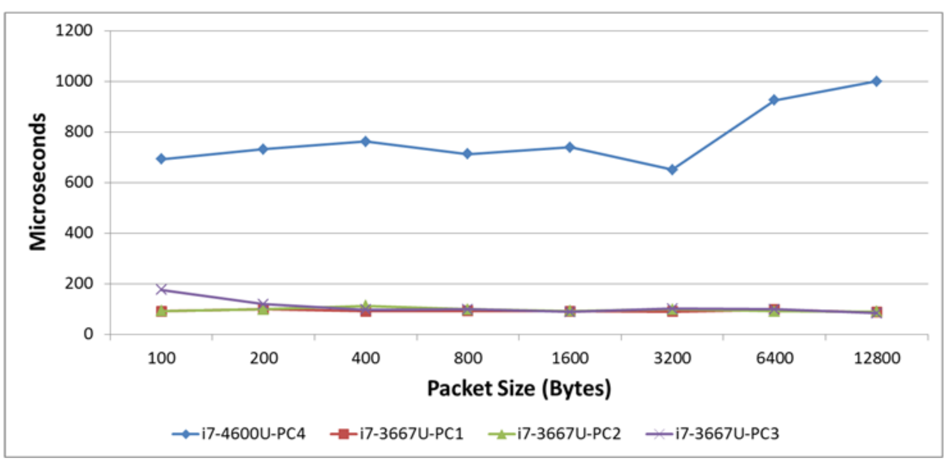

Figure 3 - Average Communication Latency

This is an excellent result showing that the real-time publish/subscribe DDS middleware was able to cope with the huge increase in data loading without any significant impact on the controller performance. Communication determinism is crucial for real-time and mission-critical applications and can be assessed by analyzing the communication latency variation. This is called the jitter analysis as measured in the variability over time of the packet latency across the communication medium. With constant latency, there is no variation or jitter. Hence, a system is more deterministic if it exhibits low jitter. Figure 4 shows the communication latency with jitter analysis for one laptop. Latency at 99th percentile means that only 1% of the data samples exhibited latency larger than this value. The variation between the minimum and 99% latency remains consistently low. This shows that the real-time publish/subscribe DDS middleware between the controller and the I/O system exhibits very low jitter and very high determinism.

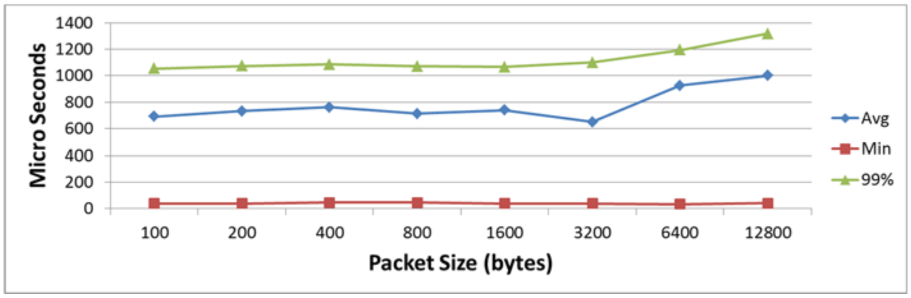

Figure 4 - Average Communication Latency for PC4

5.4. BASELINE COMMUNICATION THROUGHPUT ANALYSIS

The baseline performance test is to measure the communication throughput of one controller and one I/O system within each laptop. Maximum throughput is achieved when the publisher sends as fast as the subscriber can handle messages without dropping a packet. Figure 5 shows a sustainable throughput bandwidth within each laptop.

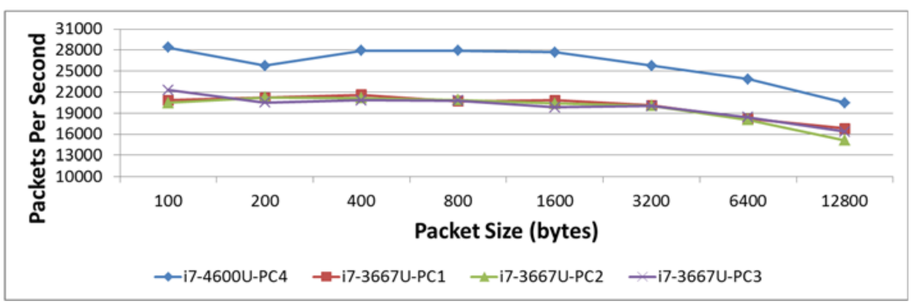

Figure 5 - Average Communication Throughput

The slight decrease in the throughput for a packet size beyond 1600 bytes is because the transmission time of the packets is higher than the communication overhead. Figure 6 shows the average communication bandwidth. It increases significantly with the increase in packet size. This indicates that the real-time DDS middleware does not impose any inherent limit on the aggregate data messaging capacity, making it suitable for scalable automation platforms.

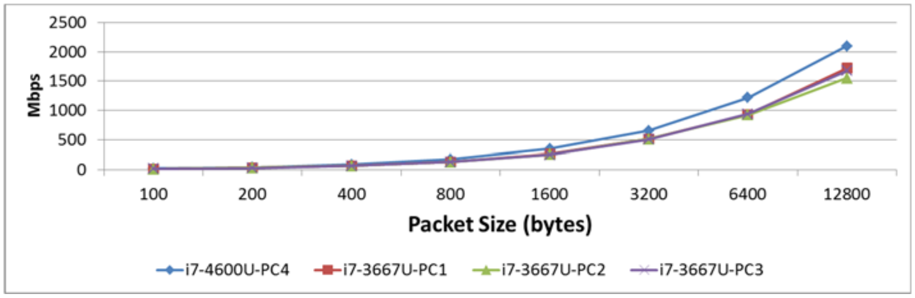

Figure 6 - Average Communication Bandwidth

5.5. COMMUNICATION LATENCY ANALYSIS

The set up for the actual performance test configuration is to host the controller application in one laptop and the I/O system in another laptop. The communication between the controller and the I/O system is implemented through a 16-port 10/100 Mbps 3Com fast Ethernet switch using real-time publish/subscribe DDS middleware. The communication latency is consistently low about 2ms when the publish/subscribe communication crossing the fast Ethernet switch with packet size up to 800 bytes as shown in Figure 7. The average communication latency starts to increase up to about 6ms, six times the baseline case, when the packet size is greater than 800 bytes. However, this result is within the target scan time resolution of 35ms.

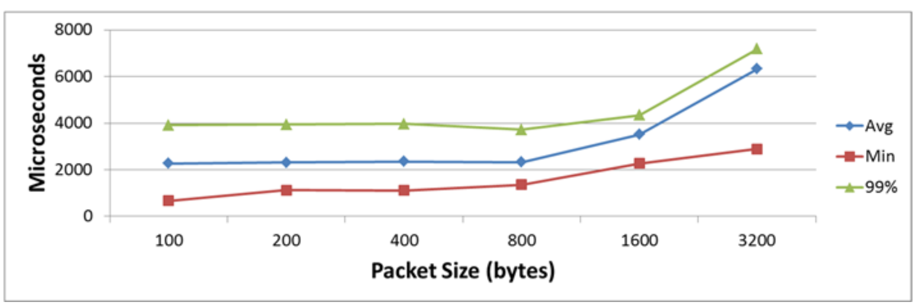

Figure 7 - Average Communication Latency Cross Network Switch

The variation between the minimum and 99th percentile communication latency remains consistently low, similar to the baseline case exhibiting low jitter and high determinism.

5.6. COMMUNICATION THROUGHPUT ANALYSIS

Figure 8 shows the publish/subscribe throughput bandwidth, in terms of packets per second and Megabits per second, between one controller and one I/O system in two laptops connected through a 16-port 10/100 Mbps 3Com fast Ethernet switch. When the packet size is greater than 800 bytes, there is an increase in communication latency and a huge decrease in the throughput in terms of number of packets per second due to the limitation in network bandwidth approaching 100Mbps. Since the quality of service is set to reliable communication, the middleware starts blocking the packets and throttle the communication with maximum bandwidth available close to 100Mbps. Because the real-time DDS middleware uses true peer-to-peer messaging with no centralized or message broker, server, or daemon processes, it does not impose any inherent limit on the aggregate messaging capacity as illustrated in the baseline performance test. It is limited only by the network infrastructure. In all cases, for large systems with packet size of 1,600 bytes and beyond, it is more efficient to use higher network bandwidth capacity such as the 1 Gbps Ethernet switch.

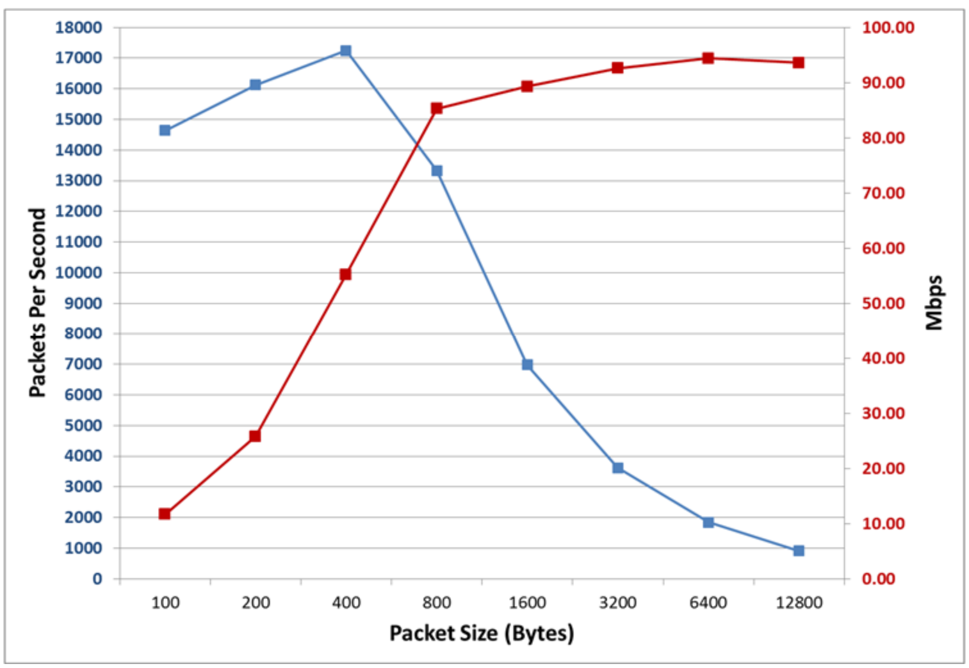

Figure 8 - Average Communication Throughput Cross Network Switch

6. CONCLUSION

The performance results obtained from the empirical tests to evaluate the communication latency and throughput between the controller and the I/O system interconnected using real-time DDS middleware demonstrate the viability of the paradigm shift in the design of automation control environment to provide standard solution based on multi suppliers and avoid automation obsolescence challenges. The following are the advantages of the ubiquitous data-centric automation and control architecture:

- It is a cost effective solution because it is based on field proven interoperable and standard commercial off-the-shelf software and hardware components resulting in optimum capital investment for the total cost of ownership throughout the systems' life span.
- It reduces initial capital investment for grass root projects since it does not require any PIBs and their system and marshaling cabinets as well as the associated wiring down to the junction boxes.
- It provides flexibility to capitalize on existing I/O systems of unsupported legacy controllers and provides a cost effective I/O replacement based on failure module by module or channel by channel. This feature can reduce up to 75% of the capital investment for addressing the obsolescence challenges due to obsolete controller, but supported I/O systems.
- It provides an optimum collaborative environment for all control level applications including regulatory control, advanced regulatory control strategies, multivariable advanced control and real-time control optimization.
- It is highly scalable based on a virtually distributed architecture where I/O modules, CPU and control application are not interwoven. System can grow up by adding multiple virtual machines and/or multiple high-performance fault-tolerant servers for both control and I/O system independently. Also, any changes to the process I/O electrical signals do not have any impact on the control applications.
- It is based on virtual machines that can achieve high utilization since the I/O systems are autonomous and completely independent of the virtual controllers. Adding additional virtual controllers does not require any capital investment for the hardware resulting in maximum total value of ownership.
- It provides centralized security layers of protection with consistent policies cross all servers, similar to the security layers used for cyber security of critical data centers.
- It does not require the actual hardware of the I/O systems during the testing and verification phase at the factory leading to an accelerated project schedule during construction and system installation. Furthermore, testing and troubleshooting the I/O systems is independent from testing and troubleshooting the controllers.

Acknowledgments

The authors acknowledge Saudi Aramco and King Fahd University of Petroleum & Minerals for all support. Special thanks to Real Time Innovations, Inc. for providing the Connext DDS professional middleware for conducting the empirical evaluation.

References

1. C. Pereira and L. Arro, *Distributed Real-Time Embedded Systems: Recent Advances, Future Trends and Their Impact on Manufacturing Automation*, Annual Reviews in Control, **31** (2007), pp. 81-92
2. F. Chen, and T. Repantis, *Coordinated Media Streaming and Transcoding in Peer-To-Peer Systems*, 19th International Parallel and Distributed Processing Symposium, (2005), pp. 56b
3. G. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications* (Springer, 2011)
4. G. McMillan and D. Considine, *Process/Industrial Instruments and Controls Handbook* (McGraw-Hill, 1999)
5. G. Pardo-Castellote, *Data-Centric Programming Best Practices: Using DDS to Integrate Real-World Systems*, Real-Time Innovations, Inc., (2010), pp. 1-18
6. G. Pardo-Castellote, *OMG Data-Distribution Service: Architectural Overview*, Real-Time Innovations, Inc., (2005), pp. 1-7
7. K. An, A. Gokhale, D. Schmidt, S. Tambe, P. Pazandak and G. Pardo-Castellote, *Content-based Filtering Discovery Protocol (CFDP): Scalable and Efficient OMG DDS Discovery Protocol*, DEBS '14, (2014), pp. 1-12
8. K. Ramamritham and S. Son, *Real-time databases and data services*, Real-Time Systems, **28/2** (2004), pp. 179-215
9. L. Zhai, L. Guo, X. Cui and S. Li, *Research on Real-time Publish/Subscribe System supported by Data-Integration*, Journal of Software, **6/6** (2011) pp. 1133-1139
10. L. Zou, Z. Wang, H. Dong, Y. Liu and H. Gao, *Time- and Event-Driven Communication Process for Networked Control Systems: A Survey*, Hindawi Publishing Corporation, Article ID 261738 (2014), pp. 1-10
11. M. Mahmoud and Y. Xia, *Analysis and Synthesis of Fault-Tolerant Control Systems* (Wiley, 2014)
12. Object Management Group, *Data Distribution Service for Real-Time Systems Specification*, Version 1.2, (2007)
13. W. Heinzelman, A. Murphy, H. Carvalho and M. Perillo, *Middleware to Support Sensor Network Applications*, IEEE Network Magazine, **18** (2004)
14. W. Levine, *The Control Handbook* (CRC Press, 2010)

Sadiq M. Sait obtained a Bachelor's degree in Electronics from Bangalore University in 1981, and Master's and PhD degrees in Electrical Engineering from King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, Saudi Arabia in 1983 & 1987 respectively. Since 1987 he has been working at the Department of Computer Engineering where he is now a Professor. In 1981 Sait received the best Electronic Engineer award from the Indian Institute of Electrical Engineers, Bangalore (where he was born). In 1990, 1994 & 1999 he was awarded the 'Distinguished Researcher Award' by KFUPM. In 1988, 1989, 1990, 1995 & 2000 he was nominated by the Computer Engineering Department for the 'Best Teacher Award' which he received in 1995, and 2000. Sait has authored over 200research papers, contributed chapters to technical books, and lectured in over 25 countries. Sadiq M. Sait is the principle author of the books (1) VLSI PHYSICAL DESIGN AUTOMATION: Theory & Practice, published by McGraw-Hill Book Co., Europe, (and also co-published by IEEE Press), January 1995, and (2) ITERATIVE COMPUTER ALGORITHMS with APPLICATIONS in ENGINEERING (Solving Combinatorial Optimization Problems): published by IEEE Computer Society Press, California, USA, 1999. He was the Head of Computer Engineering Department, KFUPM from January 2001 – December 2004, Director of Information Technology and CIO of KFUPM between 2005 and 2011, and now is the Director of the Center for Communications and IT Research at the Research Institute of KFUPM.

Ghalib A. Al-Hashim obtained a Bachelor's degree in Computer Engineering from University of Arizona, Tucson, USA in 1988, Master degree in Computer Engineering from KFUPM in 1998, Master degree in Business Administration from KFUPM in 2008, and currently working on the PhD Dissertation in Computer Science and Engineering at KFUPM. Ghalib is a certified automation professional by International Automation Society. He is also a certified functional safety professional by TUV and CFSE Governance Board. He has been working in Saudi Aramco for more than 32 years specialized on process computer systems. From 2007 until 2014, he was the Corporate Process Automation Work Director.