An Architectural Framework for Web Portal in Ubiquitous Pervasive Environment

Muhammad Taqi Raza†, Ryu Jeatek‡, Seung-Wh Yoo‡, Ki-Hyung Kim‡, Seong-Soon Joo†, Wun-Cheol Jeong†
†USN Networking Research Team
†Electronics and Telecommunications Research Institute (ETRI), South Korea
‡Information and Communication Department
‡Ajou University, South Korea
†{taqi, ssjoo, wjeong}@etri.re.kr
‡{ ryujeatek, swyoo, kkim86}@ajou.ac.kr

ABSTRACT

Web Portal functions as a single point of access to information on the World Wide Web (WWW). The web portal always contacts portal's gateway for the information flow that causes network traffic over the Internet. Moreover it provides real time/dynamic access to the stored information, but not access to the real time information. This inherit functionality of web portal limits its role for the resource constrained digital devices in Ubiquitous era (U-era). This paper presents a framework for the web portal in U-era. We have introduced the concept of Portal Locals in the proposed framework, so that the local queries could be solved locally rather than to route them over the Internet. Moreover our framework enables one-to-one device communication for the real time information flow. To provide the in-depth analysis, firstly we have evaluated our approach through mathematical analysis. At the end, we have deployed a test-bed and real time measurements are observed, which proves the efficacy and workability of the proposed framework.

Keywords

Ubiquitous era (U-era), Web Portal, Ubiquitous Sensor Networs (USN)

1. INTRODUCTION

The ubiquitous sensor network is drawing a lot of attention as a method for realizing a ubiquitous society. Through ubiquitous network, a lot of information, not so important individually, is collected in large numbers to establish important information on a wider scope [1]. To make USN a reality in our daily life, research community is paying its attention towards different branches of this next generation network. For instance, the performance of the network is widely examined from viewpoints of robustness, energy efficiency, scalability, etc. Whereas the hardware technologies, MICA motes [2], Smart-its [3], and U^3 [4] are proposed for development of the emerging applications. At the other hand, Sensor Web [5], Sensor ML [6], and 6LoWPAN [7] are the key standardization areas for USN. Moreover, a wide range of USN applications in health care, environmental monitoring, home automation, object localization etc., emerges the role of web portal in U-era. The web portal will function as a single access point to all of the desired USN applications. But the current web portal architecture is not designed for the resource constrained devices. It always communicates with the portal's servers for each user query, resulting delays and generating high Internet traffic. Also the current architecture is unable to provide access to real time information. Hence there is a need for a framework that may broaden the role of the web portal in the daily life.

In this paper we propose a framework for the web portal in U-era. There are several factors which are to be considered while designing the framework. 1) It must provide access to the real time information. 2) User must not be aware of the fact that the application interacts with a specific hardware platform need to be i.e. the proposed framework must provide an abstraction to the lower layers. 3) Every digital device would be capable of communication, resulting into high traffic generation; thus the proposed framework must provide a built in functionality of the network congestion control. We have also observed that there are two types of queries, i.e. specific and general. Specific queries address a specific set of audience, like nearest gas-station, and current temperature in Seoul etc; Whereas the general queries address a wider set of audience, like stock exchange index, flight schedule, and latest news etc.

By keeping in mind all these requirements and facts of U-era, we have proposed a framework for web portal. We have logically divided the framework into different regional locals, where each regional local only addresses a particular region. The main idea behind regional local is that the queries which are meant for the same region, are solved locally rather than to route it through the Internet. We have deployed a test-bed to evaluate it. We have offered different applications on the single platform, and tested the feasibility of these applications. We have also tested the performance of regional locals, in terms of query response time, and total traffic generation. Our evaluation shows that the framework is well suited for all the sensor applications with significantly low traffic overhead.

The rest of the paper is organized as follows. Section 2 describes the related work. The motivation is discussed in section 3. We present web portal framework in section 4, followed by the analytical and experimental performance

978-0-7695-3649-1/09 $25.00 © 2009 IEEE
DOI 10.1109/CNSR.2009.25

evaluation in section 5 and 6 respectively. Section 7 concludes the paper, and describes the future work.

2. RELATED WORK

Researchers have made various contributions by identifying potential research areas in ubiquitous pervasive environment, and paying significance efforts towards it. We can classify these into efforts with similar applications goals (sensor web system and sensor model language) and those that employ significance impact in our framework (service oriented architecture and ubiquitous service discovery).

Sensor Web System: The term "Sensor Web" is sometimes used to refer to sensors connected to the Internet. The purpose of a Sensor Web system is to extract knowledge from the data it collects, and use this information to intelligently react and to adapt according to its surroundings. It links a remote end user's cognizance with the observed environment [8]. A sensor web consists of a number of sensor platforms, also called pods, which can be fixed or mobile. Through pods Sensor Web spreads collected data and processed information throughout its entire network. By definition, a Sensor Web is an autonomous, stand-alone, sensing entity that does not require the presence of the World Wide Web (WWW) to function [9].

SensorML, A Sensor Model Language: In today's Internet, web contents are described through HTML a contraction of Hypertext Markup Language, which is predominant markup language for web pages. Similarly, there is a need to present the U-contents in a standardized way over the Sensor Web. The Open Geospatial Consortium (OGC) describes *Sensor Model Language (SensorML)*, a language for moving sensor data among each other, and onto the Internet. The OGC SensorML specification defines it as a standard model and XML Schema for describing sensors systems and processes; provides information needed for discovery of sensors, location of sensor observations, processing of low-level sensor observations, and listing of taskable properties [10].

Service Oriented Architectures: Web services approach for the design of sensor networks has been proposed in [11]. The purpose was to enable a flexible architecture where sensor networks data can be accessed worldwide. In this approach, sensor nodes are service providers and applications are clients of those services. Sensor nodes publish their services by sending the services' description to sink nodes. The Web Services Description Language (WSDL) is used to describe data and functionalities of sensor nodes. Sink nodes provide the service descriptions of the whole sensor network. They act primarily as service providers to the external environment. Applications submit their service requests to sink nodes which then interact with the appropriate sensor nodes requesting their specialized services in order to meet the user application needs.

Ubiquitous Service Discovery: Ubiquitous service discovery is not similar to the service discovery in WWW, like service discovery through JINI, UPnP, and SLP etc. This is due to the fact that, in ubiquitous environment applications and services are not deployed onto a pre-existing network, but instead the network itself grows out of the applications and services the users want [12]. This approach enables users to view the network in the manner most appropriate to them and their requirements.

3. MOTIVATION

U-age can be referred as a fourth generation age where application requirements, information type and services are different from those that are available in the Internet age. U-applications enable access to the real-time information-of-interest, rather than presenting the static information dynamically, stored in the servers. The most popular U-applications are U-health, U-monitoring, and U-traffic. U-health system refers to a system which provides appropriate advice and suggestions in a comprehensive manner according to the situation. Similarly U-monitoring and U-traffic systems are monitoring their environments and informing the customers about the important events, and comprehensively giving suggestions. These U-applications are constantly relying upon the real time information and involve the monitoring of the environment on a very regular basis. By keeping the requirements in view, we need a framework over the web portal architecture which provides a point of access to the real-time information. Moreover development in the Sensor Web system also urges us to provide a framework that broadens the scope of the web portal architecture by making it accessible over the Sensor Web or the Internet stand alone.

3.1. Why Not the Current Web Portal Architecture?

The reason of not using the web portal architecture alone is that it is designed for the web to provide real time access to the information, but not access to the real time information. Moreover it's strong client-server architecture that may not be suitable for quick flow of information, in U environment. Because it is not possible to store all types of real time information, like temperature etc., within the web servers. As a solution, device-to-device communication over the network provides quicker access to the real time information, with little overhead. The architectural difference between the current web portal and the proposed framework oriented web portal is explained in section 4.1

3.2. The Framework and Its Desirable Features

By keeping in view the requirements in U-age, we have thought about the powerful framework platform that would serve the same purpose as the web portal has in the Internet age. The framework is designed by considering the following key requirements:

- It must address the real time information.
- Framework must be scalable, fault tolerant and robust.
- It must be able to offer a huge *portal space* i.e. variety of U-applications could be offered over a single portal applications.

- It must address all kinds of sensors networks, i.e. homogeneous and heterogeneous.

4. ARCHITECTURAL FRAMEWORK OF WEB PORTAL IN PERVASIVE ENVIRONMENT

In accordance with the sensor requirements and issues, we have proposed web portal framework architecture for pervasive environment, which is hybrid in nature. The framework broadens the role of web portal and enables its importance in the pervasive environment. Figure 1 explains the architectural framework:

Sensor Applications: Portal is a platform where multiple sensor applications could be provided, like U-traffic, U-health, U-management etc. Sensor applications communicate with the sensors through a standard language, like SensorML.

Client: Client can access various sensor applications that are built upon the web portal, like temperature information, traffic information, and to a particular object information etc. Now, we can logically divide the whole region into different regional locals.

__Regional Locals Layer__: This layer serves as the backbone of the whole framework. This layer can be divided into a number of regional local(s).

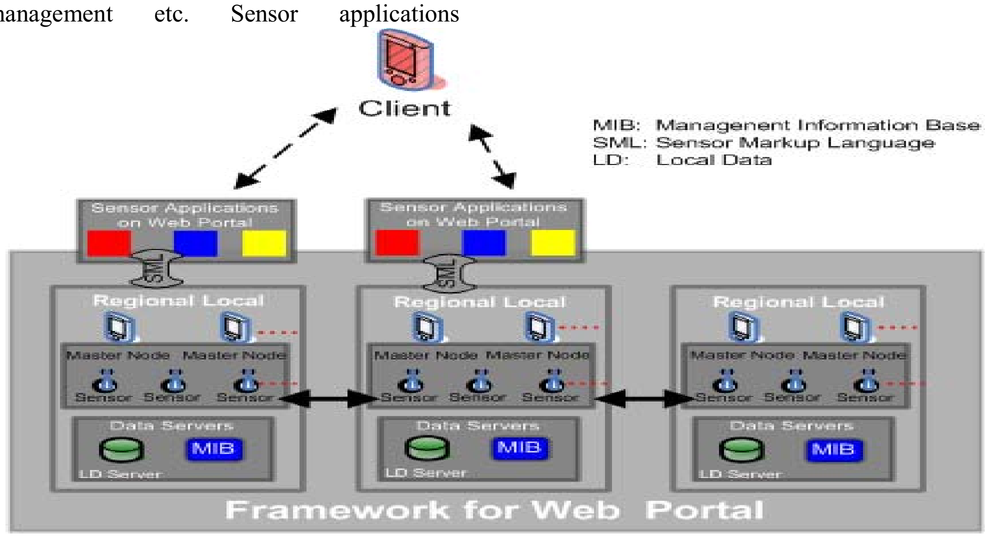

Figure 1: Architectural framework for the web portal in pervasive environment

Regional Local: In U-environment, far and more each device is equipped with the computational and communication ability. For the better information management and for the overall traffic reduction, a number of regional local tiers are introduced. Infect each regional local tier represents a particular region like a city or a town. We believe that the most of the user's queries would be pertaining to the region in which he/she belongs to, at the time of querying. Like customer could be willing to know the current temperature, best possible traffic route, the hottest deal available on a particular product offered by the nearby store, emergency notification to the nearest concerned office etc. We assert that each regional local represents a particular region, where regional queries would be solved within the region rather than contacting the global server/information hub that causes the latency and traffic congestion.

Master Node: A master node is a relatively powerful sensor node compared to the other sensor nodes in the network. It acts like a central authority which manages all the information to and from the sensor nodes that are associated with the master node. The basic motivation of introducing the master node is that the sensors are low power and low processing devices, and would not be able to compute each time queried. Moreover to resolve the query, multiple sensors collaborate with each other and a final result is replied to the user, this all management is not possible with just one or more sensors.

Data Servers: The Data Severs include Local Data Server, and MIB Server. The **Local Data (LD) Server** keeps the information and data pertaining to the region, for which they are deployed for, whereas the **MIB Server** keeps the management related information for the sensors deployed in that region.

__The Global Servers Layer__: These global servers are part of current web portal architecture. Here we want to highlight the role of these servers in pervasive environment.

Data Servers used in the current web portal architecture are also referred as Global Data (GD) Servers that provide the global information. At this point, it is very important to distinguish between LD and GD. LD means the data which is for the interest of local users, like the directions towards the nearest gas-station, or the current temperature in regional local etc. Whereas the GD is for the interest of a larger community and these queries are independent of the object's location. For example, stock exchange information, flight information, and latest news updates are part of GD and stored in GD servers. These applications may be queried by any user, or from any regional local.

Client can access to the various sensor applications. These sensor applications connect with the appropriate regional local, and then the master node, with the help of its associated sensors, tries to resolve that query. If the query is resolved within the regional local then the result is replied back to the client node. If the query could not be resolved within the regional local itself, then the master nodes try to resolve the query by involving Global Servers (of web portal) and/or other regional local (s).

5. ANALYTICAL EVALUATION OF THE PROPOSED FRAMEWORK

In this section, we provide a traffic analysis model for proposed framework of web portal in U-era. Query request that is initiated by the client, is entertained by the most nearest master node from the client. If the immediate master node is busy, then one of the sensor nodes associated with that master node is responsible to route the query packet to the next immediate master node, and so on (as shown in the Figure 2). Moreover, if the channel is busy then the client request cannot be reached and rejected; otherwise, the client is taken into service immediately.

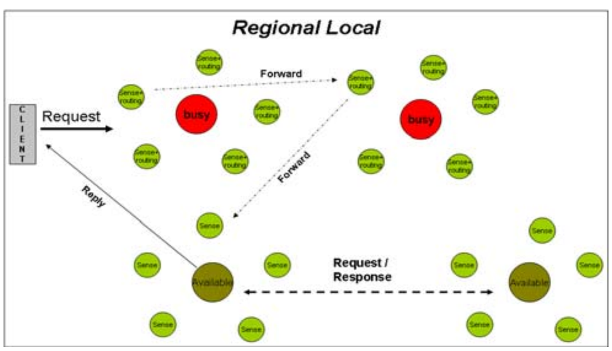

Figure 2: Traffic scenario when different master nodes are busy

We assume that incoming requests form a Poisson process with rate ηreq (in requests/second), each with an exponentially distributed holding time with mean Δconn (in seconds). Also α be the probability of queuing, when the master node is busy, and β be the probability at which the request is denied.

Firstly, the client connection time *Δconn* consists of the time to find the available master node to process the request plus the client waiting time when the channel is busy. Accordingly, the average connection time is given by

$$\Delta\text{conn} = \Delta\text{master} + \omega\text{collision}$$

Secondly, the client response time *Δresp* consists of the service discovery time, plus the target master node waiting *ωtarget* and execution times, plus the I/O waiting and service processing times. Accordingly, the average response time Δresp is given by:

$$\Delta\text{resp} = \Delta\text{service} + (\omega\text{target} + \Delta\text{exec}) + (\omega\text{io} + \Delta\text{buf}).$$

The end-to-end service time is then the sum of connection and response times, Δconn + Δresp. Also, the effective end-to-end request connection rate ηconn is given by:

$$\eta_{conn} = \eta_{req}(1-\beta_{master})(1-\beta_{collision})\frac{(1-\beta_{target})}{(1-\alpha_{rety}\beta_{tot})}$$

Where, βtot = βmaster + βcollision(1 − βmaster) + βtarget(1 − βcollision)(1 − βmaster)

Moreover, the master node interprets the query and decides, with the help of local discovery (LD) server, that whether this query can be resolved within the same regional local or not. The process of entertaining the query request can be shown in the Figure 3.

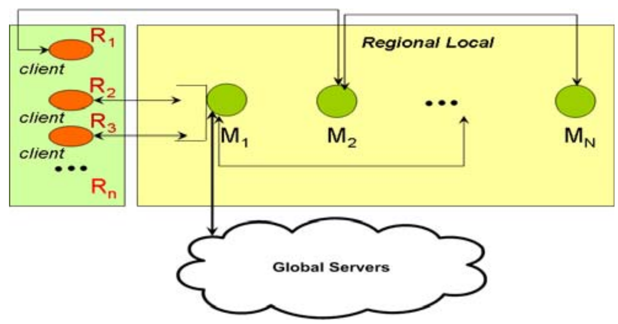

Figure 3: Working of framework oriented web portal with multiple queries at a time

Each mater node also maintains a routing table of the neighboring master nodes. Consider *'D'* be the distance between source and destination, with *'H'* be the number of hops between them, and *'μ'* be the per hop processing. Whereas *'T'* be the link transmissions speed. If *'Z'* be the size of the packet (i.e. 127 bytes), then the total delay from source to destination, for *one* packet when there is no serious queuing delay, is given by:

Delay= Total propagation delay + total transmission delay + total store and forward delay + total processing delay

$$Delay = H\times\left(\frac{D}{\text{Speed of light}}\right)+1\times\left(\frac{Z}{T}\right)+(H-1)\times\left(\frac{Z}{T}\right)+(H-1)\times\mu$$

***Case 1**: We assume that this query can be resolved within the same regional local.*

Packet loss rate can be considered as an important factor that affects the quality of transmission and may cause delays. There could be a number of packets losses within regional local, because the client communicates with the master node over the wireless link.

Let *'p'* be the data packet loss rate and *'q'* be the loss rate for acknowledgments. Then the number of required data packet transmissions, for *one* packet, is given by:

$$\lim_{n\to\infty}((1+((1-p)q+p)+((1-p)q+p)^2+((1-p)q+p)^3 +...((1-p)q+p)^n)) = \frac{1}{1-((1-p)q+p)}$$

Whereas $(1-p)q+p$ is the probability with which the packet is required to be retransmitted. Because either the packet is lost with probability *'p'*, or the acknowledgement of the data packet is lost with the probability of $(1-p)q$.

Hence the number of required acknowledgment transmissions is then given by:

$$\lim_{n\to\infty}((1-p)+(1-p)y+(1-p)y^2+(1-p)y^3+... (1-p)y^n) = \frac{1}{1-x}$$

$$\text{where } x=((1-p)q+p), y=(q+p)$$

Though TCP is used for transmission of data packet from an application to the regional local network, but its behavior is almost smooth within the regional local. Thus we have discussed TCP behavior (in *Case 2*) when data packet is routed over the Internet.

Case 2*: We assume that this query can't be resolved within the same regional local.*

If the master node doesn't discover any information about the query, or the query belongs to some other regional local; then the query is routed over the Internet for its discovery and/or execution. In this case, in addition to packet loss rate as discussed in *Case 1*, the Internet delays would be added as well.

Let *'B'* be the queue length and *'C'* be the queue capacity (packets/sec) of the Internet router, *'F'* be the probability of packet loss, and *'R'* be the round trip time. If *'W'* be the TCP window size, and *'Q'* be the number of TCP sessions. Then TCP behavior, as described in [14], can be modeled as:

$$\frac{dW(t)}{dt} = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t)}B(t-R(t))$$

$$\frac{dF(t)}{dt} = \frac{Q(t)}{R(t)}W(t) - C$$

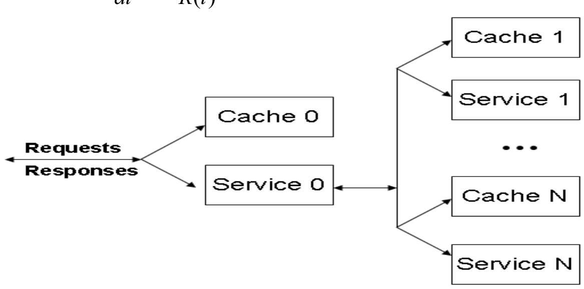

Figure 4: Analytical model for multi-tiered web applications

We have also followed the web portal's multi-tier architectural design for query processing at Global Servers. For web portal application, a single request is served by hundreds of applications operating in parallel [15]. Such elementary software applications, designed to be composed with each other, are commonly referred to as *services*. The Global Servers tier of web portal is logically modeled into the business logic tier, and the servers tier. For fast query response, the business logic code generates multiple queries to its underlying databases (servers tier) to generate a single response. There are various analysis models proposed for multi-tier web architecture, but the model presented in [16] best describes the approach. As described in [16], the query requests arrive according to a Poisson process with rate λ to a dedicated entry level 0, (where all requests arrive). If the query response is cached at caching tier 0 with probability $P0$, then the response will be made to the client, without any further processing. However, the business logic tier will execute the query, if the query needs to be processed, with probability $1-P0$, by service tier 0 and the other N levels in the queuing network, as depicted in Figure 4. The request is routed to each of the N levels in sequence, where at each level i, the request is served by caching tier i with probability P_i. If the request is not cached at the caching tier, with probability $1 - P_i$, the request is served by service tier i. When the query is solved at level i, the results are sent back for processing to service tier 0. For fast query processing, a request that is sent from the business logic to each level i, generates K_i requests back and forth, where K_i is a non-negative discrete random variable. Thus, a request has fully completed its service at level i after K_i service completions. Hence, every request that is served by service tier 0, passes this tier $1 + K_1 + \cdots + K_N$ times, and finally leaves the system after having visited all N levels.

Each caching tier i is modeled by an infinite-server queue having general service times with an average of $\gamma_{c,i}$ time units for $i = 0, \ldots, N$. Whereas, each service tier i is modeled by a processing sharing queue with an average service time of $\gamma_{s,i}$ time units for $i = 0, \ldots, N$. The mean response time of the service can be obtained through the sojourn time of the request in the system. Let $S_i^{(k)}$ be the sojourn time of the k-th visit to level i, and $M = EK_1 + \cdots + EK_N$. Then, the expected sojourn time ES of an arbitrary request arriving to the system is given by:

$$ES = E\left[\sum_{k=1}^{M+1} S_0^{(k)} + \sum_{i=1}^{N}\sum_{j=1}^{K_i} S_i^{(j)}\right]$$

Now, we say that $L_{c,i}$ and $L_{s,i}$ denote the stationary number of requests at caching tier i, and service tier i for $i = 1, \ldots, N$, respectively, we have:

$$X(L_{c,i}=l_{c,i}, L_{s,i}=l_{s,i}; i=0,...,N) = \prod_{i=0}^{N} X(L_{c,i}=l_{c,i}) \prod_{i=0}^{N} X(L_{s,i}=l_{s,i})$$

with $l_{c,i}$ and $l_{s,i} = 0, 1, \ldots$ for $i = 0, \ldots, N$. we can also determine the expected sojourn time at the entry node and the service nodes. Let $\rho_{c,0} = \lambda_{P0}\gamma_{c,0}$ defines the load on the entry node's caching tier, and $\rho_{s,0} = (M + 1)(1 - P_0)\lambda\gamma_{s,0}$ for the service tier. Similarly, the load for the basic services is given by $\rho_{c,i} = (1 - P_0)P_i\lambda\gamma_{c,i}$ and $\rho_{s,i} = (1 - P_0)(1 - P_i)\lambda\gamma_{s,i}$ for $i = 1, \ldots, N$. Then, by Little's Law, the expected sojourn time $ES_{c,i}$ at caching tier i is given by $ES_{c,i} = \gamma_{c,i}$, whereas the expected sojourn time $ES_{s,i}$ at service tier i is given by $ES_{s,i} = \gamma_{s,i}/(1 - \rho_{s,i})$ for $i = 0, \ldots, N$. Combining all the expressions for the expected sojourn time at each tier in the network, we derive that the expected response time is given by:

$$ES = E\left[\sum_{k=1}^{M+1} S_0^{(k)} + \sum_{i=1}^{N} \sum_{j=1}^{K_i} S_i^{(j)}\right] = P_0 ES_{c,0} +$$
$$(1 - P_0)\left[ES_{s,0} + \sum_{i=1}^{N} EK_i \left\{P_i ES_{c,i} + (1 - P_i) ES_{s,i}\right\}\right]$$
$$= P_0 \gamma_{c,0} + (1 - P_0)\frac{(M + 1)\gamma_{s,0}}{1 - \rho_{s,0}} +$$
$$(1 - P_0)\sum_{i=0}^{N} EK_i \left[P_i \gamma_{c,i} + (1 - P_i)\frac{\gamma_{s,i}}{1 - \rho_{s,i}}\right]$$

6. EXPERIMENTAL EVALUATION OF THE PROPOSED FRAMEWORK

We have evaluated the proposed architecture on two important aspects, i) total network delay with and without regional local, and ii) the behavior of the network, with and without framework oriented web portal, during a heavy traffic.

6. 1. Network Model of the Test-bed

We examine the proposed framework architecture by deploying a test-bed of 150 sensor nodes, in 1st, 3rd and 5th floor of eight-story building of our University. We have used temperature sensors, heat sensors, humidity sensors, and sound sensors, for framework evaluation. Each sensor node contains Chipcon's CC2420DB IEEE 802.15.4 compliant wireless mote. Each mote is assigned with an IPv6 address, and the total 50 nodes are installed in the rooms and corridors of each floor, with an average distance of 3 meters. Each Sensor mote is part of a PAN (Personal Area Networks), and there are total 5 PANs at each floor. There is one PAN coordinator within each PAN. The PAN coordinator is a powerful sensor node that is main powered, and acts as a master node. The following Figure shows the network topology of the test-bed deployed.

Figure 5: Network topology of the test-bed deployed

We have considered each floor of the building, as a regional local. Each regional local is connected with the sensor gateway, which regulates the traffic to-and-from the regional locals. To evaluate the web portal's performance without regional local, we have connected the web portal application with the external router. Now the query can either be processed through regional local or without regional local.

6. 2. Experimental Results

To evaluate the efficacy of the proposed framework, we have performed more than 1000 experiments to calculate the total end-to-end query response time, and the total network traffic overhead, with and without regional local. The Figure (6 and 7) shows the average end-to-end query response time, i.e. query processing + RTT. Note that the sensors in regional local can handle maximum traffic of 256 kbps. To route the packets over the Internet, we have used loose source routing options in the IPv6 header format. Moreover, we have seen that the round trip time varies region to region. In Figure 6, the average delay of 2ms is observed when the query is resolved within regional local or within the same country (where the query is made), i.e. Korea. The delay within regional local is very smooth (almost constant), but different delays values are observed as shown in the Figure 6 and 7. This is due to the reason that the query can be cached within the master node, which cause the minimum delay; whereas it usually requires two master nodes for processing the query, causing the delay of almost 136 ms. If the immediate master node, from the client, is busy then more than two master nodes process the query, that adds more delays. At the other hand, as shown in Figure 7, a huge delay can be seen from 150 ms to 450 ms, when the query is routed via USA or Europe. It can be observed from the

Figure 6 and 7 that the query response time can be reduced by incorporating regional local. This is due to the fact that the query is handled within the regional local, whereas routing the packet over the Internet adds processing, and queuing delay at each router of the Internet. If the service discovery time is added, then the query response time would dramatically increased in case of general Internet; whereas in regional local it would not affect a lot. From Figure 7, we can observe the difference between the estimated, and the measured end-to-end delay. The estimated delay is calculated through the equations discussed in the analytical model, whereas measured delay is the actual delay observed through the experimental results.

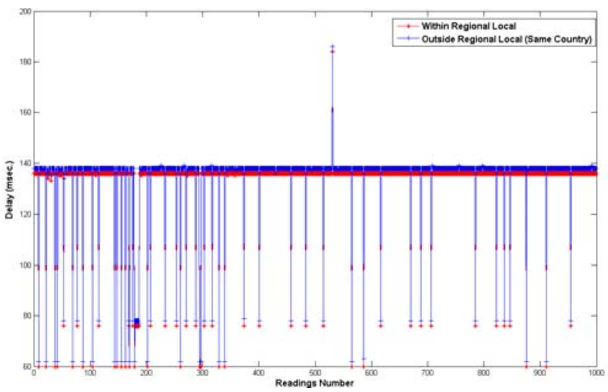

Figure 6: End-to-end delay when query is resolved within regional local or over the Internet (same region)

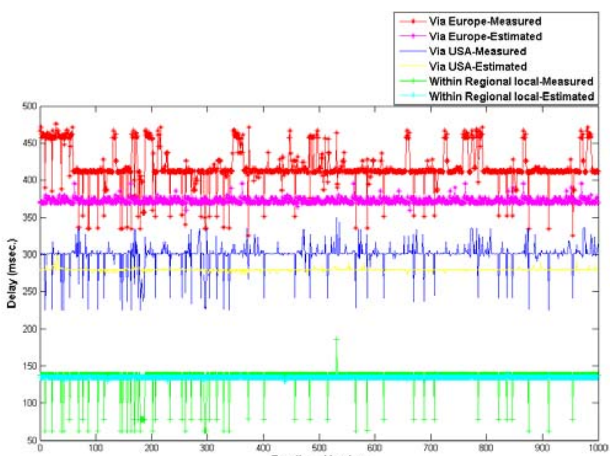

Figure 7: End-to-end delay when query is resolved within regional local or over the Internet (via Europe and via USA)

There are several factors, like link failure, change in routing path etc., which cause the network delay; but the network load is usually the main reason that causes fluctuation in end-to-end delay and, therefore, can be thought as one of the most important reasons among all the factors that cause fluctuation in delay [17]. Figure 8 shows the cumulative distribution function (CDF) of end-to-end delay of Figure 7. We can observe from the Figure 8, that different regions have different CDF values at the given point. We can observe the CDF graph of "within regional local" is almost smooth, showing the balanced network load. Delays can indicate the network load situation very well based on the analysis of the CDF of end-to-end delay in a given path.

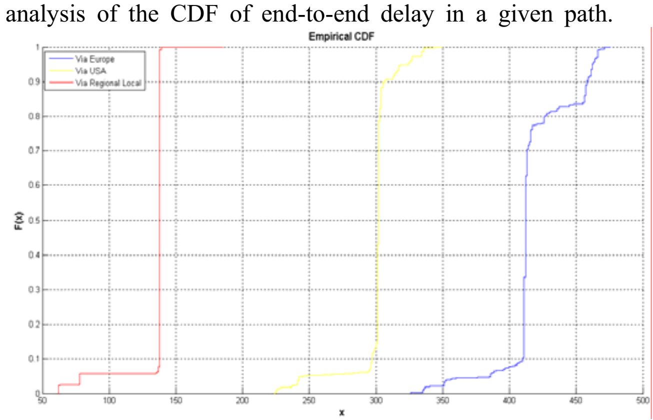

Figure 8: CDF of End-to-end delay analysis

In our experiments, we have also created a scenario of U-era by generating a huge number of packets and injected them, from each floor, into the network with an interval of < 1 ms. There are two types of packets that has been injected into the network; 1st that would be handled by the Internet router (as always in case of web portal), and 2nd that would be handled within regional local or by the regional local gateway. As mentioned before, we assert that in U-era the most of the applications would generate the traffic that is within the local scope of the application, such applications could be home automation, object tracking, shopping mall applications etc. To observe the network behavior in our assertion, we have intentionally generated 40% or fewer packets, for each floor (regional local), where packets' destination address is within the same regional local. These packets will not be forwarded to the gateway, and would be handled locally. It can be seen from the Figure 9 that the Internet router has started dropping the packets, whereas the regional local gateway is still able to handle the traffic. This behavior is resulted due to the fact that the query packets which are destined to the same regional local are handled locally; and are not forwarded to the regional local gateway, resulting less congestion on regional local gateway as compared to the Internet router. Moreover Internet and sensor networks both work at different network bandwidth speed. In our scenario, sensors network bandwidth is 256kbps, whereas the Internet bandwidth is upto 1Gbps. There is a gateway that regulates the traffic between two different networks, known as PAN gateway. If the PAN gateway receives the packet with the speed more than that it can handle, then it also start dropping the packets as well.

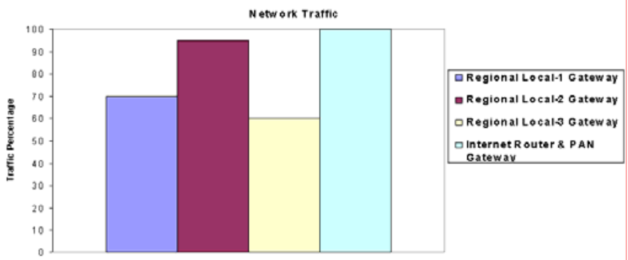

Figure 9: Heavy traffic at network gateways/router

7. CONCLUSION

U-age enables a communication environment in which any informational service is accessible to anyone through any digital deice and network without any limitations of space and time. A variety of services would be offered in U-era that creates the requirement to make changes in the current web portal architecture. We have described in details that the current web portal architecture is not feasible in U-era due to the nature of the query, which is real time, and also that may not require interaction with the database as well. It can also be observed that in U-era the most of the queries would belong to the same region where the queries are made. In this paper, we have proposed the architectural framework for the web portal in pervasive environment, and introduced the concept of regional local. We have shown that through regional local not only the query response time is improved but also the network overhead is reduced as well. We have evaluated our framework through the test-bed. We have observed that the web portal's implementation with our proposed framework achieves the desired results and enhances the web portals requirement in U-era

7. 1. FUTURE WORK

As for our future work, we would verify the importance of MIB server, after deploying a large number of sensor nodes, and would discuss the management of sensor nodes. As the aim of this paper is to introduce web portal's importance in U-era, its concept and the architecture, i.e. framework for web portal; So our next step is to discuss the results from management of large number of sensors point of view.

ACKNOWLEDGEMENT

This work is partially supported by the IT R&D program of MIC/IITA, [2007-S-027-01, Development of IP based sensor network (IP-USN) technology].
Also the partial work is supported by the Ministry of Knowledge Economy (MKE) of the Republic of Korea. [2008-F-052, Scalable/Mobile/Reliable Wireless Sensor Network Technology]

References

[1] Shigeru Fukunaga, et al., "Development of Ubiquitous Sensor Network", Oki Technical Review, October 2004/Issue 200 Vol.71 No.4
[2] Crossbow Mica motes: http://www.xbow.com/
[3] Smart-its: http://www.smart-its.org/
[4] Yoshihiro Kawahara, et al., "Design and Implementation of a Sensor Network Node for Ubiquitous Computing Environment," In Proceedings of IEEE VTC2003-Fall, Orland, USA, October 2003
[5] Yan Ke, Suman Nath, et al., "IrisNet: An Architecture for a Worldwide Sensor Web", IEEE pervasive computing magazine OCTOBER–DECEMBER 2003
[6] A Sensor Model Language: http://www.sensorsmag.com/articles/0403/30/main.shtml
[7] David Culler, "Secure, low-power, IP-based connectivity with IEEE 802.15.4 wireless networks", Industrial Embedded Systems 2007
[8] Kevin A. Delin, "The Sensor Web: A Distributed, Wireless Monitoring System", SensorsMag, Apr 1, 2004
[9] "Sensor Web", http://en.wikipedia.org/wiki/Sensor_Web
[10] Mike Botts, "A Sensor Model Language: Moving Sensor Data onto the Internet", SensorsMag, 2003
[11] Flávia Coimbra Delicato, et al., "A service approach for architecting application independent wireless sensor networks", *Cluster Comput.* 8 (2–3) (2005)
[12] Siva Sivavakeesar, et al., "Service Discovery Strategies in Ubiquitous Communication Environments" IEEE Comm. Magazine 06
[13] Web sites handle the load of millions of visitors http://computer.howstuffworks.com/question342.htm
[14] V. Misra,W. B. Gong, and D. Towsley, " Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED," *ACM* SIGCOMM, 2000.
[15] Vogels, W., "Learning from the Amazon technology platform." ACM Queue 4(4) (2006)
[16] Sandjai Bhulai, et al., "Analysis of End to End Response time of Multi Tier Internet Service", Technical Report WS2006-11, Vrije Universiteit Amsterdam, 2006.
[17] Wei Zhang and Jingsha He, "Modeling End-to-End Delay Using Pareto Distribution", ICGD&BC 2007