

Research Article

AVL and Monitoring for Massive Traffic Control System over DDS

**Basem Almadani,¹ Shehryar Khan,¹ Muhammad Naseer Bajwa,¹
Tarek R. Sheltami,¹ and Elhadi Shakshuki^{2,3}**

¹Department of Computer Engineering, KFUPM, Dhahran 31261, Saudi Arabia

²Jodrey School of Computer Science, Acadia University, Wolfville, NS, Canada B4P 2R6

³King Faisal University, Al-Ahsa 31982, Saudi Arabia

Correspondence should be addressed to Shehryar Khan; g201303590@kfupm.edu.sa

Received 10 November 2014; Accepted 15 January 2015

Academic Editor: Ansar-Ul-Haque Yasar

Copyright © 2015 Basem Almadani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a real-time Automatic Vehicle Location (AVL) and monitoring system for traffic control of pilgrims coming towards the city of Makkah in Saudi Arabia based on Data Distribution Service (DDS) specified by the Object Management Group (OMG). DDS based middleware employs Real-Time Publish/Subscribe (RTPS) protocol that implements many-to-many communication paradigm suitable in massive traffic control applications. Using this middleware approach, we are able to locate and track huge number of mobile vehicles and identify all passengers in real-time who are coming to perform annual Hajj. For validation of our proposed framework, various performance matrices are examined over WLAN using DDS. Results show that DDS based middleware can meet real-time requirements in large-scale AVL environment.

1. Introduction

Applications of distributed mobile networks exist in everyday life in the form of transportation systems, healthcare systems, weather and environment monitoring systems, and so forth. Such systems require their mobile nodes to communicate and share data among them in real-time. Mobile nodes in these scenarios may be hand held devices, vehicles [1, 2], or robots [3, 4]. With the advancement in embedded systems, it is now possible to allow thousands of mobile nodes to communicate and share huge amount of data. It is also possible to collect the data at the sensor level and forward it to the application level for processing and analysis, all in real-time. These nodes need to share their context updates regularly. In AVL applications, vehicles assume the role of distributed mobile nodes and they require sharing their information such as their locations, vehicles identification, and passenger status. The passengers' information can be extracted using Radio Frequency Identification (RFID) system.

The city of Makkah in Kingdom of Saudi Arabia hosts millions of Hajj pilgrims every year. Hajj consists of number

of rituals, spanning over five designated days that are to be performed in specific locations in and around the city of Makkah. The number of pilgrims is increasing each year and about 3.1 million pilgrims performed Hajj in 2012 [5]. Some of these people get lost in huge crowd and, therefore, there is dire need to identify them, trace their position, and inform their families. There are many models that have been proposed to facilitate the organizers such as in [6] three security requirements are detailed and lattice model is proposed for flow of information. In [7] a bus transportation system within the city of Makkah is proposed and validity of this system is tested using simulation and experiments.

Addressing this challenge necessitates formulation of a framework that identifies and monitors the pilgrims as well as their road transport. Such a large-scale system calls for a vast and scalable infrastructure that supports reliable and instant context updates for sharing among the mobile nodes [8] and, at the same time, is able to dynamically adjust to the load demand. In distributed mobile communication environments, the nodes are limited in power and resources.

The network connections for these mobile nodes are also fluctuating and may suffer from frequent disconnections.

The publish/subscribe (PS) model is by far the most suitable for mobile distribution environment. Although many researchers have attempted to develop publish/subscribe model [9–13], yet only a few of them are able to support mobile networks. The publish/subscribe model has two distinct characteristics. First, it efficiently distributes large amount of data to large number of users. Secondly, the publisher and the subscriber are not required to connect simultaneously in order to distribute data. In PS model, both communicating participants do not know about each other's existence. Nowadays, industrial automation, aerospace, and defense applications use Data Distribution Service (DDS) based middleware [14, 15]. The work presented in this paper also uses DDS based middleware for our application of Automatic Vehicle Location.

DDS is specification for real-time scalable middleware. Its architecture is decentralized and it realized an asynchronous communication model. It specifies many Quality of Service (QoS) policies such as reliability, data flow prioritization, data persistence, and other optimization schemes that are used to control various aspects of data transmission. The unique property of DDS based middleware is that the efficiency of network resources and latency can be controlled by fine tuning some of the QoS policies like latency budget, deadline, and transport priority. In our application, we chose DDS based middleware because of the following characteristics that support mobile distributed environments.

(A) *Asynchronous Interaction.* Network connections in distributed mobile environment have high error rate, high disconnection frequency, and bandwidth limit due to the limited power or lack of spectrum availability. Therefore, asynchronous communication is better than synchronous communication in this type of applications.

(B) *Data Sharing.* The data to mobile devices must be available at all times even when disconnected from the main server. This capability of data sharing and disconnected operation is significant feature in DDS based middleware. It should be noted that consistent, reliable, and efficient accessibility to the database should be provided by the mobile distributed information system. This implies that global distributed database should also be synchronized among all nodes even after disconnection.

(C) *Reflection and Dynamic Reconfiguration.* A heterogeneous environment of dynamic context is encountered by the devices in mobile communication scenarios. In many heterogeneous systems, devices may exhibit different behaviors due to diversity in their network protocols, I/O interfaces, and OS. Thus, mobile nodes/devices should be adaptive to the available resources, which require techniques to optimize the behavior of nodes. Therefore, the middleware should detect the changes in the availability of resources to support dynamic reconfiguration. It also has to notify the application for the reallocation of resources to different nodes.

2. Related Work

For large-scale mobile system, a middleware called Scalable Context-Aware Middleware for Mobile Environments (SALES) is developed in [16]. It is a tree based classified model of nodes for performance evaluation, load balancing, and calculating communication cost among the following four types of nodes: (1) central node, (2) base node, (3) coordinator user node, and (4) simple user. SALES does not take advantage of real-time DDS and uses UDP. Two main terminologies are used: Quality of Context (QoC) and Context Data Distribution Level Agreement (CDDL). QoC is associated with context information distributive service whereas CDDL is quality agreement between consumer and producer imposed by the middleware. This SALES architecture lacks the functionality of fault tolerance, QoS support, and context updates by mobile nodes.

For ubiquitous computing, a middleware named Solar is developed in [17]. It uses two protocols: TCP for interplanetary communication and Distributed Hash Table (DHT) for routing and discovery. It is built on the basis of self-organizing peer-to-peer network and is designed for scalability among the set of communicating nodes. It employs filter and pipe programming model. Each filter has group of entry and exit points as well as sources and sinks. Each node in the solar architecture is viewed as planet and each planet has a number of satellite nodes. Solar has the reliability of TCP that may not be suitable for some real-time applications. It also lacks the functionality of fault tolerance. A limited research is done in the implementation of mobile distributed applications using DDS based middleware. Among few of them is [4]. Its architecture supports mobile nodes and provides reliable data delivery. It also supports handover by switching the wireless access points. The mobile nodes in the proposed middleware execute light version of DDS, whereas the fixed nodes execute full version and are responsible for routing and data delivery among all nodes. Due to the architecture of this middleware, mobile nodes have to run in a single domain and stable wireless connectivity is necessary. This proposed middleware, too, lacks the functionality of fault tolerance.

REliable and VErsatile News delivery support for aGENcies (Revenge) is a DDS based middleware, which serves as news dispatching service among mobile nodes [18]. It efficiently disseminates the data among the mobile nodes in DDS domain. It is also based on self-organizing peer-to-peer network and is fault tolerant. Revenge efficiently detects the crashed nodes and reroutes the paths from any source to any sink. This architecture, however, lacks the functionality of handover. Another DDS based middleware is proposed in [19] for real-time communication between mobile nodes using proxy approach. A proxy DDS client is used for managing, coordinating, and forwarding all the data to mobile nodes. This architecture provides both reliable and unreliable data delivery. Due to the Firewall/NAT restrictions, mobile nodes have to run within a single domain with continuous connectivity.

A DDS based middleware called Scalable Data Distribution Layer (SDDL) [20, 21] is proposed for real-time tracking of mobile nodes. This middleware connects the stationary

DDS nodes in a wired network to the mobile nodes with IP based wireless connection. Two protocols are used in this middleware: RTPS wire protocol for communication among the stationary nodes and mobile reliable UDP protocol for communication among the mobile nodes. In wired core network, there are three types of stationary nodes with unique roles: (1) gateway that is responsible for maintaining separate connections with mobile nodes through Mobile Reliable-(MR-) UDP connections; it is also responsible for notifying other core network nodes when a mobile node is connected or disconnected; (2) Point of Attachments (PoA) Manager that distributes the list of points of attachment/gateways to mobile nodes and it may later switch to different gateway; mobile nodes may also switch the gateway when they detect weak connection with the current gateway; and (3) group definers, which are responsible for evaluation of group-membership of mobile nodes. They subscribe to a specific DDS topic and map each node to one or more groups corresponding to application specific group membership logic. When a new message is sent to a group, a gateway asks for group-to-mobile-node mapping to know which mobile node is ready to send messages.

An RFID system based on publish/subscribe middleware is introduced in [22]. Since different applications require different data, middleware has to adapt to all applications. When more applications need to be incorporated, it is necessary to adjust middleware functionalities to satisfy them. This, however, costs time and effort. Using publish/subscribe mechanism applications can subscribe to events from the RFID reader dynamically. In this model filtering of messages can be topic based or content based. In topic based filtering, the subscriber receives messages published to a specific topic. All subscribers will receive the same messages for a particular topic. In content based filtering, subscribers will receive messages if the content of the messages match the constraints set by the subscribers. Four components are proposed for publish/subscribe model in this RFID middleware: (1) list of publications maintained by the message manager, (2) list of subscriptions maintained by the message manager, (3) message manager that acts as a controller and responds to the requests of the reader manager for maintenance; it also responds to the client queries for the list of publication and maintains the list of subscribers, and (4) an API with a set of routines and protocols to allow the clients to subscribe and unsubscribe.

3. DDS Architecture

DDS specifies a communication model that is data centric publish/subscribe for various computing and distributed environments. This data centric publish/subscribe (DCPS) uses a Global Data Space (GDS) where publishers post and the DCPS model disseminates this information to all the interested subscribers.

Figure 1 illustrates all the basic constructs in a simple DCPS model. A domain is a virtual network area and all the publishers and subscribers can send and receive messages within a domain. A publisher is an object that, according to the publisher's QoS policies, distributes data and publishes

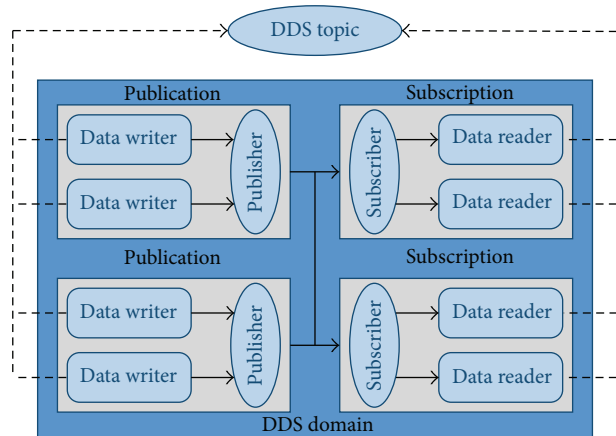


FIGURE 1: DDS architecture [19].

different types of data objects. Data writers are used by the publishers to write data to the GDS. A subscriber receives the publishing data using data readers and, according to subscriber's QoS policies, makes them available to the receiving application. It reads the topics from GDS for which there exists a matching subscription and data readers are informed that data is received.

The DDS has two types of discovery protocols: the Participant Discovery Protocol (PDP) and the Endpoint Discovery Protocol (EDP). Through these protocols participants can dynamically discover the existence of other participants and also inform other participants about the end points such as data readers and data writers and so forth. The basic architecture of DDS is described as the unidirectional data communication in a fixed network. In this network a publisher pushes the data to the subscriber which is saved in its history caches. Topic is the data type that links the publishers and subscribers. Implementation requires this discovery protocol to identify the existence, presence, and absence of the endpoints when the network is joined or left by them. Another key distinguishing feature of DDS as compared to another publish/subscribe middleware is that it has rich QoS support. The behavior and performance of the DDS service and how it performs the various coordination tasks depend upon how its QoS policies are configured. For mobile environment the useful QoS policies are durability, history, and reliability. Below is a brief description of each of them.

- (i) For mobility support, RELIABILITY QoS policy is used indicating the reliability level between publisher and subscriber. If RELIABILITY is set to BEST_EFFORT then the publisher will push the samples without retransmissions or expecting acknowledgements. If RELIABILITY is set to RELIABLE then publisher will guarantee the delivery of samples to the subscriber. If the subscriber is disconnected or unavailable and samples are unacknowledged then the publisher can retransmit the samples using the DURABILITY QoS policies.

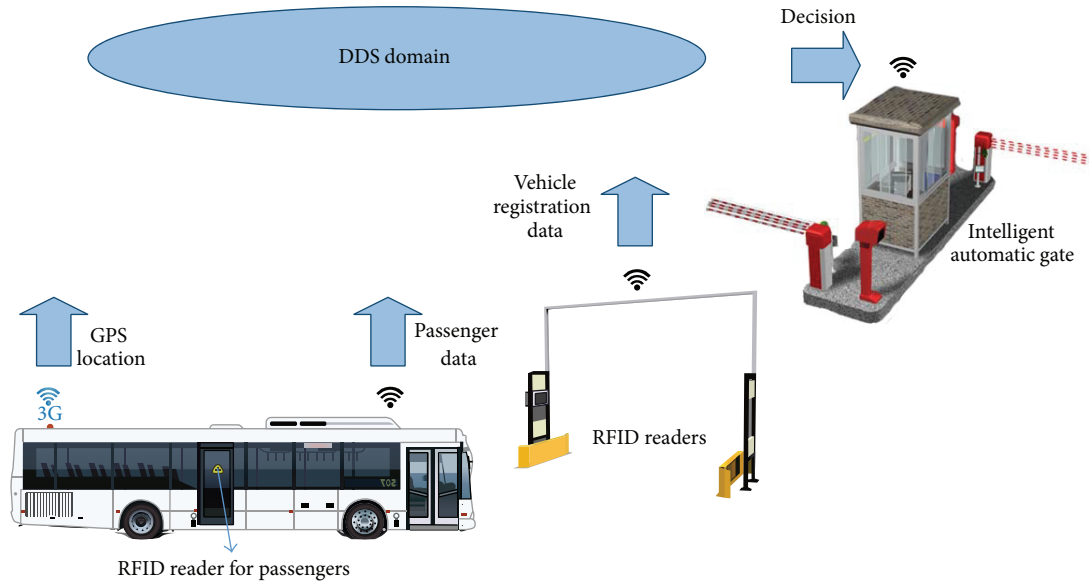


FIGURE 2: AVL and monitoring system architecture.

- (ii) DURABILITY QoS indicates whether previous data samples will be made available to late joining subscriber by the publisher or not. By setting the DURABILITY policy to PERSISTENCE, the past data samples are stored in the permanent memory such as FLASH, HARD DISK. Due to this the subscriber can have this data when it reconnects to the system at any time. The DURABILITY policy depends upon the HISTORY QoS policy and RESOURCE_LIMITS QoS policy.
- (iii) HISTORY QoS specifies the maximum number of data samples that are stored in the History Cache of publisher for reconnecting or late joining subscribers.
- (iv) RESOURCE LIMIT specifies maximum number of samples and instances which publisher or subscriber can manage.

4. Proposed Solution

In this section we propose a framework for location and monitoring system based on DDS based middleware especially for pilgrims of Hajj. To provide control and location services of this huge crowd, each pilgrim can be provided with a GPS transmitter; however, implementing this solution is too expensive. Similarly, another possibility is that each person can be provided with a radio transmitter with unique identifier and the transmitter can update pilgrim's location by sending update to the database via base station. Although it may be useful, signal coverage can be a problem. Our solution addresses these challenges during journey of pilgrims and offers crowd management and prevention of accidents using the architecture depicted in Figure 2.

Proposed approach can manage a huge system consisting of about 50,000 buses as mobile nodes as well as the passengers inside these vehicles. This system is especially useful for

ensuring that all pilgrims have valid Hajj permits. Assume there are various entry points to the city of Makkah and on each entry point there is check posts almost 10 Km from the city. Three layers of security are implemented on each check post. These include RFID readers to read electronic license plates, distributed database server, and automatic gates access based upon previous two security layers. The detailed layout of the framework is given below.

4.1. Bus or Mobile Node

- (i) *GPS and 3G Device.* Every bus has a 3G device that periodically updates its current locations using GPS.
- (ii) *Thermal Camera.* These cameras count number of passengers in the bus and send this information to local database in the bus.
- (iii) *RFID Readers.* Each passenger has RFID wrist bands with their personal information. This information is collected using RFID readers inside the bus and sent to local database.
- (iv) *WiFi Device.* Local database is synchronized with DDS cloud using WiFi.

The message format for RFID passenger data is shown in Figure 3.

4.2. *The Check Post.* The check posts are established at different entry points of the city. Each check post has various parallel lines. They have distributed servers and databases which are part of the DDS cloud. Also, multiple RFID readers are installed for vehicle's identification through passive RFID license plates.

4.3. *The Automatic Gate.* At the end of each check post there is an automatic gate that will only open after a positive

Pilgrim ID	Name
Country	Passport number
ID expiration	Group
Agency	

FIGURE 3: Message format in RFID message.

decision is made inside the cloud for a particular vehicle based on different factors such as vehicle identification, passenger counting through thermal cameras, and passenger identification and status through RFID data. When a bus arrives at a check post it has to pass through all three stages before it is finally allowed to enter the city.

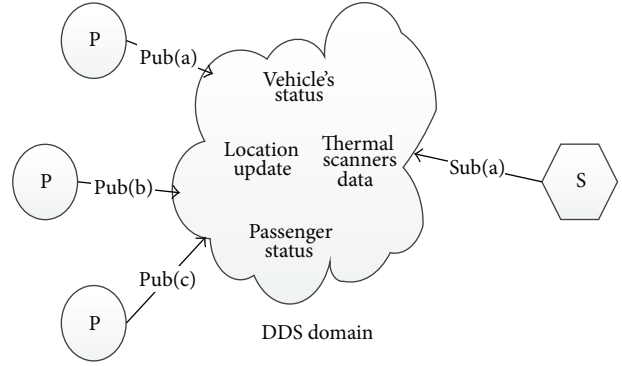
In the first part of this clearance, there are RFID readers for the verification and authentication of license plates and a decision is made whether this bus is allowed to operate for Hajj pilgrims or not. At this stage it is also verified whether this mobile node is physically present at the checkpoint or not using its GPS location update. This cross checking can deal with identity theft of vehicles. In second part the stored RFID data of passenger identification and status inside the bus is passed to the cloud via WiFi for authentication and decision making. If a mobile node/bus has more than a certain number of unauthorized passengers then the whole bus is rejected; otherwise manual checking is done. In third stage the thermal cameras installed on the bus will do the head counting and will convey this data to the DDS cloud over WiFi. At this point the passenger count data from RFID readers and the thermal cameras are cross checked for further verification in case there are multiple readings for RFID tags.

At the end when a mobile node reaches the automatic gate, a decision is made based on the data provided to the cloud during the previous stages. In our case the number of publishers is in thousands whereas subscriber is one. There are four topics for which publishers are publishing as shown in Figure 4. These topics are vehicle location update, vehicle registration information, passenger head count, and RFID passenger's data.

5. Experimentation Setup

Before going to experimentation and results, some performance parameters are defined below to validate the timeliness of our proposed solution.

- (i) The time taken by a data packet to reach the receiver side is known as the latency. This includes both the propagation delay as well as the queuing delay and can



S: subscriber
P: publisher

FIGURE 4: DDS domain.

be calculated by dividing Round Trip Time (RTT) by 2:

$$\text{Latency} = \frac{\text{RTT}}{2}. \tag{1}$$

- (ii) Variation in latency is known as the jitter. The smaller the value of the jitter, the smaller the variation in the delay of packets. It means that with small values of jitter we can be sure that delay of the packets will be same for most of the time during their journey from sender to receiver. Jitter can be calculated as given in

$$\text{Jitter} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}, \tag{2}$$

where the total number of delay samples is represented by N and mean value of N delay samples is represented by \bar{x} .

- (iii) The average rate of successful data transmission is known as the throughput. This includes the payload as well as the protocol overhead. Following equation is used to calculate the throughput:

$$\text{Throughput} = \frac{\text{Packet Size} \times \text{Number of Packets}}{\text{Total Time}}. \tag{3}$$

We provide hardware platform used in our experiment as well as software tools needed in Tables 1 and 2, respectively.

To support mobility and ensure real-time behavior of the system, QoS policies used in our experimentation are shown in Table 3.

TABLE 1: Platform specifications for WLAN.

	Host A publisher	Host B subscriber
CPU	Intel(R), Core (TM)2 Duo CPU P8800 @ 2.66 GHz	Intel(R), Core (TM)2 Duo CPU P8800 @ 2.66 GHz
OS	Windows 7 64 bit operating system	Windows 7 64 bit operating system
Memory	4 GB	4 GB
Network connection	WLAN IEEE 802.11, 54 Mbps	WLAN IEEE 802.11, 54 Mbps

TABLE 2: Software specifications for WLAN.

Tool	Version	Purpose
RTI Connex- RTI DDS Latency Test	5.0.0	Measure latency (one way delay)
Wire-shark	1.2.3	Measure throughput

TABLE 3: QoS policies for mobility.

QoS policy	Publisher	Subscriber
DURABILITY	PERSISTENCE	PERSISTENCE
RELIABILITY	RELIABLE	RELIABLE
HISTORY	KEEP_ALL	KEEP_LAST
RESOURCE_LIMIT	LENGTH_UNLIMITED	1

TABLE 4: Latency and jitter for many-to-one model.

Number of publishers	Minimum (m sec)	Maximum (m sec)	Avg. (m sec)	Avg. [23] (m sec)	Jitter (m sec)
1	1.28	63.51	2.41	12.18	2.35
2	1.38	57.29	2.47	13.17	1.96
4	1.77	112.44	4.00	12.45	4.68
6	2.06	454.14	4.23	12.71	5.16
8	2.22	707.40	5.95	12.93	7.53

6. Results and Analysis

The bottleneck in our proposed framework can be data transmission over WiFi at check posts. Therefore, we conducted experiments to measure various performance matrices for this wireless medium. Latency and jitter can validate if the proposed solution can fulfill real-time requirements in AVL application for Hajj pilgrims. The values of throughput can shed light on whether WiFi can withstand high data-rate demand of such scenario. The obtained results for latency, jitter, and throughput for Wifi over DDS are tabulated in Tables 4–7. There is significant improvement in the results compared to [23] because of the introduction of QoS policies as well as controlled environment for experimentation making sure that only DDS applications are running over WiFi.

For latency and jitter, 1024-byte payload is used. First, one subscriber and multiple publishers scenario is examined.

TABLE 5: Latency and jitter for many-to-many model.

Number of publishers	Number of subscriber	Minimum (m sec)	Maximum (m sec)	Avg. (m sec)	Jitter (m sec)
2	2	1.42	131.93	2.69	2.95
4	2	1.64	237.24	3.43	3.36
4	4	1.55	106.48	2.40	3.39
8	4	2.43	291.53	6.32	4.28

TABLE 6: Throughput for many-to-one model.

Number of publishers	Total packets ($\times 1000$)	Total time (sec)	Throughput (Mbps)	Throughput [23] (Mbps)
1	4585	5536	6.810	0.842
2	2794	3277	6.958	1.598
4	3330	3807	7.168	3.386
6	3780	3986	7.772	3.515
8	5040	5544	7.498	2.91

TABLE 7: Throughput for many-to-many model.

Number of publishers	Number of subscribers	Total packets ($\times 1000$)	Total time (sec)	Throughput (Mbps)
2	2	3560	3706	7.87
4	2	4032	4360	7.67
4	4	4032	2767	12.37
8	4	4032	2730	12.59

In each run 20,000 to 50,000 packets are sent and the minimum, maximum, and average values are calculated. Tests are repeated up to 10 times and overall average is taken to account for any random fluctuation in performance measures.

Table 4 records measurements obtained from experiments. Jitter is calculated using (2). We can see that both latency and jitter increase linearly as the number of publishers grow. Figures 5 and 6 show latency and jitter graphs corresponding to Table 4. In Figure 5 we can observe that there is considerable improvement in latency as compared to results in [23] because specific QoS policies for mobility are introduced in our scenario and Wifi experimentation was carried out in a controlled lab setup to avoid other applications running on the same WiFi.

Latency and jitter are also calculated for many-to-many communication scenario. Tests are run to examine the effect of multiple publishers and multiple subscribers on the latency and jitter. As expected, both performance measures have higher values when multiple participants try to transmit the data over a single channel. These results are tabulated in Table 5. Figures 7 and 8 show the results graphically.

The data packets' size and the frequency of the transmission affect the throughput. Data size used in our case is up to few hundred bytes except for the local database updates that are synchronized with DDS cloud. These updates are sent in

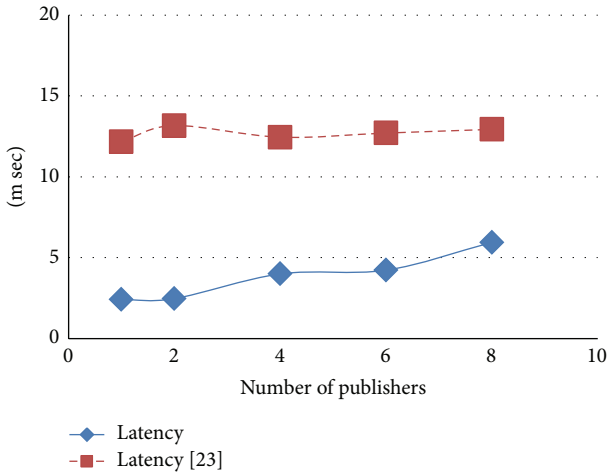


FIGURE 5: Average latency in many-to-one model.

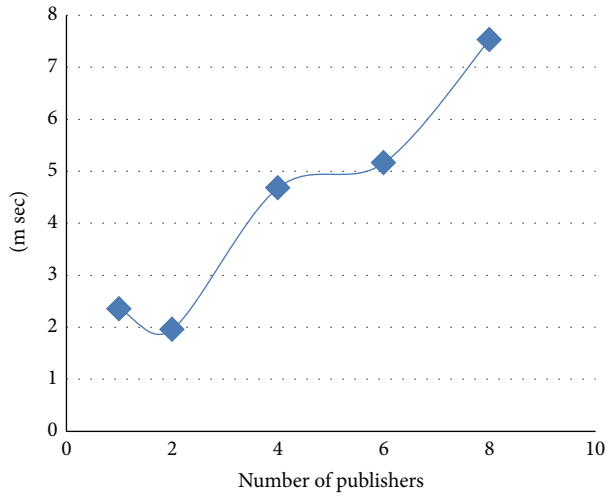


FIGURE 6: Jitter in many-to-one model.

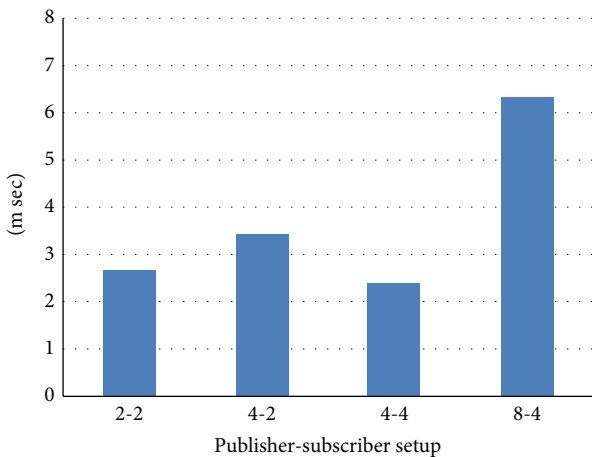


FIGURE 7: Average latency in many-to-many model.

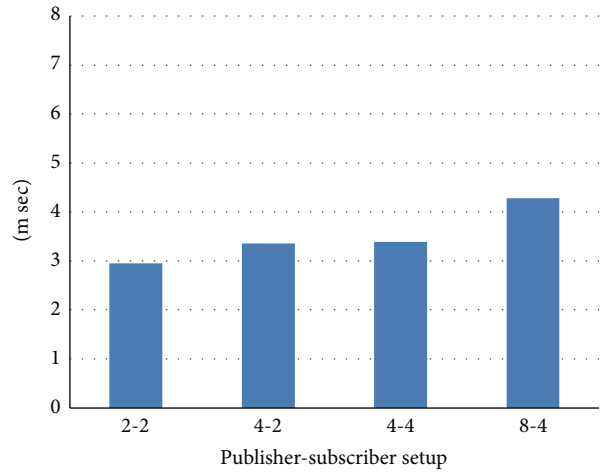


FIGURE 8: Jitter in many-to-many model.

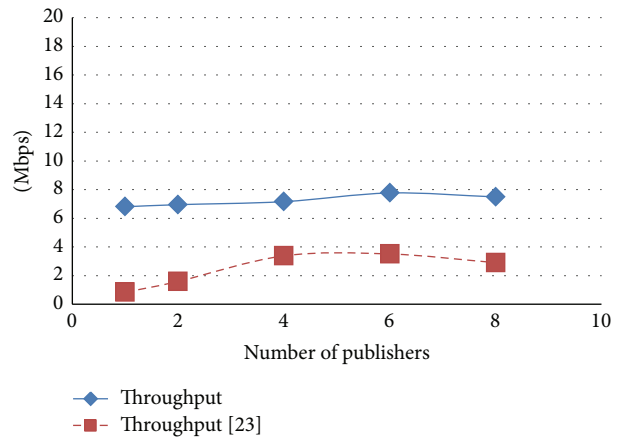


FIGURE 9: Throughput in many-to-one model.

packets up to 1 KB. Hundreds of thousands of samples are sent from publisher side and received on the subscriber side in each iteration. Total time for this communication is observed and throughput is calculated using (3). The tests are repeated at least 10 times to get more accurate results.

Similar to the latency results we can see in Table 6 and Figure 9 that there is considerable difference in the throughput. The reason is again judicious selection of QoS values in mobile applications. However, we can observe that with the increase in number of participants there is negligible difference in the throughput.

Like many-to-many latency and jitter experiments, we also conducted tests to measure average WiFi in many-to-many mode. Various configurations of publishers and subscribers are examined. Table 7 summarizes the results obtained. We can observe from Figure 10 that throughput is not significantly affected by the number of participants in many-to-many scenarios as well.

Here, it is worth pointing out that the maximum numbers of publishers in our experiments are limited to only eight whereas the total numbers of vehicles are in thousands. The

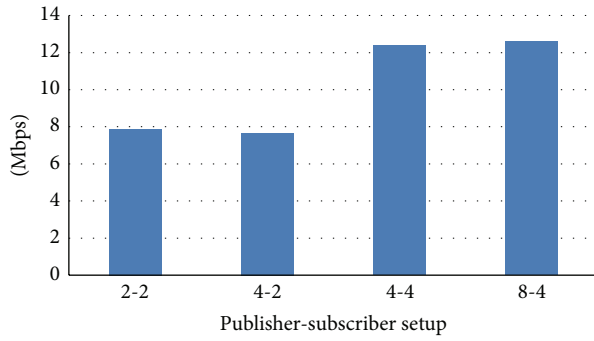


FIGURE 10: Throughput many-to-many model.

reason is that this experimentation shows the data communication over WiFi at checkpoints. If, on each highway, there are four lanes for vehicles then this framework can deal with eight buses simultaneously. And the small values of latency points out that the buses are cleared almost instantaneously within any let or hindrance.

7. Conclusion

Monitoring and tracking vehicles during Hajj have been a tough task for Saudi Arabian authorities for a long time. Illegal Hajj pilgrims also pose a challenge in organization of annual event. In this paper, we introduced a framework to solve this problem of vehicle tracking during Hajj by using OMG's DDS middleware specification. We discussed pilgrims' difficulties and transport congestion problems faced during this period. An automated solution based on Real-Time Publish/Subscribe middleware is proposed for tracking and monitoring of vehicles as well as pilgrims. The use of DDS based middleware is motivated by its data centric and asynchronous communication paradigm along with rich set of QoS policies. Experiments are performed for WLAN over DDS to validate real-time characteristics of our proposed framework. Obtained results for various performance parameters such as latency, jitter, and throughput show that the proposed approach can withstand stringent real-time requirements in AVL application under consideration. These preliminary results are encouraging enough to serve as bases for further development of the framework, and ultimately the infrastructure.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are thankful to RTI Inc. for granting license for this research. The authors are also thankful to College of Computer Science and Engineering (CCSE), KFUPM, Saudi Arabia, for providing necessary lab equipment and setup for experimentation.

References

- [1] D. Stojanović, B. Predić, I. Antolović, and S. Dordević-Kajan, "Web information system for transport telematics and fleet management," in *Proceedings of the 9th International Conference on Telecommunications in Modern Satellite, Cable, and Broadcasting Services (TELSIKS '09)*, pp. 314–317, October 2009.
- [2] J. Rybicki, B. Scheuermann, W. Kiess, C. Lochert, P. Fallahi, and M. Mauve, "Challenge: peers on wheels—a road to new traffic information systems," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '07)*, pp. 215–221, September 2007.
- [3] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, "Robomote: a tiny mobile robot platform for large-scale ad-hoc sensor networks," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1143–1148, May 2002.
- [4] A. Herms, M. Schulze, J. Kaiser, and E. Nett, "Exploiting publish/subscribe communication in wireless mesh networks for industrial scenarios," in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA '08)*, pp. 648–655, Hamburg, Germany, September 2008.
- [5] F. Abdessemed, "An integrated system for tracking and control pilgrims shuttle buses," in *Proceedings of the 14th IEEE International Intelligent Transportation Systems Conference (ITSC '11)*, pp. 384–389, IEEE, Washington, DC, USA, October 2011.
- [6] MakkahGIS, <http://www.MakkahGIS.net>.
- [7] HajjHousing, <http://www.hajjhousing.net>.
- [8] M. Grossmann, M. Bauer, N. Hönle, U.-P. Käppeler, D. Nicklas, and T. Schwarz, "Efficiently managing context information for large-scale scenarios," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 331–340, March 2005.
- [9] Y. Huang and H. Garcia-Molina, "Publish/subscribe in a mobile environment," *Wireless Networks*, vol. 10, no. 6, pp. 643–652, 2004.
- [10] M. Castro, P. Druschel, A.-M. Kermarrec, and A. I. T. Rowstron, "Scribe: a large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1489–1499, 2002.
- [11] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Design and evaluation of a wide-area event notification service," *ACM Transactions on Computer Systems*, vol. 19, no. 3, pp. 332–383, 2001.
- [12] W. W. Terpstra, S. Behnel, L. Fiege, A. Zeidler, and A. P. Buchmann, "A peer-to-peer approach to content-based publish/subscribe," in *Proceedings of the 2nd International Workshop on Distributed Event-Based Systems (DEBS '03)*, pp. 2–9, San Diego, Calif, USA, June 2003.
- [13] P. R. Pietzuch and J. M. Bacon, "Hermes: a distributed event-based middleware architecture," in *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, pp. 611–618, Vienna, Austria, 2002.
- [14] OMG Data Distribution Service for Real-time Systems Specifications, 2012, <http://www.omg.org/spec/>.
- [15] N. Wang, D. C. Schmidt, H. van't Hag, and A. Corsaro, "Toward an adaptive data distribution service for dynamic large-scale network-centric operation and warfare (NCOW) systems," in *Proceedings of the IEEE Military Communications Conference (MILCOM '08)*, pp. 1–7, November 2008.
- [16] A. Corradi, M. Fanelli, and L. Foschini, "Adaptive context data distribution with guaranteed quality for mobile environments,"

- in *Proceedings of the IEEE 5th International Symposium on Wireless Pervasive Computing (ISWPC '10)*, pp. 373–380, May 2010.
- [17] G. Chen, M. Li, and D. Kotz, “Data-centric middleware for context-aware pervasive computing,” *Pervasive and Mobile Computing*, vol. 4, no. 2, pp. 216–253, 2008.
- [18] A. Corradi, L. Foschini, and L. Nardelli, “A DDS-compliant infrastructure for fault-tolerant and scalable data dissemination,” in *Proceedings of the 15th IEEE Symposium on Computers and Communications (ISCC '10)*, pp. 489–495, June 2010.
- [19] K.-J. Kwon, C.-B. Park, and H. Choi, “A proxy-based approach for mobility support in the DDS system,” in *Proceedings of the 6th IEEE International Conference on Industrial Informatics*, pp. 1200–1205, Daejeon, Republic of Korea, July 2008.
- [20] D. Lincoln, V. Rafael, A. Lucas, A. Rafael, B. Gustavo, and E. Markus, “A large-scale communication middleware for fleet tracking and management,” in *Proceedings of the 30th Brazilian Symposium on Computer Networks and Distributed Systems*, pp. 964–971, May 2012.
- [21] L. David, R. Vasconcelos, L. Alves, R. André, and M. Endler, “A DDS-based middleware for scalable tracking, communication and collaboration of mobile nodes,” *Journal of Internet Services and Applications*, vol. 4, no. 1, pp. 1–15, 2013.
- [22] J. Y. J. Yu and S. L. S. Lai, “Message publish/subscribe system of RFID middleware,” in *Proceedings of the International Conference on New Trends in Information and Service Science (NISS '09)*, pp. 288–292, Beijing, China, June-July 2009.
- [23] B. Almadani, S. Khan, T. R. Sheltami, E. M. Shakshuki, M. Musaddiq, and B. Saeed, “Automatic vehicle location and monitoring system based on data distribution service,” *Procedia Computer Science*, vol. 37, pp. 127–134, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

