Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 110 (2017) 394–401

Procedia Computer Science

www.elsevier.com/locate/procedia

The 12th International Conference on Future Networks and Communications (FNC 2017)

Data Distribution Service (DDS) based implementation of Smart grid devices using ANSI C12.19 standard

Basem AL-Madani[a], Hassan Ali[b]*

[a,b]*Computer Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia*

Abstract

Today's power grid has so many challenges in terms of centralized power generation, limited flow of information, limited support for distribution, poor management of peak loads and power disruptions. Due to these limitations several organizations are working on Smart grid. Smart grid consists of numerous kind of heterogeneous devices that increase the complexity and inefficiency. To cope with heterogeneity and provide interoperability among the communication of these devices, middleware is considered to be the best approach. There are so many middlewares that have been proposed so far but Data Distribution Service (DDS) middleware provides high level of reliability and efficiency by addressing more performance metrics and several QoS policies especially in real time and mission critical applications. We have considered Smart grid standard ANSI C12.19 based DDS deployment in transmission and consumption sides. Data structures are obtained for topics formation over RTI Connext to establish communication and to conduct experimental study for the analysis of interoperability and other performance metrics to prove that DDS is better solution for Smart grid data interoperability and high reliability.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: DDS; Smart grid; RTI Connext; RTPS; QoS; performance metrics; ANSI C12.19.

1. Introduction

The traditional power grid only deals with the data that it has supplied but obtains no data in response for the delivery of energy and power. Smart grid is modern power grid that uses digital and analogue communication

* Corresponding author. Tel.: +923004923758.
E-mail address: m.hassan017@hotmail.com

1877-0509
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.06.082

system to collect the information and then it responses according to the collected information. In this way it collects the information from consumer and suppliers in an automated way, thus ensures the reliability, economics, sustainability and efficiency of the production and distribution of electricity[1].

Smart grid contains large amount of various kind of devices, hardware and software of distributed systems that require real time responses. Middleware is like a software that addresses heterogeneity and complexity in these kind of distributed systems. Middleware is used to integrate, control and manage large amount of heterogeneous electrical elements together. Because of their heterogeneous nature, one can find great difficulty in establishing communication among these. So as to provide interoperability, establish reliable communication and to provide QoSs, the middleware architecture is the best solution so far[2,3]. There are several middlewares that have been developed for Smart grid such as GridStat[4], RabbitMQ[5], XMPP[6], USN[7], Service Oriented[8] and DDS[9].

Data Distribution Service (DDS) is considered as a best middleware for Smart grid due to its data management capabilities related to real time publish subscribe and mission critical applications. It also consists of rich Quality of Service (QoS) policies that can be used for various puposes[2,10]. As DDS provides almost every kind of performance metrics so we considered DDS middleware to conduct our experimental study over Smart grid. DDS communication model is based on publish subscribe architecture so we considered Smart grid devices as publisher and subscriber applications to communicate with each other. ANSI C12.19 Smart grid standard[11] is used to get data structures to build topics to establish communication among these devices by setting certain QoS contracts provided by DDS. RTI Connext is used to perform publish subscribe communication and to conduct several experimental analysis that include latency, jitter and throughput.

This paper explores to provide middleware based solution by implementing ANSI C12.19 using DDS. The solution is independent of physical communication network that is flexible, scalable and with rich QoS functionalities. DDS RTI Connext has been used as an implementation tool and several tests have been performed to validate ANSI C12.19 implementation on DDS.

Rest of the paper is organized in the following way. Section 2 briefly explains DDS basics and architecture, section 3 highlights some of the related research work that explains the popularity of middleware specially DDS in various domains. In section 4 some description related to ANSI C12.19 is given while section 5 discusses its implementation over DDS. Section 6 deals with experimental setup and QoS policies used to assess proposed approach. Results and analysis are given in section 7 that concludes with a hint to adopt DDS in Smart grid communication infrastructure.

2. DDS basics and architecture

DDS is a middleware technology based on publish subscribe model[12]. It is particularly for distributed real time and mission critical systems in which heterogeneous data related to development, deployment and maintenance is obtained. DDS is data centric model that provides rich set of QoS policies through which real time systems requirement can be fulfilled. There are several other platform dependent middlewares either with standalone components or with built in modules in various shapes such as Web Services[13], Java RMI[14], OPC[15], CORBA[16] based on either message passing communication model or client server model but they are not that appropriate for real-time mission critical systems[3,17]. DDS is released by OMG[12] for heterogeneous real-time applications which is a platform independent publish/subscribe middleware standard. DDS is able to implement wide variety of mechanisms by defining certain QoS according to required behavior. With the help of these rich specifications, several applications are developed that support certain programming languages in combination with traditional general purpose operating systems. There are some open sources of DDS such as OpenSplice[18] and Open DDS[19] while DDS Connext is Real Time Innovation's proprietary implementation[20]. DDS architecture is shown in Fig. 1 in which components that require QoS can also be seen.

3. Related work

Abdel Rahman et al.[2] surveyed about Message Oriented Middleware (MOM) for real time and distributed environment in smart grid. So far this is the only paper that analyzes the smart grid middlewares based on their functionality and performance.

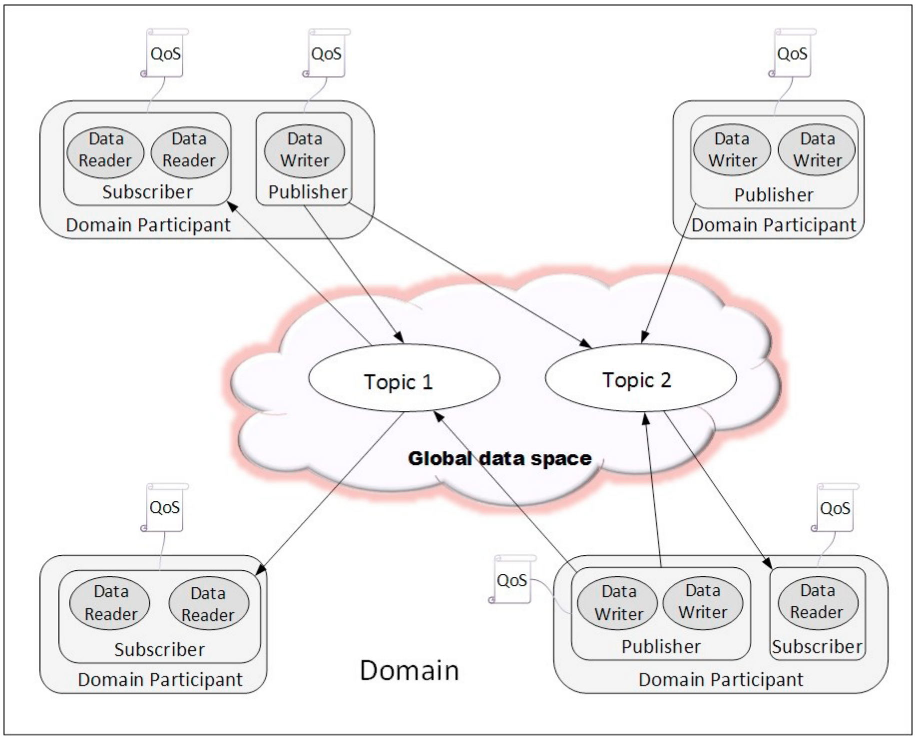

Fig. 1. Real Time Publish Subscribe DDS Architecture

They illustrate MOM in publish subscribe paradigm (PSMOM) that delivers high scalability and asynchronous nature for Many to Many communication infrastructure. They have also considered the support of QoSs in MOM to analyze its role in system performance. They have studied and analyzed three of major industrialized QoS providing middlewares for Smart grid.

1) Extensible Messaging and Presence Protocol (XMPP) is related to XML and provides request response services, presence and real time messaging. It supports QoS functionalities and is a best effort protocol augmented with some extension protocols (XEP). The extended protocol (XEP-0203) provides timestamp information related to stored messages that can be useful in case of slow delivery. The rules to handle time sensitive messages of an application can be defined with the help of advanced message processing extensions (XEP-0097). Negotiating XML streams compressions are delivered through (XEP-0138) and the priorities for resources that are connected are specified by (XEP-0168) protocol. There are several other protocols that it supports.

2) RabbitMQ is based on Advanced Messaging Queue Protocol (AMQP) and is an open source message broker. It addresses wire protocol and the protocol that defines AMQP implementation semantics. In this way these implementations become interoperable with other implementations. Brokering task defined by AMQP is divided among message queues and exchanges which is identical to router that is based on a set of rules (deciding message routing queues) and accepts incoming messages. The third middleware specified in this paper is DDS which is concluded as a most suitable technology for Smart grid applications that is already explained in Section II.

Kai Shi at el.[10] implemented data communication platform in smart micro grid. It is implemented as a wind turbine real-time monitoring system to show the feasibilities and advantages of using data distribution service (DDS) middleware into smart micro-grid monitoring over the IP Network using TCP/IP and UDP protocols.

A web server is built to provide the interface for user to monitor and analyze data. Real time data base monitoring for forecasting, controlling and historical data base is used to monitor data for play back analysis, system performance and energy efficiency. MySQL DBMS is used to store data collected from generation and consumer side along with RTI Connect to provide publish subscribe paradigm. DSpace board is connected to wind turbine to collect and monitor data thus provide data acquisition system.

The data publisher can detect new coming data and send them out into the DDS domain in the network. RTI Real-Time Connect (RTC) service works at the background together with the real-time database. This service worked seamless with the MySQL database server and is able to subscribe the data of interest from DDS domain and simultaneously store the data into the real-time database. The real-time data monitoring service is performed in web server. This web application reads the latest data from the real-time database every second and graphically displays these data to users for monitoring and analysis purpose. Users can access the web site when their devices are connected into the network.

L. Jiang et al.[21] described and tested the deployment feasibility of substation automation standard IEC 61850. They have designed a data model to reduce complexities and difficulties of IEC 61850 implementation by considering Abstract Communication Interface (ACSI) for interoperability. The GOOSE messages from different electric devices are taken from the standard to perform experimental work and establishing communication among these devices and some performance tests are conducted. REST (Representational State Transfer) Services are used along with IEC 61850 data model to demonstrate delivery of abstraction and interoperability enhancement. They have proposed an approach to enable WAN in control systems and electric power Tele-monitoring IEDs for high speed transmission and reliable exchange of data.

4. ANSI C12.19

ANSI C12.19[11] delivers common data structures to transfer data among Smart grid utility end devices especially in meters and other user/customer devices. Standard data structures are represented in terms of table sets that are joined together to form sections known as decades.
For our implementation and experimentation purposes, we have selected two data structures from the decades of this standard. One is 'Electric Element Descriptions' and other one is 'Utility Information'. The definition of these structures are shown in Fig. 2(a) and Fig. 2(b).

(a)

```
struct Electric_Element_Descriptions
    {
        uint    E_FREQ;
        uint    E_NO_OF_ELEMENTS;
        uint    E_BASE_TYPE;
        uint    E_ACCURACY_CLASS;
        uint    E_ELEMENTS_VOLTS;
        uint    E_ED_SUPPLY_VOLTS;
        string  E_CLASS_MAX_AMPS;
        string  E_TA;
        string  E_KH;
        string  E_KT;
        uint    E_INPUT_SCALAR;
        string  E_ED_CONFIG;
        uint    E_ELEMENTS;
        uint    E_VOLTS;
        uint    E_AMPS;
    }
```

(b)

```
struct Utility_Information
{
    string    OWNER_NAME;
    string    UTILITY_DIV;
    string    SERVICE_POINT_ID;
    string    ELEC_ADDR;
    string    DEVICE_ID;
    string    UTIL_SER_NO;
    string    CUSTOMER_ID;
    string    TARIFF_ID;
    string    EX1_SW_VENDOR;
    uint      EX1_SW_VERSION_NUMBER;
    uint      EX1_SW_REVISION_NUMBER;
    string    EX2_SW_VENDOR;
    uint      EX2_SW_VERSION_NUMBER;
    uint      EX2_SW_REVISION_NUMBER;
    string    PROGRAMMER_NAME;
    string    MISC_ID;
    }
```

Fig. 2. (a) Electric element IDL definition; (b) Utility element IDL definition.

5. ANSI C12.19 implementation over DDS

All Interface Descriptive Language (IDL) is based on Remote Procedure Call (RPC) software that is very useful to link machines with different OS, languages and architectures. For our experimental work setup we have constructed IDL files using two topics that are shown in Fig. 2(a) and Fig. 2(b). We have developed a methodology so far based on our IDL files to establish communication among publish subscribe Smart grid applications or devices. Fig. 1 depicts over all behaviour of DDS in which Smart grid device will construct its data structures based on ANSI C12.19 standard. When there is a need to transmit some data, one of the Smart grid device will deliver these data structures to Data Writer. The Data Writer analyses the data whether it is according to structure of specified topic, if so then Data Writer writes the data on related topic through publisher. Subscriber receives this data through Data Reader. Communication is established based on certain QoS contracts.

6. Experimental setup

All The hardware and software specifications used in experimental work is shown in Table 1. The experiment is performed in one of the labs in KFUPM Computer Engineering department. We have used certain QoS policies in our experimentation work through which the behaviour of system is analysed with the help of default and modified QoS policy values. The default and modified values for these QoS policies can be seen in Table 2. By changing the values of QoS policies we have analysed that we can change the behaviour of our system according to our own requirements. Default value for DURABILITY QoS policy is VOLATILE for both publishers and subscribers that means middleware does not keep any data samples for late joining participants due to which scalability can be affected.

This value is modified to TRANSIENT LOCAL, because it allows us to deliver data samples to any late joining new participant. In this way any late joining participant may become part of communication system if it holds same features of declared topics. Default value of RELIABILITY QoS policy is RELIABLE for publishers and BEST_EFFORT for subscribers. These values are swapped in publishers and subscribers to analyse latency and throughput behaviour at both sides.

Table 1. Hardware and software specification

Hosts Specifications	Machine 1 (Publisher)	Machine 2 (Subscriber I)	Machine 3 (Subscriber II)
CPU	Intel(R) Core 2 Duo CPU P8800 @ 2.66 GHz	Intel(R) Core 2 Duo CPU P8800 @ 2.66 GHz	Intel(R) Core 2 Duo CPU P8800 @ 2.66 GHz
OS	Window 7 64 bit	Window 7 64 bit	Window 7 64 bit
Memory	16 GB	16 GB	16 GB
Network	LAN/Wi-Fi 100 Mbps	LAN/Wi-Fi 100 Mbps	LAN/Wi-Fi 100 Mbps

Table 2. Default and modified QoS policies of publishers and subscribers

QoS Policies	Default QoS policies		Modified QoS policies	
	Publisher Value	Subscriber Value	Publisher Value	Subscriber Value
DURABILITY	VOLATILE	VOLATILE	TRANSIENT_LOCAL	TRANSIENT_LOCAL
RELIABILITY	RELIABLE	BEST_EFFORT	BEST_EFFORT	RELIABLE
HISTORY	KEEP_LAST	KEEP_LAST	KEEP_ALL	KEEP_LAST
RESOURCE_ LIMITS	LENGTH_ UNLIMITED	LENGTH_ UNLIMITED	LENGTH_ UNLIMITED	1

HISTORY and RESOURCE LIMITS QoS policies are chosen to support RELIABILITY QoS, because for RELIABLE value, it has to resend the data samples according to HISTORY and RESOURCE LIMITS QoS settings. First we performed communication test by sending and receiving different packets of strings on these machines. Latest version of RTI DDS Connext 5.2.3 is used to establish communication for both publisher and subscriber that has two topics shown in Fig. 2(a) and Fig. 2(b). Ostinato is an open source tool which is used to generate network traffic to load the network to get the realistic performance results. C++ code is generated for publishers and subscribers using rtiddsgen utility on Visual Studio 2012 while Wireshark 1.2.3 and RTI performance test tools are used to do performance measurement over LAN and Wi-Fi.

7. Results and analysis

Random data values are generated at publisher's side and sent through a communication channel to one or more subscribers. The simulation is run on machines specified in Table 1 that are connected through LAN and Wi-Fi. The generated values are transmitted among these machines and required numeric values are taken and stored for analysis.

7.1. Latency and jitter analysis

We used packet sizes from 100 to 200 Bytes based on data structure size in topics for our scenarios to perform latency and jitter tests. Table 3 shows the behavior of latency with default and modified QoS settings. Similarly Table 4 shows the results of jitter. Fig. 3(a) and Fig. 3(b) show the latency graphs that contain comparison between default and modified QoS settings over LAN and Wi-Fi. While Fig. 4(a) and Fig 4(b) show the jitter graphs that contains the comparison between default and modified QoS settings over LAN and Wi-Fi. Latency and jitter is calculated at publisher side while considering the additional network traffic load. It can be analyzed that latency and jitter obtained with default QoS policies values are lesser as compared to modified QoS policies values.

Table 3. Latency performance with default and modified QoS settings

	Latency performance with default QoS settings						Latency performance with modified QoS settings					
Plot	**LAN**			**Wi-Fi**			**LAN**			**Wi-Fi**		
P-S	**Lat. Min (ms)**	**Lat. Max (ms)**	**Lat. Mean (ms)**	**Lat. Min (ms)**	**Lat. Max (ms)**	**Lat. Mean (ms)**	**Lat. Min (ms)**	**Lat. Max (ms)**	**Lat. Mean (ms)**	**Lat. Min (ms)**	**Lat. Max (ms)**	**Lat. Mean (ms)**
1-1	7.8271	12.7639	9.2689	7.2536	13.2871	9.1283	9.3672	15.6327	12.7253	9.8981	15.7253	12.5137
2-2	7.9917	14.2877	9.9127	7.7712	14.1132	10.6148	9.9816	17.2737	14.6927	10.3673	17.9873	13.9001
3-3	8.7113	15.8376	11.2894	8.0182	15.2665	12.0011	10.7365	18.3864	15.9185	12.3379	19.3720	16.1526
4-4	9.0285	16.3679	13.8932	9.9175	17.2651	13.7651	11.2657	20.8509	17.8135	13.8276	21.0019	17.7256
5-5	10.2643	18.1661	14.9028	10.5254	17.8917	14.8716	13.1736	22.1899	18.4162	15.9871	23.1739	19.2567
6-6	11.4325	20.3718	15.8929	11.7355	19.9013	15.5562	15.2637	24.7251	20.1427	16.9908	25.0190	20.8716

Table 4. Jitter performance with default and modified QoS settings

Plot	**With Default QoS Policies**		**With Modified QoS Policies**	
P-S	**LAN Jitter (ms)**	**Wi-Fi Jitter (ms)**	**LAN Jitter (ms)**	**Wi-Fi Jitter (ms)**
1-1	1.87	1.56	3.38	3.36
2-2	2.16	2.34	3.48	4.16
3-3	2.99	3.33	4.28	4.82
4-4	3.76	3.60	4.91	5.21
5-5	3.89	4.24	5.28	5.76
6-6	4.21	4.70	5.87	5.79

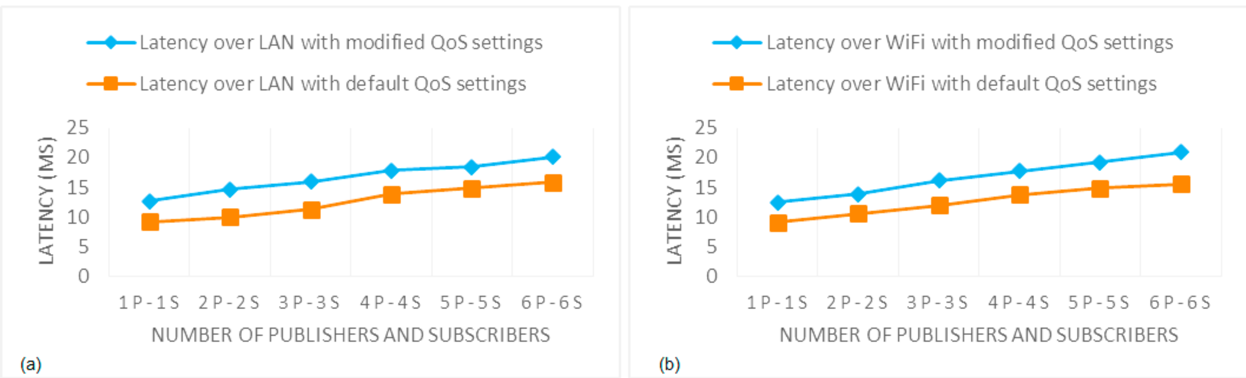

Fig. 3. (a) Latency over LAN with Default and Modified QoS settings; (b) Latency over Wi-Fi with Default and Modified QoS settings

This behavior is obtained due to the RELIABLE value of RELIABILITY QoS policy at publisher side which is the default value, while modified value of RELIABILITY QoS policy that is BEST_EFFORT, offers more latency and jitter. Through graphs we can analyze that by changing the values and settings of various QoS policies, we can modify performance efficiency of the system and can achieve desired results. As by tuning various QoS policies settings, our latency and jitter performance can be changed according to system requirements.

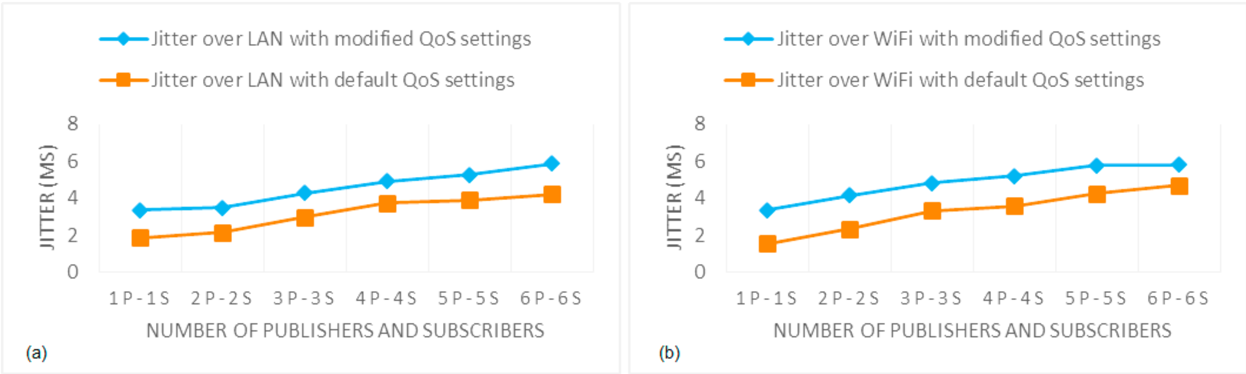

Fig. 4. (a) Jitter over LAN with Default and Modified QoS settings; (b) Jitter over Wi-Fi with Default and Modified QoS settings

7.2. Throughput analysis

Throughput depends on packet size and number of packets sent over a communication channel. Total time consumed for this transmission was recorded and throughput is calculated. Table 5 shows the results obtained for throughput with default and modified QoS settings over LAN and Wi-Fi. Throughput is also measured at the publisher side with default that is RELIABLE and modified that is BEST_EFFORT values of RELIABILITY QoS policy.

Table 5. Throughput performance with default and modified QoS settings

	Throughput performance with default QoS settings						Throughput performance with modified QoS settings					
Plot	**LAN**			**Wi-Fi**			**LAN**			**Wi-Fi**		
P-S	**Throu. Min (Mbps)**	**Throu. Max (Mbps)**	**Throu. Mean (Mbps)**	**Throu. Min (Mbps)**	**Throu. Max (Mbps)**	**Throu Mean (Mbps)**	**Throu Min (Mbps)**	**Throu. Max (Mbps)**	**Throu. Mean (Mbps)**	**Throu. Min (Mbps)**	**Throu. Max (Mbps)**	**Throu. Mean (Mbps)**
1-1	16.73	20.64	18.26	16.52	21.02	18.11	19.67	25.26	22.63	19.45	25.98	22.32
2-2	17.15	21.37	19.62	16.99	21.89	19.35	20.62	25.87	23.06	20.27	26.22	23.29
3-3	17.84	22.78	20.27	17.67	22.53	20.46	20.88	26.81	23.74	21.01	26.89	24.72
4-4	18.47	23.87	21.83	18.90	23.60	21.27	21.76	27.56	24.92	21.79	27.34	25.11
5-5	19.14	24.82	22.54	19.38	24.27	22.98	22.60	28.25	25.73	22.87	27.93	25.73
6-6	19.87	25.92	23.72	20.71	25.78	23.83	23.51	29.27	26.09	23.21	28.99	26.36

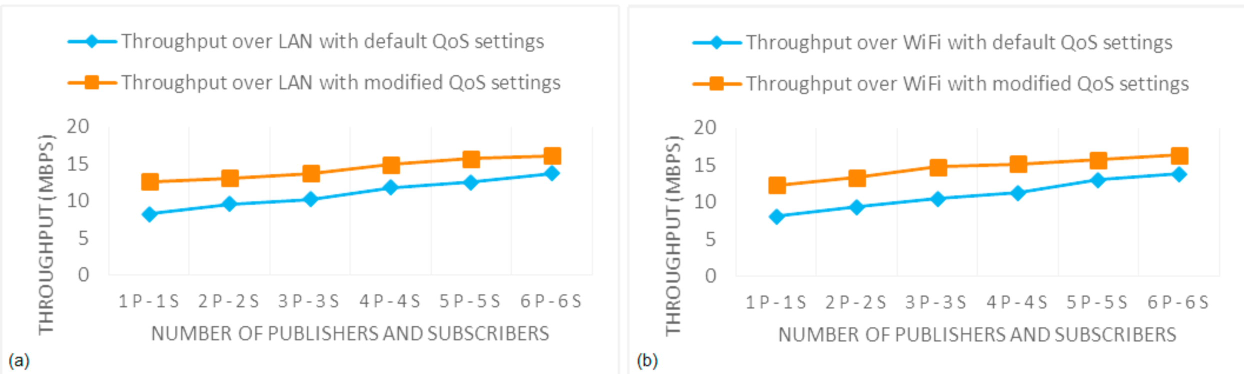

Fig. 5. (a) Throughput over LAN with Default and Modified QoS settings; (b) Throughput over Wi-Fi with Default and Modified QoS settings

It can be analysed that throughput with modified QoS policies is higher than the default values because Data Writer of publisher writes all the time without taking in account the queue size of Data Reader of subscriber. Fig. 5(a) and Fig 5(b) show the throughput graphs. Simulation graphs describes the overall behaviour of the system where throughput increases tremendously by changing the values and settings of various QoS policies, we can change and increase performance efficiency of the system and achieve our desired results based on certain QoS settings. As by doing certain modifications in QoS policies settings mentioned in Table 2, our throughput performance has increased both over LAN and Wi-Fi.

As we deal with so many limitations in regular existing power grid, we need to shift towards Smart grid that provides numerous functionalities on demand but has high complexity in terms of heterogeneity. It has been analysed that in order to provide interoperability and establish communication between numerous heterogeneous devices of Smart grid, a QoS providing middleware is required. Several middleware architectures for Smart grid have been proposed in literature but they only provide limited functionality. Based on previous studies it can be concluded that DDS is the most suitable middleware for Smart grid communication devices. DDS offers rich sets of QoS policies that can be used for flexible and definite outcomes. For ANSI C12.19, communication experiments of end devices are performed over LAN and Wi-Fi to prove real time feature of our proposed framework. Experimental results of latency, jitter and throughput show that this middleware can withstand with tight communication requirements while providing low latency and jitter, consuming less bandwidth and maximizing average throughput with zero packet loss or error. These results are encouraging to deploy DDS for Smart grid devices communication.

References

1. Deotare, P., & Dole, L. (2015, January). Overview of automation of Smart grid network. In *Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on* (pp. 1-3). IEEE.

2. Alkhawaja, A. R., Ferreira, L., Albano, M., & Garibay-Martínez, R. (2012). Qos-enabled middleware for smart grids.

3. Martínez, J. F., Rodríguez-Molina, J., Castillejo, P., & De Diego, R. (2013). Middleware architectures for the smart grid: survey and challenges in the foreseeable future. *Energies*, *6*(7), 3593-3621.

4. Gjermundrod, H., Bakken, D. E., Hauser, C. H., & Bose, A. (2009). GridStat: A flexible QoS-managed data dissemination framework for the power grid. *IEEE Transactions on Power Delivery*, *24*(1), 136-143.

5. Videla, A., and J. J. W. Williams. RabbitMQ in action: distributed messaging for everyone. *Rabbit MQ in action*. (2012).

6. Saint-Andre, P., Smith, K., Tronçon, R., & Troncon, R. (2009). *XMPP: the definitive guide*. " O'Reilly Media, Inc.".

7. Zaballos, A., Vallejo, A., & Selga, J. M. (2011). Heterogeneous communication architecture for the smart grid. *IEEE Network*, *25*(5).

8. Zhou, L., & Rodrigues, J. J. (2013). Service-oriented middleware for smart grid: Principle, infrastructure, and application. *IEEE Communications Magazine*, *51*(1), 84-89.

9. Pardo-Castellote, G. (2003, May). Omg data-distribution service: Architectural overview. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on* (pp. 200-206). IEEE.

10. Shi, K., Bi, Y., & Jiang, L. (2014, September). Middleware-based implementation of smart micro-grid monitoring using data distribution service over IP networks. In *Power Engineering Conference (UPEC), 2014 49th International Universities* (pp. 1-5). IEEE.

11. https://www.nema.org/Standards/ComplimentaryDocuments/ANSI-C1219-2008-contents-and-scope.pdf

12. Omg, O. M. G. (2006). Data distribution service for real-time systems. *OMG, Tech. Rep. OMG Available Specification formal07-01-01*.

13. The World Wide Web Consortium, Web Service Architecture, W3C, 2004, http://www.w3.org/TR/ws-arch/.

14. Oracle Corporation, Java Platform Standard Edition 8 Documentation, 2014, http://docs.oracle.com/javase/8/docs/index.html.

15. OPC Foundation, 2013, https://opcfoundation.org/.

16. Object Management Group, CORBA Specifications, version 3.3, 2012, http://www.omg.org/spec/CORBA/3.3/.

17. Jaloudi, S., Schmelter, A., Ortjohann, E., Sinsukthavorn, W., Wirasanti, P., & Morton, D. (2011, June). Employment of programmable controllers, control systems and service oriented architecture for the integration of distributed generators in the grid. In *Industrial Electronics (ISIE), 2011 IEEE International Symposium on* (pp. 1033-1038). IEEE.

18. Corsaro, A. reasons for choosing opensplice dds, 2009." URL http://www. slideshare. net/Angelo. Corsaro/10-reasons-for-choosing-opensplice-dds (10).

19. http://download.ociweb.com/OpenDDS/OpenDDS-latest.pdf

20. https://www.rti.com/products/dds/index.html

21. Bi, Y. B., Jiang, L., Wang, X. J., & Cui, L. Z. (2013, July). Mapping of IEC 61850 to Data Distribute Service for digital substation communication. In *Power and Energy Society General Meeting (PES), 2013 IEEE* (pp. 1-5). IEEE.

22. Le, T. T. H., Palopoli, L., Passerone, R., & Ramadian, Y. (2013). Timed-automata based schedulability analysis for distributed firm real-time systems: a case study. *International Journal on Software Tools for Technology Transfer (STTT)*, 1-18.