

An Integrated Framework for Wireless Sensor Web Service and Its Performance Analysis Based on Queue Theory

Luqun Li^{1,1}

¹ Department of Computer Science of Shanghai Normal University, 100 Guilin Road, Shanghai, China 200233
liluqun@gmail.com

Abstract. Wireless sensor networks (WSNs) have unlimited and extensive potential application in different areas. Right now, how to integrate WSNs into the web service or grid computing framework is becoming an issue in related research areas. In this paper, we proposed an integrated web service framework for WSNs, and built mobile device hosted web service, moreover we built its corresponding queueing model and gave it performance analysis. Analysis and simulation show that our framework is practical in QoS guaranteed messages transmitting with different priority.

Keywords: web service; wireless sensor networks; QoS; queue theory; Little's Law

1 Introduction

Right now, wireless sensor networks (WSNs) are currently receiving significant research attention both in theory and application, due to their unlimited and extensive potential application in different areas[1]. WSNs serve as a role that bridges the gap between the physical and logical worlds, by gathering certain useful information from the physical world and transmitting that information to more powerful logical devices that can process it[2]. Each node in WSNs has its unique computing ability in its specific working area, moreover it may own wireless network connection with WLAN or Internet. It can act as a computing nodes in network based computing, such as grid computing and web service computing[1]. This seems to be one of the phenomena in Pervasive Computing era[1, 3, 4]. While as for the limited computing ability, storage, battery energy and wireless band width of WSNs, how to efficiently and QoS grantee integrate WSNs with the architecture of current web service or grid service[5, 6] has become an issue in web service integration application. In this paper, firstly, we analysis related research works on this issue then put forward an integrated framework for wireless sensor web service, then we build the differential service queueing model of this framework and give its performance analysis and numeral results. Finally, based on the results of performance analysis and numeral result, we give some optimization parameter selection in this framework.

2 Related Work and Challenges

In recent years, there has been a growing interest in sensor data collection and management, with the main focus on integrates WSNs with the current web service architecture framework. Till now, there are many research works on some specific application. These studies[1-11] provide effective techniques for sensor databases and query systems, but their main limitation is the problem scale, being within a single sensor network, besides studies above only give designs of the framework, most of studies did not provide their corresponding mathematics model and performance analysis. Another import fact is in real WSNs, there may exist many different classes

¹ This work was supported in part by the Shanghai Education Bureau Grant CL200652, SK200709 and PL531.

users, for example the administrators or common users, different classes users need differential service such as the latency of service. Due to these issues we think the following challenges need to be addressed:

- The framework of integrate WSNs with current web service;
- The mathematics model of the framework;
- How to evaluate the performance of the framework.

To address these key challenges, we have designed the corresponding framework and mathematics model.

3 Proposed Framework

Web service has become an integral part of many web applications in nowadays. Platform-independent, ubiquitous and easy access web services using common standardized protocol SOAP, WSDL, UDDI, have been one of the principal drivers behind this success.

In WSNs, it is impossible for all nodes to be connected with wired network or Internet, usually only the sink nodes in WSNs which have high battery energy will connect with wired network or Internet (See Fig.1).

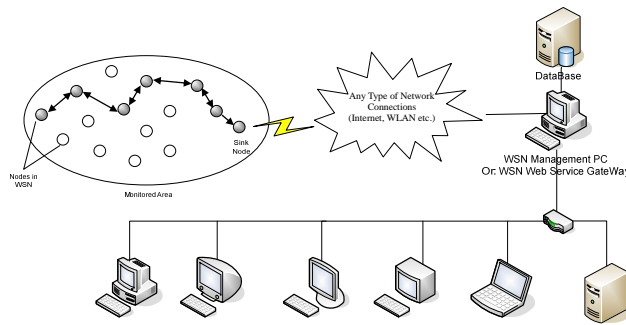


Fig.1 WSNs Connected With Wired Network

Different from the wired web service provider, web service based on wireless sensor nodes (such as Mica2, Zigbee etc) which only own constraint computing resources and battery, and very narrow band of wireless network. Moreover, to save the battery, if there is no request for sensor data; nodes in WSNs are usually in battery saving state. Besides these issues, XML and SOAP are not efficient protocol for WSNs nodes to transmit messages, while raw binary code related protocols are still dominated protocols for these devices. Concern on the unique characteristics of WSNs and the standard web service architecture, we prompted the following integrated web service framework for WSNs framework for mobile device hosted web service (See Fig.2).

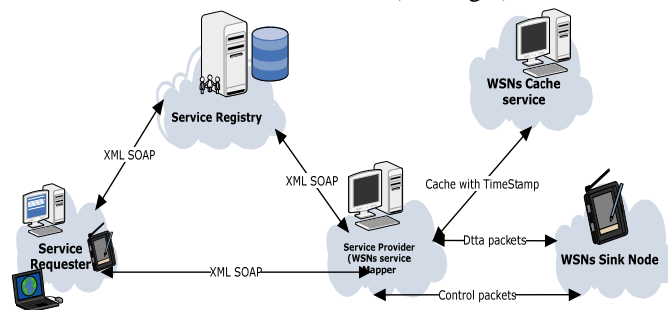


Fig.2 Integrated Web Service Framework for WSNs

We use the following abbreviation to denote each role in the framework above:

SG : denotes Service Registry; SR : denotes Service Requester;
 WSM : denotes WSNs service Mapper;
 $WSNd$: denotes the sink node in WSNs;
 WCS : denotes WSNs Cache Service.

In this framework, we can see that Service Registry (SG) and Service Requester (SR) are still the traditional roles in web service model. Different from the standard web service model, we introduce a new role which is WSNs service Mapper (WSM). WSM acts as a hybrid role.

From the Service Requester viewpoint, WSM is a standard web service provider, roles on the left side of the framework communicate with each other by SOAP or XML protocol. Though most nodes in WSNs have wireless network connection, they may not own valid IP address; they can not directly provide service just like a server in internet. WSM seems to be one of the most practical approach to solve the problem, it is like a gateway to bridge WSNs and wired network, so WSM is an essential role in this framework.

In our framework above, on the right side, WSM acts as a web service consumer, it can also be taken as a wrapper for sink node in WSNs hosted web service, WSM is usually hosted on a wired computer, it accepts the web service request by standard web service protocol, it requests web service from sink node of WSNs by any protocols, we denote it by $WSNd$.

Because each WSN may need to be managed, the user of WSNs may be classified into different privilege levels, to make the problem simple, the users are classified into two classes, or the administrators and the common users. Usually the administrators send control packet to sink nodes of WSNs to regulate the states of WSNs, the common user only request for the data from $WSNd$.

Besides, to enhance the throughput of the system and save the battery energy of $WSNd$, we also introduce a web service cache system WCS to reduce unnecessary repeated request to $WSNd$.

As for, in this paper we only focus on the framework and its performance analysis, we will not go any further in program coding, you can see the details in program code implementation of our system in [1].

3 Queuing Model for the Framework

To analysis the performance of the integrated web service in Fig.2, we build the following queueing mode (See Fig.3)

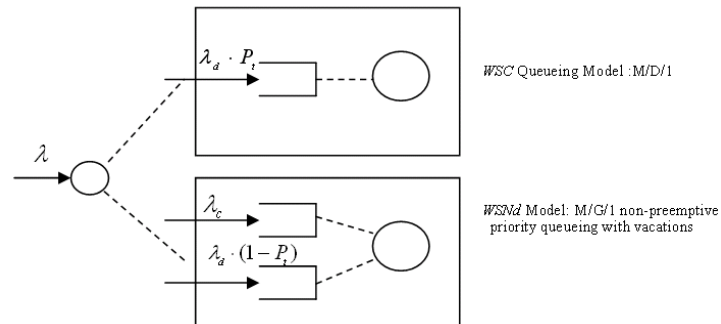


Fig.3 Queue Model for the Framework

Actually, there are two queues in the queueing model above, and they are:

- Queue.1 The first queue is for WCS , the cache system is only for WSNs data packet, WSNs cache system is usually a data search operation, and we use M/D/1 for this queueing model. Web service request messages come to WCS queue at the rate of $\lambda_d \cdot P_i$, where

λ_d is the rate of data request messages, P_t is probability of getting data from cache service *WSC*.

- Queue.2 This queue is for *WSNd*. This queue is work for both WSNs control and data packet request, as for the service time of *WSNd* is usually with general distribution and it may be in the state of energy saving (in vacations states), we use M/G/1 non-preemptive priority with vacations for this queueing model.

All *WSNd* web service request message are firstly sent to *WSM*. *WSM* will check the message type and the time stamp to determine which queue the request should be forward to process.

In this model, we assume web service request message types are classified in to n priority classes. Messages of each priority class $i (i = 1, 2, \dots, n)$ arrive according to a Poisson process with rate λ_i and to be served by *WSNd* with a general service time distribution of mean \bar{x}_i and second moment \bar{x}_i^2 .

To make the problem simple, we assume there are two priority classes in our model, the first one is control message, which is usually the control message sent by the administrators or some urgent message request, it comes with the rate of λ_c ; the second one is data message, it comes with the rate of $\lambda_e = \lambda_d \cdot (1 - P_t)$. The arrival λ_c and λ_d are assumed to be independent of each other and service process. Note that the rate of all messages comes at $\lambda = \lambda_c + \lambda_d$.

λ_c is to be served by *WSNd* with a general service time distribution of mean \bar{x}_c and second moment \bar{x}_c^2 .

λ_e is to be served by *WSNd* with a general service time distribution of mean \bar{x}_e and second moment \bar{x}_e^2 .

Another import thing to be noted is *WSNd* may be in energy saving state or so-called a "vacation". In this state *WSNd* does not process any requests. Assume that v_1, v_2, \dots are the residual of *WSNd*'s successive vacation time. The mean of vacation time v_1, v_2, \dots is \bar{V} , and the second moment is \bar{V}^2 .

Then we can summary the parameters and their relationships by the followings:

λ_c : denotes the rate of control messages to *WSNd*;

λ_e : denotes the rate of data messages to *WSNd*

P_t : denotes the probability of data message to *WSC*

ρ_c : is called the traffic intensity or utilization factor for control message to process;

ρ_e : is called the traffic intensity or utilization factor for data message to process;

ρ : denotes the traffic intensity of the system;

μ_c : denotes the service rate to process control messages;

μ_e : denotes the service rate of to process data messages;

\bar{x} : denotes the average service time of the system;

\bar{x}^2 : denotes the second moment of the system;

\bar{V} : denotes the mean of vacation time v_1, v_2, \dots ;

$\overline{V^2}$: denotes the second moment of v_1, v_2, \dots

We can deduce the relationship among these parameters above in the followings:

$$\lambda = \lambda_c + \lambda_d = \lambda_c + \lambda_d \cdot P_t + \lambda_e, \lambda_e = \lambda_d \cdot (1 - P_t);$$

$$\rho_c = \lambda_c \cdot \overline{x_c}, \rho_e = \lambda_e \cdot \overline{x_e}, \rho = \frac{\lambda}{\mu} = \rho_c + \rho_e;$$

$$\overline{x} = \frac{\lambda_c}{\lambda} \cdot \overline{x_c} + \frac{\lambda_e}{\lambda} \cdot \overline{x_e}, \overline{x^2} = \frac{\lambda_c}{\lambda} \cdot \overline{x_c^2} + \frac{\lambda_e}{\lambda} \cdot \overline{x_e^2}$$

4 Analysis on the Queueing Model

As for Queue.1 above is a $M / D / 1$ queueing system, this queue is rather easy, we will not go any further. Now, we will focus on Queue.2.

First, we analysis the residual service time R for all the messages in the system. By Fig.4 we can get that R equals the mean service time of all message with different priority plus the mean service in vocation time ($WSNd$ is in power saving state), we can simple denote it by:

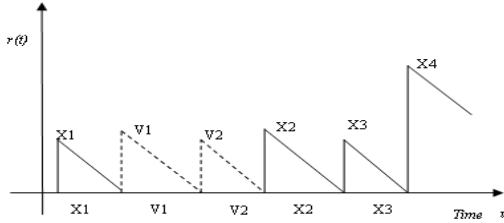


Fig.4 Residual Service Time for All Messages

K is the message with higher priority than j in the queue before j



Fig.5 Messages in Queue.2

$$R = \overline{ServiceTime} + \overline{ServiceInVacationTime};$$

$$\text{where, } \overline{ServiceTime} = \sum_{k=1}^n \rho_k \left(\frac{\overline{x_k^2}}{2\overline{x_k}} \right) = \frac{1}{2} \sum_{k=1}^n \lambda_k \overline{x_k^2},$$

$$\overline{ServiceInVacationTime} = \frac{\overline{V^2}}{2 \cdot \overline{V}}$$

Where, $\overline{ServiceTime}$ is the system mean service time, $\overline{ServiceInVacationTime}$ is the mean time that $WSNd$ is in power saving state; and j is current message that just is in queue, k is the message with higher priority than j in the queue before j (See Fig.5).

From analysis above, we can directly use the conclusions in $M / G / 1$ [12], and arrive at the waiting time of each message with different priority :

$$W_i = \frac{R}{(1 - \rho_1 - \rho_2 - \dots - \rho_{i-1})(1 - 1 - \rho_1 - \rho_2 - \dots - \rho_i)} \quad (1)$$

where W_i is the waiting time of message to $WSNd$ with priority i , and ρ_i is the traffic intensity or utilization factor for message to $WSNd$ with priority i .

Then from equation (1), according to **Little's Law** in queue theory, we can get the average number of messages N_i with the same priority i in their waiting queue:

$$N_i = \lambda_i \cdot W_i = \frac{\lambda_i \cdot R}{(1 - \rho_1 - \rho_2 - \dots - \rho_{i-1})(1 - 1 - \rho_1 - \rho_2 - \dots - \rho_i)} \quad (2)$$

The total time T_i for each message with priority i spent in Queue.2 is:

$$T_i = W_i + \bar{x}_i = \frac{R}{(1 - \rho_1 - \rho_2 - \dots - \rho_{i-1})(1 - 1 - \rho_1 - \rho_2 - \dots - \rho_i)} + \bar{x}_i \quad (3)$$

The total number of all messages N in Queue.2 is:

$$N = \sum_{k=1}^n N_k + \rho = \sum_{k=1}^n \left(\frac{\lambda_k \cdot R}{(1 - \rho_1 - \rho_2 - \dots - \rho_{k-1})(1 - 1 - \rho_1 - \rho_2 - \dots - \rho_k)} \right) + \rho \quad (4)$$

In summary, we can evaluate the performance of our framework by W_i , N_i and T_i .

5 Numerical Results and Analysis

To analysis the results that we have deduced above, we give the following initial parameters in the framework:

- The mean data message process time $\bar{x}_d = 0.02$, with variance $\sigma_d^2 = 0.05$ and the second moment $\bar{x}_d^2 = (\bar{x}_d)^2 + \sigma_d^2$;
- The mean control data message process time $\bar{x}_c = 0.002$, with variance $\sigma_c^2 = 0.02$ and the second moment $\bar{x}_c^2 = (\bar{x}_c)^2 + \sigma_c^2$;
- The mean control data message process time $\bar{V} = 0.06$, with variance $\sigma_v^2 = 0.01$ and the second moment $\bar{V}^2 = (\bar{V})^2 + \sigma_v^2$; to increase the efficiency of system, we change the $\bar{V}^2 = ((\bar{V})^2 + \sigma_v^2) / \lambda$, it means intelligent in power saving state;
- $P_t = 0.1$;

According equation (1)(2)(3)(4), by increase the value of λ , we will also get increased traffic intensity of the system ρ , the we can get the queue size and wait time of each message with different priority, see Fig.6 and Fig.7.

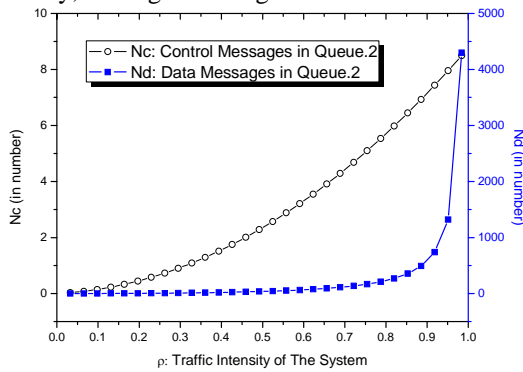


Fig.6 Queue size of N_c and N_d with ρ

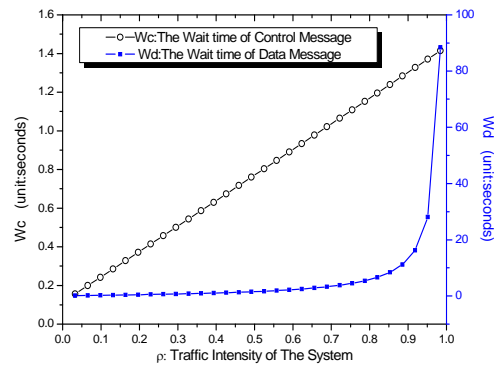
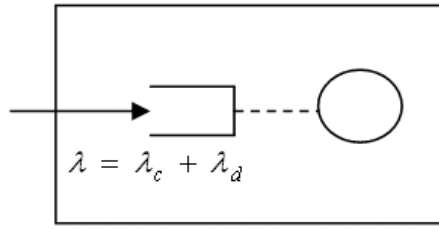


Fig.7 Wait time of W_c and W_d with ρ

Fig.6 shows that with the increasing of ρ , N_c will also increase, but the queue size will be below than 9; while N_d will increase very fast, and the queue size nearly 4300;

Fig.7 shows that with the increasing of ρ , W_c will also increase, but the wait time will be below than 1.4 second; while W_d will increase very fast, and the wait time will below 84 second;

We can see that latency of the control message to be processed will be guaranteed.



Queueing Model: M/G/1 with vacation without Priority

Fig.8 Queueing Model M/G/1 with Vacation without Priority

To compare the framework we proposed in Fig.3 with the traditional queueing model M/G/1 with vacation without priority in Fig.8, we run the simulation with the same initial parameters above, then we get Fig.9 and Fig.10.

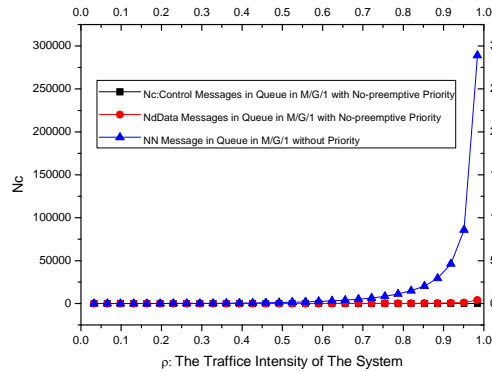


Fig.9 Queue size in different model with ρ

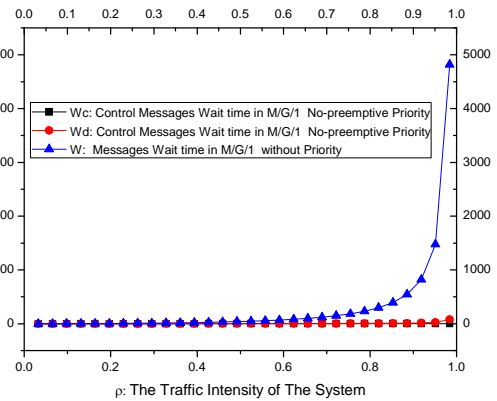


Fig.10 Wait time in different model with ρ

Fig.9 shows that in our framework in Fig.3 with the increasing of ρ , N_c will also increase, but the queue size will be below than 9; while N_d will increase very fast, and the queue size nearly 4300; in the traditional model in Fig.8, the queue size will increase very fast, and the queue size is very large;

Fig.10 shows that with the increasing of ρ , W_c will also increase, but the wait time will be below than 1.4 second; while W_d will increase very fast, and the wait time will below 84 second; in the traditional model in Fig.8, the wait time size will increase to 4800;

With the queueing model in Fig.3 and $WSNd$ is intelligent in power saving state, we can make a conclusion that the performance of the framework we proposed is much better than the traditional queueing model in Fig.8

6 CONCLUSIONS

In this paper, we introduced an integrated framework for wireless sensor web service, then, based on queue theory, we give its performance analysis. Analysis and simulation show that our framework is QoS guaranteed for different messages with different priority, and the performance of our is much better than the traditional queueing model in Fig.8.

References

- [1]. Li, L. and M. Li, The Study on Mobile Web Service Computing for Data Collecting. 2004 International Conference on Communications, Circuits and Systems IEEE., 2004. Volume II: p. P1497~1501.
- [2]. Yu, Y., L. J. and Rittle, Supporting Concurrent Applications in Wireless Sensor Networks. SenSys'06, November 1-3, 2006, Boulder, Colorado, USA., 2006: p. P139-152.
- [3]. Whitehouse, K., F. Zhao, and J. Liu, Semantic streams: A framework for composable semantic interpretation of sensor data. Proc. European Workshop on Wireless Sensor Networks (EWSN'06), Zurich, Switzerland (Feb. 2006).
- [4]. Zhu, F., M.W. Mutka, and L.M. Ni, Service discovery in pervasive computing environments. Pervasive Computing, IEEE, 2005. 4(4): p. 81-90.
- [5]. Delicato, F.C., et al., A flexible web service based architecture for wireless sensor networks. Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on, 2003: p. 730-735.
- [6]. Gaynor, M., S.L. Moulton, and M. Welsh, Integrating Wireless Sensor Networks with the Grid. 2004.
- [7]. Cardell-Oliver, R., et al., Field Testing a Wireless Sensor Network for Reactive Environmental Monitoring. Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 2004.
- [8]. Karl, H. and A. Willig, Protocols and Architectures for Wireless Sensor Networks. 2005: Wiley.
- [9]. Peng, R., K.A. Hua, and G.L. Hamza-Lup, A Web services environment for Internet-scale sensor computing. Services Computing, 2004.(SCC 2004). Proceedings. 2004 IEEE International Conference on, 2004: p. 101-108.
- [10]. Shi, J. and W. Liu, A service-oriented model for wireless sensor networks with Internet. Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on, 2005: p. 1045-1049.
- [11]. Woo, A., et al., A spreadsheet approach to programming and managing sensor networks. Proceedings of the fifth international conference on Information processing in sensor networks, 2006: p. 424-431.
- [12]. Hock, N.C., Queueing Modelling Fundamentals. JOHN WILEY & SONS Chichester. New York. Brisbane. Toronto. Singapore, 1997.
- [13]. Broll, G., et al., Supporting Mobile Service Usage through Physical Mobile Interaction. Fifth Annual IEEE International Conference on Pervasive Computing and Communication (PerCom 2007), White Plains, NY, USA, 2007.
- [14]. Dorn, C. and S. Dustdar, Sharing hierarchical context for mobile web services. Distributed and Parallel Databases, 2007. 21(1): p. 85-111.
- [15]. Khoi Anh Phan, PZahir Tari, and P. Bertok, A benchmark on soap's transport protocols performance for mobile applications. Proceedings of the 2006 ACM symposium on Applied computing 2006 (SESSION: Mobile computing and applications (MCA)): p. 1139 - 1144
- [16]. Oh, S. and G.C. Fox, Optimizing Web Service messaging performance in mobile computing. Future Generation Computer Systems, 2007. 23(4): p. 623-632.
- [17]. Park, Y.H., Method for supplying a mobile web service by transceiving xml data based an soap and a system therefo, Pantech Co Ltd.
- [18]. Sakkopoulos, E., M. Lytras, and A. Tsakalidis, Adaptive mobile web services facilitate communication and learning Internet technologies. Education, IEEE Transactions on, 2006. 49(2): p. 208-215.
- [19]. Sangyoon Oh and P.C. Fox, Optimizing Web Service messaging performance in mobile computing Future Generation Computer Systems, May., 2007. 23(4).
- [20]. Satish Narayana Srirama, Matthias Jarke, and W. Prinz, Mobile Web Service Provisioning Proceedings of the Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services, Feb. 2006.
- [21]. Satish Narayana Srirama, P. Matthias Jarke, and P. Prinz, A Mediation Framework for Mobile Web Service Provisioning Proceedings of the 10th IEEE on International Enterprise Distributed Object Computing Conference Workshops, Oct. 2006(0-7695-2743-4): p. 14

About the Author:

Li luqun is a professor in Computer Department of Shanghai Normal University. His main research interests are computer networks, wireless communication and swarm intelligence.

Email: liluqun@gmail.com

Mobile Phone: +86-13671988511