Chapter

# Middleware Architecture - History and Adaptation with IEEE 802.11

*Rochak Bajpai, Atul Bansal, Jyoti Tripathi and Sridhar Iyer*

## Abstract

Communication, which intends to provide a link between any two people, is now moving towards man-to-machine and machine-to-machine connection for transferring different types of data. This transmission scenario, with and ever expanding number of active and passive users, lays the foundation to variety of communication protocols owing to the different types of data which is involved in the process. Within this ever expanding communication arena, Middle-ware can be thought of as a set of hardware and software which is used to connect different platforms with the end-users that are increasing in number day-by-day, with a possible wide spread over any region spanning from few meters to several kilometers. IEEE 802.11 is the set of standards which guides the wireless technology for device implementation and demands seamless integration across the entire protocol stack. This in turn demands an overview of the middleware architecture in broader perspective. This chapter explores the concept of middleware in the existing communication scenario, current trends and future scope.

**Keywords:** Middleware architecture, Communication protocols, IEEE 802.11, Network layer, Application layer, TPM, RPC, MOM, ORB

## 1. Introduction

Widespread usage of communication technology, intended to transfer information from one person to another, is moving rapidly towards information exchange between man and machine. The advent of computer along with digital formatted communication in light of advanced algorithms, low cost VLSI (Very Large Scale Integration) and high efficiency in computation paves the way for massive information exchange among individuals.

Simultaneously, ubiquitous computing paradigm with sensor based data communication presents unique man-to-machine information exchange which not only spread across large area but is also found to be a source of voluminous data, that needs to be carefully segregated and segmented for figuring out intelligence from a large pile of existing data.

Conventionally, middleware is said to be collection of algorithms, components and devices which enables the information exchange between different entities. As per the OSI model, middleware can be placed between the transport and the application layer, as shown in **Figure 1**.

In existing literature, the concept of middleware was discussed starting in the year 1968 in a report of the North Atlantic Treaty Organization [1], where it was placed between the application programs and the service routines. This paradigm

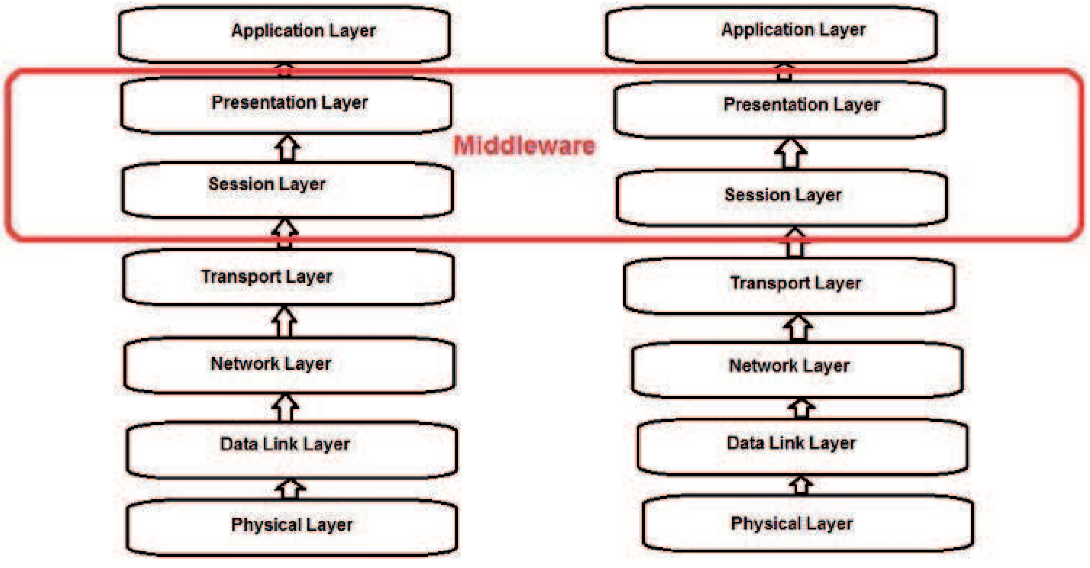


**Figure 1.**
*Location of middleware in OSI reference model.*

was quite pragmatic due to its ability to interconnect new components with the existing hardware within the same distributed system.

With the advent of the recent 5G technologies along-with strong data networking capability, a plethora of innovative sensor-based application are appearing over the horizon. This motivates us to look at the middleware architecture at a microscopic level as it involves different sensors with their applications without considering the interconnection involved i.e. it hides the heterogeneous nature of the underlying sensing data to yield the support system with the help of unified interface for the final application. This results in an easy access of the network by sensors as well as different applications, which can call the sensing network with the middleware open interface. This helps in the reduction of the design and the development cost with improved efficiency.

## 2. History and motivation

The initial development of the computer was based on the premise and promise of high-speed calculations which was iterative in nature. It was initially proposed through mechanical gear system but was found to be excessively time consuming. With the advent of vacuum tubes, the hurdle of timing is attended to and further, investigation in the field of solid state electronics which results in devices such as, diodes, transistors and operation amplifiers paves way for low power consuming, portable size devices which are termed as analog computers. This was the first era of computers [2]. The more robust, fast and complex Integrated Circuits (IC) were responsible for the design and development of digital computers which are having tremendous capability to perform high speed and complex calculations [3].

With the elaboration of theoretical concepts and their implementation in terms of ideas such as, distributed computing, abstraction, object-oriented programming, etc., results in the coding of complex software programs to an easily solvable problem at hand. At the same time, in order to reduce the design and development cost of the software and its interaction with the hardware, the concept of re-usability comes into software design paradigm [4].

The parallel progress in networking paradigm, which was simply started with possible interconnection of machines, later resulted in the development of OSI model to yield a layered structure of communication among different machines [5]. The layered architecture is profoundly based on the software abstraction model which felicitates a very specific section/property of the network to change without hampering the process of communication among other machines through different layers. This paradigm was the primary basis of middle ware architecture which was placing it between the transport and the application layer, as shown in **Figure 1**.

With the expansion of networking paradigm, Middleware architecture found its place among the widely used concepts, especially in the era of plug and play design methodology. In the subsequent section, we will look at the different attributes which Middleware workable.

### 2.1 Features of middleware

As discussed in the introduction, middleware is a set protocol for data exchange between the transport and the application layer. It helps different machines to share the data with the network and vice versa without considering the heterogeneity of the underlying data from both sides, and results in reduced development cost and improvement in the efficiency [6].

The following features are the major attributes of the middleware architecture:

1. From the Software management perspective, Middleware is found to be more and more integrated into the operating system, which results in the application evolved in a machine to be safely ported on the network as well, wherein the network is acting as an up-scaled version of the machine. Also, this results in resource management mechanism based on service quality and the flexible configuration capability [7].

2. With the widespread availability of the Internet, middleware architecture provides web-based services and resource sharing capacity, making middleware architecture almost like connecting glue which supports in the running of the application software successfully. Also, the resource sharing methods based on Internet services are more universal, cost effective and efficient.

3. Middleware architecture has transformed the conventional spoke and wheel system into the distributed system by combining different technologies such as, cloud computing, big data and virtualization, which provides the capacity to integrate different resources and yields more robust service capability. With this architecture, one can solve the issue of data storage, processing and transmission among the different internetworked systems.

Depending on the different attributes, the middleware architecture can be classified into four types [8, 9].

1. Transaction Processing Monitors (TPM)

2. Remote Procedure Call (RPC)

3. Message-Oriented Middleware (MOM)

4. Object Request Brokers (ORB)

These types are based on the varied services offered by the specific process.

**Transaction Processing Monitors (TPM)** is designed to monitor the successful transactions from one stage to another stage. In case of any error, TPM takes an appropriate action to rectify the error. A TPM supports optimal resource sharing among the applications with the following functionalities:

- Monitoring operation/transactions
- Managing queues
- Coordination among resources
- Creating new processes on requirement
- Secure access to services
- Wrapping data messages into messages
- Unwrapping messages into data packets
- Handling errors
- Hiding the details of inter-process communications from programmer

**Remote Procedure Call (RPC)** is an inter-process communication facility generally used in a client server based model with the following functionalities:

- Supports process-oriented or thread-oriented models
- Hiding details of inter-process communications from programmer
- Useful in local environment as well as distributed environment
- Performance improvement can be achieved by omitting unwanted protocol layers

**Message-Oriented Middleware (MOM)** is an asynchronous technique that passes the messages between transmitting and receiving application with a communication channel. Its asynchronous nature makes the applications decoupled from each other as MOM is responsible for message management system.

**Object Request Brokers (ORB)** acts like a broker between a client request for a service from a distributed object and the completion of that request with following functionalities:

- Life cycle service
- Persistence service
- Naming service
- Event Service
- Concurrency control service

*DOI: http://dx.doi.org/10.5772/intechopen.97124*

- Transaction service
- Relationship service
- Externalization service
- Query service

In view of the above-mentioned services offered by the different types of middleware, following are the major attribute of services offered by any middleware structure [8]:

- **Presentation management**: Forms manager, graphics manager, hypermedia linker, and printing manager.
- **Computation**: Sorting, math services, internationalization services (for character and string manipulation), data converters, and time services.
- **Information management**: Directory server, log manager, file manager, record manager, relational database system, object-oriented database system, repository manager.
- **Communications**: Peer-to-peer messaging, remote procedure call, message queuing, electronic mail, electronic data interchange.
- **Control**: Thread manager, transaction manager, resource broker, fine grained request scheduler, coarse-grained job scheduler.
- **System management**: Event notification service, accounting service, configuration manager, software installation manager, fault detector, coordinator, authentication service, auditing service, encryption service, access controller.

## 3. IEEE 802.11 adaptation with middleware

The emerging trend of wireless technology and the associated innovative applications has changed the communication landscape drastically. Now, the reduced cost of data, high computation power of smartphones and the 5G enabled sensor technology is the major driving force behind the widespread adaptation of network services such as, map enabled movement, shipment tracking, and interactive gaming such as, Pokemon.

Inherently, multimedia services are found to be wideband in nature which, with the unpredictable channel characteristic of wireless medium, place a challenging condition to maintain reliable communication with a predefined Quality of Service (QoS).

IEEE 802.11 standard, proposed by IEEE for local area network (LAN) protocol, specifies the physical layer (PHY) and media access control (MAC) protocols for the implementation of wireless local area network (WLAN) communication in the frequency bands such as, 2.4 GHz, 5 GHz, 6 GHz, and 60 GHz. Presently, various tributaries of IEEE 802.11 are framing our day-to-day communication across the world. As per the convention, IEEE 802.11 standards are defined for physical and data link layer of the OSI data communication network protocol and become the de facto standard for wireless communication.

Moreover, the heterogeneous nature of different services such as, Wi-max, Wi-Fi, Bluetooth, and many more, with their unique data link layer presentation have their own quality control mechanism for data transmission. This results in a complex connection strategy management at network, data link, and physical layer.

In comparison, at the application layer, the QoS management parameters such as, semantic, presentation etc., have their own constraints to be followed. These constraints may not be followed every time due to resource limitations at the physical layer or multiplexing issue due to the fluctuating number of users sharing the resource pool in the end system and the network.

In the process of communication, wireless channel demands adaptive QoS implementation for peer-to-peer communication, due to its dynamic behavior. This places a bound on the middleware to manage the communication as it connects the lower layers of the communication protocol (which actually perform the communication at the data level) to the application layer (which is responsible for communication in the correct semantic and form) [10, 11]. Thus, the unique position of middleware demands a monitoring as well as an adaptive perspective such that the desired QoS requirements can be maintained at the application layer without disturbing the underneath communication.

This QoS maintenance requires a two-fold strategy, first is the monitoring of application performance and the second is adaptation of service to maintain pre-specified quality of service [11]. This adaptation strategy requires a perfect synchronization of middleware level with lower level of protocols.

These conditions lay the foundation for the middleware control framework. The middleware control framework has three primary concerns to address [12]:

- Coordinate the adaptation of all the concurrent application tasks in the end system globally i.e., maintain fairness in the architecture.

- Increase the adaptation effectiveness to maintain the QoS.

- To monitor on-the-fly dynamics in the heterogeneous environment to achieve optimum control over the smooth functioning of the applications.

### 3.1 IEEE 802.11 - an interesting journey

IEEE 802 project was aimed to establish the standards for physical layer (PHY) and medium access control (MAC) layer to support deployment of the local area network. The first candidate was IEEE 802.3, popularly known as Ethernet. It was a wired connection mechanism to connect devices based on carrier sense multiple access with collision detection (CSMA/CD) mechanism. Its wide acceptability with industry and home users, motivates the researcher to look for its replication in wireless domain as well which yields the IEEE 802.11 standard [13, 14].

Due to inherent unpredictable nature of radio/wireless medium paves the way for numerous deliverables of IEEE 802.11 standards, which have been flourished over the last twenty years [15, 16], such as 802.11a, 802.11e, 802.11f, 802.11 t and so on. Readers are encouraged to refer [17] to explore the need and their solutions under IEEE 802.11 horizon.

### 3.2 Middleware in IEEE 802.11 environment

Under the middleware architecture a brief account of various cases has been summarized below.

Authors in [18] address the issue of mobility management in increasing integration of Internet with telecommunication network which give rise to distributed computing environment. Authors proposed three mobile computing services by incorporating user virtual environment (UVE), mobile virtual terminal (MVT), and virtual resource management (VRM) based on mobility middleware solution for mobile agent.

In [19], the Authors illustrate Quality of Service (QoS) maintenance for multimedia applications over wireless network which are characterized by their limited bandwidth. Authors proposed a novel two level QoS architecture by providing service differentiation at network level and service adaptation at the middleware level. Authors validate their results with experiments and show that specific QoS levels for multimedia applications can be optimally achieved in IEEE 802.11 based wireless network.

Costa et al. in their path breaking work proposed the real time-WiFi architecture to address the issues faced by IEEE 802.11 networks in high density industrial environment [20]. Authors compare the performance of proposed architecture against the standard distributed coordination function (DCF), point coordination function (PCF), hybrid coordination function (HCF) controlled channel access (HCCA) and enhanced distributed channel access (EDCA) medium access control mechanism. Authors used a realistic error-prone model to monitor the impact of message losses in the real time Wi-Fi architecture and their result confirm that the proposed architecture performs better as compared to existing IEEE 802.11 standard mechanisms. Their architecture also offers almost consistent access delay which is one of the major requirements for real time applications.

Authors in [21] use context meta-information to improve the system performance by describing a context management middleware that can successfully handle context irrespective of the execution environment's heterogeneity.

Cruz et al. in [22], reviewed the middleware framework for Internet of Things (IoT) from software perspective. Authors rigorously explored the existing literature and analyzed the reference model for IoT platforms and proposed the basic security feature for this software. Authors also detailed the difficulties in achieving and enforcing a universal standard for middleware in the IoT structure.

Authors in [23], present a fundamental analysis to quantify the goodput performance parameter for IoT. Authors show the closed form expression of goodput as a function of the data payload length, frame retry count, data rate of transmission and wireless channel condition. Authors proposed a novel link adaptation scheme for MAC protocol data units for known wireless channel model.

Increasing use of mobile devices for location sharing applications such as Google map, OLA, Uber, pose a challenge of maintaining adequate user privacy with location sharing services and exchange of information across high heterogeneity among connecting technologies and devices. Authors in [24] proposed a middleware prototype to answer these challenges with two level proxy-based architecture as a solution.

Hamidreza et al. in [25], proposed service-oriented architecture for middleware to resolve the issue of heterogeneity among various sensors in IEEE 802.15 based wireless sensor network.

Authors in [26] discuss different types of sensor network applications with overview of related middleware and infer that none of the existing approaches can provide all the management tools required by sensor network applications. Authors showcase their new middleware MILAN with sensor-based health monitoring system.

In [27], the authors explored the concept of inter-vehicular communications. The field of vehicle to infrastructure and vehicle to vehicle communications were undertaken as well. This work provides detailed account of underlying technology under each layer with rich resource references.

Authors in [28], address the combination of vehicular ad-hoc networks (VANETs) with the social Internet of things (SIoT). Their work describes two fold relations which can be established between the vehicles and between the vehicle and road side unit (RSU). Authors proposed a social Internet of vehicle middleware to incorporate the functionalities of the intelligent transportation systems station architecture, defined by ISO and ETSI standards to integrate VANET with SIoT. They present their proof of concept with simulation results.

Pease et al. in [29] present an adaptive middleware methodology to provide robust mission critical/ military communication by providing timely MANET communications with predictive selection and dynamic contention reduction, without going for invasive protocol modification. To address the issue they proposed a novel Real-time Optimized Ad hoc Middleware based architecture (ROAM). They demonstrate the adaptability, scalability of the architecture along-with the capability to bound maximum delay, jitter and packet loss in complex and dynamic MANET's with extensive simulations.

In [30], the authors elaborated a novel mobile collaboration architecture (MoCA), a service oriented middleware architecture which support the development and deployment of distributed context-aware applications for mobile users. Authors explained the compatibility of proposed MoCA with existing software engineering principles responsible for design and implementation of context aware applications. Authors also present different prototype applications that have been developed on the top of MoCA.

Authors in [10] addressed end to end delays and security issues in application implementation. Authors proposed an integrated solution with middleware adaptation to provide tunable delay and security support according to network condition. To Support the proof of concept, authors perform test-bed experiments to showcase successful meeting of delay and security requirements in IEEE 802.11 based wireless environment.

## 4. Conclusion

Middleware architecture defines the connection protocol between the network layer and the application layer. In view of the on-going advances in the mobile communication, there is a requirement of a better understanding about Middleware functionality with IEEE 802.11 protocols which are responsible for the design rules of modern wireless communication.

In this chapter, we addressed the key functionalities of the middleware architecture in addition to its adaptation to the IEEE 802.11 protocols.

In the current scenario, where the need of ubiquitous connectivity is reality, the need of minimum end to end delay with almost no loss in data i.e. maintaining stringent Quality of Service at every end, is a challenging task.

This monograph will be helpful for the researchers to investigate middleware architecture in IEEE 802.11 framework to deliver robust design solution in wireless network.

## Conflict of interest

"The authors declare no conflict of interest."

## Author details

Rochak Bajpai[1]*, Atul Bansal[2], Jyoti Tripathi[3] and Sridhar Iyer[4]

1 KIET Group of Institutions, Delhi, NCR, India

2 GLA University, Mathura, India

3 G.B. Pant Government Engineering College, Okhla, New Delhi, India

4 S.G. Balekundri Institute of Technology, Belagavi, India

*Address all correspondence to: rochakbajpai@gmail.com

## References


[1] Bauer F.L., Bolliet L., Helms H. J., "Software Engineering: Report on a conference sponsored by the NATO science committee", Garmisch, Germany, Tech. Rep. Jan. 1969. \url{ http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF} [Accessed: 04 January 2021]

[2] Mindell DA. "Between human and machine: feedback, control, and computing before cybernetics". JHU Press; 2002 Oct 11.

[3] Graham S, Baliga G, Kumar PR. "Abstractions, architecture, mechanisms, and a middleware for networked control". IEEE Transactions on Automatic Control. 2009 Jun 30;54(7):1490-503.

[4] Prieto-Diaz R. "Status report: Software reusability". IEEE software. 1993 May;10(3):61-6.

[5] Kumar S, Dalal S, Dixit V. "The OSI model: Overview on the seven layers of computer networks". International Journal of Computer Science and Information Technology Research. 2014 Jul;2(3):461-6.

[6] Dong H. "Middleware Technologies". Journal of Anhui Vocational College of Electronics \& Information Technology. 2006:04.

[7] Lu ZJ, Yan Z. "Middleware Technology Research and Interface Design in Ubiquitous Network". In2015 Seventh IEEE International Conference on Measuring Technology and Mechatronics Automation 2015 Jun 13 (pp. 659-662).

[8] Bernstein PA. "Middleware: a model for distributed system services". Communications of the ACM. 1996 Feb 1;39(2):86-98.

[9] Tian Y, Geiger JV, Su H, Kumar SV, Houser PR. "Middleware-based sensor web integration". IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2010 Jun 21;3(4):467-72.

[10] Li B, Nahrstedt K. "A control-based middleware framework for quality-of-service adaptations". IEEE journal on selected areas in communications. 1999 Sep;17(9):1632-50.

[11] Ong CS, Xue Y, Nahrstedt K. "A middleware for service adaptation in differentiated 802.11 wireless networks". InProceedings. 2004 12th IEEE International Conference on Networks (ICON 2004)(IEEE Cat. No. 04EX955) 2004 Nov 19 (Vol. 1, pp. 364-368). IEEE.

[12] Li B, Nahrstedt K. A control-based middleware framework for quality-of-service adaptations. IEEE journal on selected areas in communications. 1999 Sep;17(9):1632-50.

[13] Tanenbaum AS, Wetherall D. Computer networks. Prentice-Hall international editions. 1996 Mar:I-XVII.

[14] Peterson LL, Davie BS. Computer networks: a systems approach. Elsevier; 2007 Apr 16.

[15] Stallings W. Local networks. ACM Computing Surveys (CSUR). 1984 Mar 29;16(1):3-41.

[16] Stallings W. Data and computer communications. Pearson Education India; 2007.

[17] Hiertz GR, Denteneer D, Stibor L, Zang Y, Costa XP, Walke B. The IEEE 802.11 universe. IEEE Communications Magazine. 2010 Jan 22;48(1):62-70.

[18] Bellavista P, Corradi A, Stefanelli C. Mobile agent middleware for mobile computing. Computer. 2001 Mar;34(3):73-81.

[19] Ong CS, Xue Y, Nahrstedt K. A middleware for service adaptation in differentiated 802.11 wireless networks. InProceedings. 2004 12th IEEE International Conference on Networks (ICON 2004)(IEEE Cat. No. 04EX955) 2004 Nov 19 (Vol. 1, pp. 364-368). IEEE.

[20] Costa R, Lau J, Portugal P, Vasques F, Moraes R. Handling real-time communication in infrastructured IEEE 802.11 wireless networks: The RT-WiFi approach. Journal of Communications and Networks. 2019 May 29;21(3):319-34.

[21] da Rocha RC, Endler M. Middleware: Context management in heterogeneous, evolving ubiquitous environments. IEEE Distributed Systems Online. 2006 May 15;7(4).

[22] da Cruz MA, Rodrigues JJ, Al-Muhtadi J, Korotaev VV, de Albuquerque VH. A reference model for internet of things middleware. IEEE Internet of Things Journal. 2018 Jan 23;5(2):871-83.

[23] Qiao D, Choi S, Shin KG. Goodput analysis and link adaptation for IEEE 802.11 a wirelessLANs. IEEE transactions on Mobile Computing. 2002 Oct;1(4):278-92.

[24] Bellavista P, Corradi A, Giannelli C. Efficiently managing location information with privacy requirements in wi-fi networks: a middleware approach. In2005 2nd International Symposium on Wireless Communication Systems 2005 Sep 5 (pp. 91-95). IEEE.

[25] Abangar H, Barnaghi P, Moessner K, Nnaemego A, Balaskandan K, Tafazolli R. A service oriented middleware architecture for wireless sensor networks. InProceedings of future network and mobile summit conference 2010 Jun.

[26] Heinzelman WB, Murphy AL, Carvalho HS, Perillo MA. Middleware to support sensor network applications. IEEE network. 2004 Jun 28;18(1):6-14.

[27] Jawhar I, Mohamed N, Zhang L. Inter-vehicular communication systems, protocols and middleware. In2010 IEEE Fifth International Conference on Networking, Architecture, and Storage 2010 Jul 15 (pp. 282-287). IEEE.

[28] Nitti M, Girau R, Floris A, Atzori L. On adding the social dimension to the internet of vehicles: Friendship and middleware. In2014 IEEE international black sea conference on communications and networking (BlackSeaCom) 2014 May 27 (pp. 134-138). IEEE.

[29] Pease SG, Phillips IW, Guan L. Adaptive intelligent middleware architecture for mobile real-time communications. IEEE Transactions on Mobile Computing. 2015 Apr 27;15(3):572-85.

[30] Viterbo J, Sacramento V, Rocha R, Baptista G, Malcher M, Endler M. A middleware architecture for context-aware and location-based mobile applications. In2008 32nd Annual IEEE Software Engineering Workshop 2008 Oct 15 (pp. 52-61). IEEE.