Computers and Electrical Engineering 62 (2017) 44–52

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A gradient sensing middleware to handle flash flood[☆]

Nova Ahmed[*], Shuvashis Ghosh, Rifat Ahmed Hassan, Sian Iftekher Galib, A.K. Azad[1], Minhaz Ahmed Syrus[1]

Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh

ARTICLE INFO

Article history:
Received 28 September 2016
Revised 10 August 2017
Accepted 10 August 2017
Available online 18 August 2017

Keywords:
Flash flood
Middleware
Internet of thing
Solution for natural disaster in Bangladesh
Low cost distributed System
Flash flood prevention

ABSTRACT

Flash Flood is a natural disaster that floods away large area in a sudden manner. Bangladesh is affected by this natural disaster and loses valuable assets regularly. We currently have developed a system that has cloud infrastructure for intelligent long term data processing and low cost sensing devices that are able to make local decisions in its surrounding area. The system is adopted to work in a developing country such as low cost sensing that is locally available along with ability to adapt to sparse network infrastructure. Our system is called Shonabondhu: Golden Friend in Bengali, and we share our design, development and evaluation cycle of this system.

1. Introduction

Our advancement in technology is not reflected when we are going back to a third world country fighting natural disasters often. We have considered Bangladesh as our field of study and we consider the problem of Flash Flood. Flash Flood occurs when the river water level rises rapidly within a short period of time (often within 2 hours water level can rise as high as few meters) mainly caused by sudden heavy rainfall in locality. It is a major problem in many different countries like Bangladesh, India, China, Nepal, Malaysia, Philippines and many other countries [1,2,3]. It must be mentioned that the low area that are flooded by the flash flood are used for fishing through the monsoon season and living with this periodical presence of water is part of the lives of people residing in our low land area (Haor Area, called in Bangladesh). People are generally aware of the water lever rise in monsoon season—the problem lies in the sudden nature of water level increase. Residents of Haor area have mentioned that they would be greatly benefited if they could have an earlier warning. The current water level measurement systems are manual where dedicated personnel go out to measure the water level which has its shortcomings of missing the critical time to warn people with enough time to prepare.

There have been an array of research studies that consider water level sensing, monitoring and /or prediction and offer valid system support for deployment of such system. Elizabeth et al. [4] have worked on water monitoring system deployed at Honduras, a river in Massachusetts using a predictive sensor environment; Stoianov et al. [5] used sensor network to monitor Boston Sewer Commission Data, Sunkpho and Ootamakorn. [6] used real time flood monitoring system deployed at Thammarat a Southern province of Thailand; Hughes et al. [1] used sensor based system to study flood monitoring system

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. Tie Qiu.

* Corresponding author.
E-mail address: nova.ahmed@northsouth.edu (N. Ahmed).
[1] Independent Researcher, Dhaka, Bangladesh

http://dx.doi.org/10.1016/j.compeleceng.2017.08.006
0045-7906/

on the River Ribble in North West England (Details of our discussion will be found in related work section). The major difference lies in the problem scopes. In the cases of previously conducted studies—a single river and river bank is studied. When the geographic location is closely situated—a light weight solution like wireless sensor network based architecture can be an excellent solution as we have seen in many existing systems. In Bangladesh, however, we have a network of rivers that contribute to the flash flood problem. Our water monitoring system must include various locations that are geographically distant. We need a solution that is able to study the water level, consider history data and is very responsive so that real time alerts can be generated that is coherent with the river water flow system. We also have to focus on a solution that is low cost i.e., that exploits existing network infrastructure along with low cost, locally available sensing devices.

We have been working on the Flash Flood problem from 2012 [3,7–9] and have modified the system based on environmental challenges, available resources and low cost considerations which we share in our research work. Our system Shonabondhu (Golden friend, in Bengali) is a middleware system that uses cloud infrastructure, long term learning decisions at server level and uses Internet of Things (IoT) devices enabled with sensors that have path abstraction to communicate among sensor modules and decision making capability to locally disseminate warning messages. We have used the network available for mobile phone communication which is rich in terms of coverage in our country. We have designed the system, developed it and deployed it for experimental purposes in few sites of Bangladesh. We are in the process of actual deployment of sensing system with the collaboration of the Water Development Board of our country.

Shonabondhu consists of servers which are assigned a gradient information as a function of its current state in terms of local water level, rainfall along with historical information for that region regarding that particular time of year. These servers are called gradient servers and these servers collect information from the sensor nodes and process them. If for any environmental factor, the gradient information changes in a server—it will propagate that information to interested servers which may be affected by the change of that particular server. And the gradient information may be changed in cascading fashion accordingly. It can be the case where a gradient server near the source of the river experiences sudden rise of water level and as electrons travel faster than any physical entity, we can expect that server to propagate that water level rise information to other servers along the connected river system and if certain thresholds are crossed, adequate alarms can be generated. We have also considered a webserver that will act as the spokesperson for the entire system and will be responsible for doing long term data analysis unlike the local servers. In our infrastructure, we have considered the central server to be part of a cloud for resilient and reliable performance. At the lower level we have IoT devices with processing capability and communication capability to make locally intelligent decisions. For example, a sensor node will bundle up water level information over few reading when the water level is stable and far below from the danger level. Once the water level starts to rise, it will increase the frequency of message dissemination along with communication to the local authority. We have published several parts of our incremental work previously [3,7–9] and here we have the IoT sensing system presented which has not been published previously. It must be noted that the paper does not work on the hydrological aspects of flood monitoring system—it uses terminologies provided by the water development board, Bangladesh such as danger level and water level and works accordingly. Specifically, our contribution lies in the following areas:

- Design a distributed sensor system using gradient information learned from previous sensor data
- Extensive study on individual sensors and modularized test in laboratory as well as in outdoor setup
- Design and development of IoT Sensing system that can make local decisions
- Study of simulation environmental setup using Cloud in case of heavy or light load of sensor generated data

The rest of the paper is organized as follows: we discuss the related work in Section 2, followed by background in Section 3. We discuss about our proposed system in Section 4, evaluation in Section 5 and finally, the conclusion and future work is presented in Section 6 in consecutive order.

2. Related work

We have looked at works that are similar in concept with our research in many different ways: existing work on water level sensing, disaster sensing mostly uses wireless sensor network as infrastructure and we have consulted existing state of the art work, there are some work from different research studies that use the concept of gradient sensing works but in different context. At the same time, our path based communication has been inspired by some previous work which is illustrated here in the following discussion.

Elizabeth et al. [4] have worked on a deployment that covers Honduras, a river in Massachusetts using a predictive sensor environment. It uses a two tier approach to minimize cost—some nodes work in long range using radio signal while other nodes work for shorter range. It uses three sensor nodes for testing purpose to cover the particular river of interest. Sunkpho and Ootamakorn [6] discuss about a real time flood monitoring system. It is installed to monitor Nakhon Si Thammarat, a southern province of Thailand where flooding is a recurrent event. It uses sensing monitoring in 15 sites using internet based real time monitoring. The middleware named VitualCom is in charge of communication among sensors and Application Server. Stoianov et al. [5] used sensor networks to monitor water supply and sewer system. This system uses wireless sensor network to cover the area of interest. It is used at Boston Water Sewer Commission and it collects water level information in real time. The remote monitoring system uses high data rate sampling which is 1000 samples per second. The entire system is based on Wireless Sensor Network (WSN). LooCI System [1] developed by Hughes et al. focuses on a novel component binding model using even binding a wireless sensing system. It can be a good candidate to implement

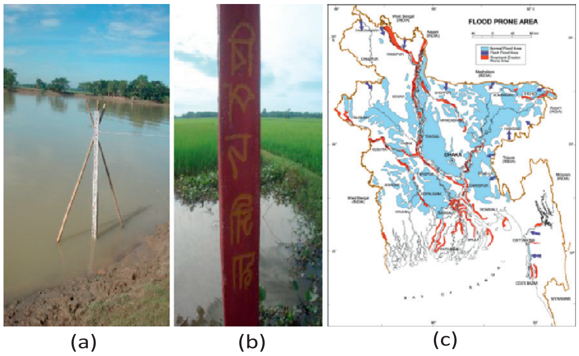

Fig. 1. Manual flood water measurement (a) Lined stick (b) Cement installation with danger note on it (c) Flash flood prone area in Bangladesh (Shown in Red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

real time sensing system. Hughes et al. [10] discusses Gridit middleware used for flood monitoring and warning scenario. It supports a WSN based flood monitoring on the River Ribble in North West England using GridStix hardware platform. It uses an overlay based framework in Gridit. Rekhis et al. [11] use a combination of WSN and RFID readers to collect water quality information. It does not discuss about any middleware explicitly but it works based on data collected from underlying sensor system and analyzes them in real time. The systems of consideration for water level monitoring provide great insight towards implementing a real time monitoring system using sensors. However, wireless sensor network may not be suitable for a country wide development. Wireless sensor network modules can be locally deployed but at the same time there must be a coordination among the local components. Shonabondhu also focuses on a server to server trigger based communication which actually mimics the flow of river and spreads out data in a similar fashion. This is where our system stands out from the rest of the system. Chen et al. [12] discus challenges in using WSN for monitoring natural disasters. WSN having quick response, low cost and being scalable suffer when there is a presence of large amount of data or requirement for low level processing at lower level. Seeger et al. [13] similarly approaches challenges deploying WSN in the wild and middleware centric issues. These insights provided us with valuable insights selecting a platform for a robust system. Park et al. [14] worked on water quality monitoring for hypothetical river using simulation based study. It should be able to provide a base line to compare our system theoretically.

The work developed by Ye et al. [15] consider gradient broadcast as a method to deliver messages for large scale sensor network rather than consider the sensors to have gradient information. There has been gradient based routing [16] by Faruque et al., where the gradient information is used for routing which is similar to our concept while we use the gradient information more for data collection importance rather than involving that information for routing. Our concept is similar to the concept proposed by Lin et al. [17] where the gradient information is used among the sensors for sensing capabilities.

The concept of path is used in many different contexts including fault tolerance [18], compiler optimization techniques [19]. Our work is inspired by the use of paths in these various contexts. We have used the concept of path used to improve system reliability from RF^2ID [20]. Cloud computing is the most recent abstraction based distributed system that offers computing services as pay-as-you-go basis [21] which has been considered in our system for long term data evaluation purposes.

3. Background of flash flood and motivation

Flash flood refers to a phenomenon where water level raises rapidly within a short period of time. It is caused by sudden incident of heavy rainfall. Flash flood is a major problem in many countries such as India, China, Nepal, Malaysia, Philippines and many other countries [1,2,3]. We have focused our attention to flash flood problem in Bangladesh [22]. There are many places that are prone to this problem as can be seen in Fig. 1(c). Flash flood is considered a major disaster in the hilly region of Bangladesh. The north eastern part of Bangladesh has forceful rivers in the hilly areas consisting of Shunamganj, Sylhet and Netrokona districts. Heavy rainfall in India, where most of rivers are originated from, causes flash flood in short notice of time. There has been major flash flood incidents recorded in 1954, 1961, 1966, 1970, 1987, 1988, 1993, 1999, 2004, 2006, 2007 and 2010. The current water monitoring system is primitive and manual where the water level is periodically checked by people against an installed water level measurement stick as can be seen in Fig. 1(a) and (b).The person in charge makes a phone call to the central office which is logged by an office on a three hour interval. Often water level rises are not recorded as it rises up during the night when water level is not measured. Recently there have been some installations to the measuring stick in the water along with few sensor installations on strategic points and important rivers (e.g., recently installed bridges like Bangabandhu Bridge). Many of these deployed sensors are extremely expensive which is not scalable considering the developing country context of Bangladesh.

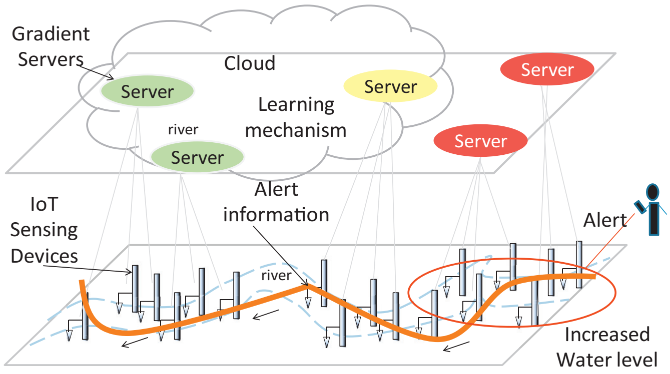

Fig. 2. System architecture.

4. Proposed System: Shonabondhu

Shonabondhu System works in three different planes as can be seen in the schematic presented in Fig. 2—the first plane consist of Server/Servers residing in Cloud and is responsible for long term data forecasting which impacts strategic planning and making long term decisions regarding flood and flood affected areas; the second plane consists of the distributed local servers known as Gradient Servers which use real time decision making and forecasting at local points based on its gradient information calculated by local water level, rainfall level, time and expected water level; the third plane is responsible for real time monitoring and consists of water level sensors deployed at various localities.

4.1. Query server

The query server is responsible for handling the user level queries during monsoon season. It is responsible for aggregated data collection and long term data analysis so that corresponding strategic decisions can be made during off season. The amount of data to handle may vary based on the time of the year and local situation of flood. Our analysis for processing a local emergency situation has shown 2000 hits on a server in a minute. In such a dynamic environment, a cloud server can be a great option which can add hosts on demand.

4.2. Gradient server

Gradient servers are distributed multithreaded servers placed at various physical locations. The servers communicate among themselves using their gradient information which is a function of its current water level, rainfall, time and expected water level. The servers are responsible for making local forecasting of data and work proactively to alert other servers along the physical vicinity so that advanced alert messages can be generated. It uses three levels of communication—a stable and secured communication mechanism to send summary data to cloud server periodically, a stable and gradient based communication system (described in the following text) to forecast flood locally and a wireless communication mechanism to collect, filter and process continuously generated sensor data.

Gradient Level: Every server has a gradient level which is a function of its current water level, current rainfall information, time and expected water level. The gradient information of a specific server is related to the neighboring servers as the river water is continuously flowing but the depth and intensity may vary from a particular location to the other one.

Inter Server Communication: Inter server communication can take place in two different ways. The servers that are at the same gradient level and are located in closer geographic locations use this mechanism known as Neighbor Information Exchange, share their current gradient information periodically. This period gradient information plays a role in improving the overall system accuracy in forecasting. If there is a sudden rise in a server to gradient level L, then other servers are expected to have similar changes and if that is not the case, care can be taken to check whether there is any error in the readings or there is a sudden anomaly in the water level system.

Sensor to Server Communication: Sensor to server communication takes place using SMS messaging system. The detail of our implementation is presented in the following section.

4.3. Communication to central server

The cloud server is sent summary data on hourly basis. There is also a trigger based communication to the cloud server in case of water level rise or any other kind of anomalies (extreme rain level) that deviates from the acceptable input value.

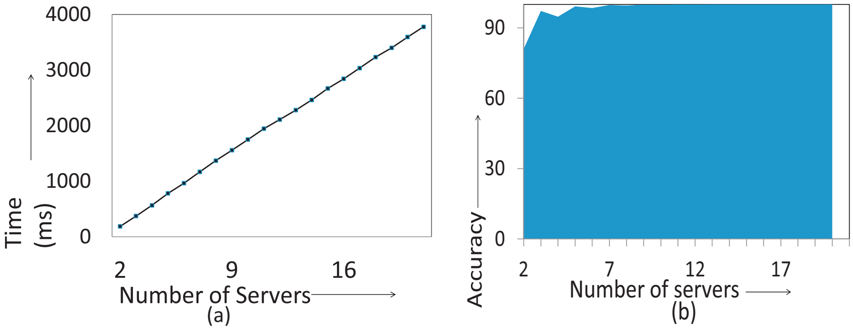

Fig. 3. (a) Scalability study (b) Accuracy study.

4.4. IoT sensor system

We have modified the basic sensor system to an IoT system which is able to make local decisions in case of disaster scenario. The IoT sensing system communicates to other sensor systems, servers and local users using SMS messages. The IoT system makes opportunistic decisions to accumulate messages—the system sends collection of messages when the water level is stable, on the other hand, when the water level starts deviating from previous readings over a window of time, it starts sending frequent messages.

We discuss the system implementation along with the evaluation in the next section.

5. Evaluation

Our system implementation is based on a proof of concept system deployed in a smaller region near a river bank which has been emulated in a larger distributed system setup. We discuss the implementation along with the system's performance evaluation in this section.

5.1. Gradient server

We have considered the Meghna basin in our current experiments. The Meghna system originates in the hills of Shilong and Meghalaya of India. Other rivers like Barak, Amalshid, Surma and Kusiara rivers join in the journey in this river system which are mostly highly flashy and steep rivers originated where the highest average waterfall is recorded (Cherapunji in Asam, 10,000 mm annual rainfall). There are 11 existing water level monitoring stations in this area. We have used the rainfall data and river water level data over the monsoon period (May to September).

Communication among Gradient Servers: Communication is conducted by API support in our middleware system. Nodes that are along the flow periodically exchange current water level information of the Gradient nodes and adjust their gradient level accordingly.

We have considered the study of Flow based communication using a local setup using a cluster of computers in the laboratory having total 32 numbers of HP Compaq dx 7510, each with Inter Core 2 Quad Processor, Clock speed of 2.66 GHz, 3 GB RAM in a Linux cluster setup. The flow based communication overhead plays a role in disseminating real time information in the presence of sudden water level rise.

Our server communication across the Gradient nodes is conducted by sensing emulated sensor data packet transformation. As can be seen in Fig. 3(a), the system is scalable across the increase of gradient nodes, showing linear rise of communication overhead. In a country like Bangladesh, the actual data transfer may differ from this scenario as there are unexpected factors like network delay, power loss and variability of speed across various physical locations. The increased number of servers also plays a positive role in improving the sensor level accuracy. We have considered accuracy level of 90% and one sensor per server (to study the worst case behavior of system) using emulated sensors. Our study shows perfect accuracy level (100%) when we have around 5 servers along the path as evident in Fig. 3(b).

We have investigated the network delay across our country. We have considered the farthest flash flood point of Sylhet from Dhaka, where the flash flood server is connected. We have sent 200 data packets where 1914 packets were received in time showing 4% data loss; approximate travel time of data packets was on average 178 ms having a minimum 159 ms and maximum of 1010 ms—the histogram of data receive time is shown in Fig. 4.

The country wide network consists of a combination of optical fiber network and microwave connectivity. IoT Sensing Module.

The Gradient server wirelessly communicates to the ultrasonic sensor, optical flow sensor or emulated sensor. The ultrasonic sensor and liquid level sensors are attached to the arduino board [23] for its flexible programming interface. We have considered Ultrasonic Sensor (HC SR04 [24]) and Liquid level sensor (LL103101from Honeywell [25]) for our initial stud but have concentrated on the first one for detailed evaluation and implementation. The reason behind the choice was based on

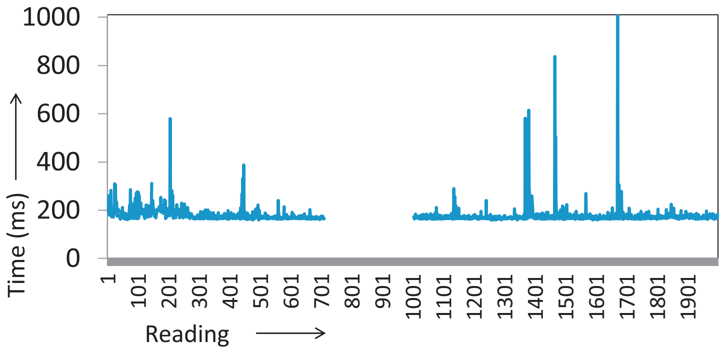

Fig. 4. Data transfer delay across Dhaka to Sylhet.

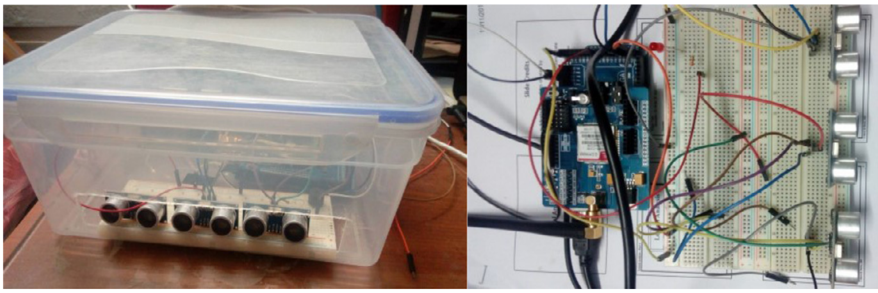

Fig. 5. Recent model of our project.

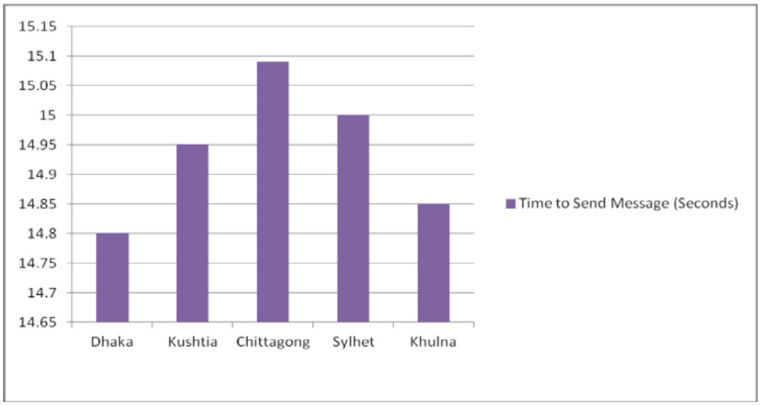

Fig. 6. Time for sending SMS to different parts of the country.

the inputs from Bangladesh Water Development Board as they suggested sensors that are noncontact based as often sensor modules can be affected by the great force of flood water.

IoT Integrated System: The main system will contain an arduino mega, couple of ultrasonic sensor, sim module for sending the alarming message, server and power source. The system will measure the water level and there might be a threshold level of water and when this threshold level exists then the alarming message will send to the people of that area and also to the server of the flood alarming authority and also there must be alarming system which will notify through siren. The system is shown in Fig. 5.

Software Module: We have also developed a web site that always show the sensor boards data from different parts of the country by updating every value and also show the warning that the flash flood can be happen if the water level reaches at the danger.

Communication Testing: We have considered the overhead of SMS message based communication where the source sensor was located in Dhaka and destinations where located in various destinations throughout the country as can be seen

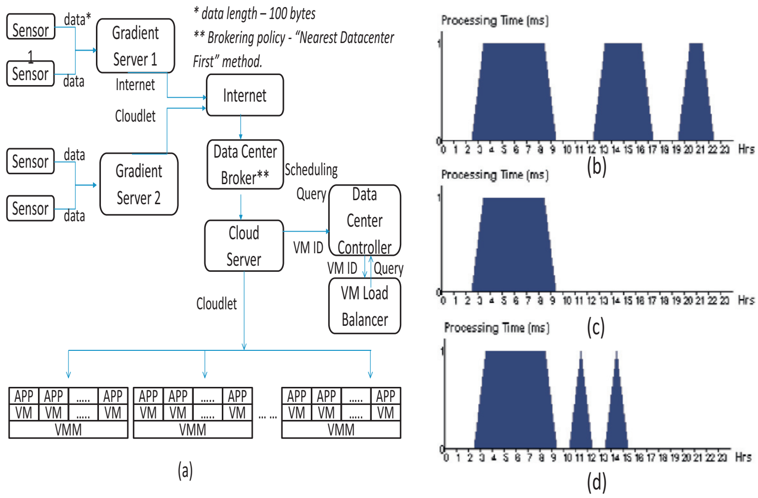

Fig. 7. (a) Gradient server cloud server connection in simulation (b)–(d) Performances of Lightly Loaded Nodes.

in Fig. 6 where the time is considered in seconds. It is observed that the overhead is negligible and we can rely of this messaging system since our country is well covered in cellular network.

5.2. Gradient server and cloud server

We have considered the Meghna basin in our current experiments. The Meghna system originates in the hills of Shilong and Meghalaya of India. Other rivers like Barak, Amalshid, Surma and Kusiara rivers join in the journey in this river system which are mostly highly flashy and steep rivers originated where the highest aveage waterfall is recorded (Cherapunji in Asam, 10,000 mm annual rainfall).

We have used CloudSim simulation tool to design and evaluate our cloud server. We have a simple version of the Virtualized data center where *Virtual Machines* (*VM*) are the basic computing block. Here the *Userbase* defines the overall behavior of the users in a geographic region of the world. The block diagram is presented in Fig. 7. In our simulation, for simplicity, we combine the user requests from huge number of users in a Userbase to group them to internet cloudlets. These internet cloudlets are sent to Data Center via internet. As can be seen in this simulation we used 100 bytes for user request length and for Brokering policy we used "Nearest Datacenter First" method. The broker is able to make independent decision on how to distribute the load among VMs.

We have considered two distinct scenarios in our study where the server is lightly loaded which reflects the current web server load located in Flood Forecasting Center. The other scenario of consideration the scenario when the server is heavily loaded considering a situation that may arise during a flood or when a flood is imminent and there are many hits to the central server. The server setup under various load are discussed below. Although the current web server has not incurred such traffic so far, we have designed our assumptions taking other Government web servers used to publish public data (e.g., Server that publishes Secondary School Exam results). We have set 6 user bases which incorporate sensors in the architecture, with 10 requests per user per hour. data size per byte set to 100, peak hour starts at 3 and ends at 6 GMT, average peak users being 1000 and off peak users being 100. Data Centers have 5 brokers, number of virtual machine is set to 50, image size 10,000, memory size 2048 (MB). For data centers we also have considered x86 architecture, Linux Xen VMM and cost per VM to be 0.1 $/s, memory cost to be 0.05 $/s, data transfer 0.1 $/s for our cost analysis. We have used round robin scheduling for load balancing. Heavily loaded system considers similar parameters as the lightly loaded system parameters except for some changes. We have considered 60 requests per user per hour (as opposed to 10), having peak starting at 3 and ending at 5 GMT, average peak users being 10 times more than lightly loaded system having peak and off peak users to be 10,000 and 1000 respectively.

We have an average response time of 50.69 ms for lightly loaded cloud nodes where the minimum and maximum response time is 35.88 ms and 65.40 ms, respectively. Individual response time at user bases remain close to 50 ms throughout the execution time as shown in Fig. 7(b)–(d). The data center service time is around 1 ms on average with as low as 0.07 ms and as high as 1.62 ms. The processing time varies over time dpending on the assigned load as can be seen in Fig. 7 (a)–(c). Every data center of our consideration does not provde more than 10,000 requests per hour over the functioning time. In general, our simulation study has estimated a total virtual machine cost of 720.22 in dollars and data transfer cost of 8.98 dollars which is affordable considering the data security and reliability.

Our study on Shonabondhu system provides an overview of the system along with its large scale deployment feasibility. Although most of our large scale experiments are studied using emulated sensors, we have seen tolerable latency of the system in terms of communication overhead among servers.

6. Conclusion and future scope

Sensor based distributed system enables great way to handle natural disasters such as flash flood. The solution approaches in developing country requires specific attention to low cost solution approaches along with attention paid to the limited available infrastructure. The problem of flash flood impacts Bangladesh, a developing country with a network of rivers. We have presented our system named Shonabondhu, a distributed gradient sensing middleware to handle flash flood in our country.

Shonabondhu works in two plains—on the lower plain it utilizes low cost sensing modules that are intelligent and able to make local decisions if there is sudden water level rise; the upper plain includes the servers and cloud system that gathers data from sensors and analyzes them for long time prediction and decision making. The innovation of the system comes in the way the system is able to perform in regions where physical network links are not present. The system utilizes existing cellular network deployment for SMS based periodic message transfer among sensing nodes. We have placed careful attention to ensure low cost sensing that are locally available in Bangladesh as sensor availability is a common problem for economically backward countries.

We have shown the feasibility and scalability of a large scale system utilizing cloud infrastructure using CloudSim simulation tool and small scale deployment and system level studies of Shonabondu in our presented research work. Our field tests and laboratory test brings promises regarding the opportunities of using the system in Bangladesh. We have worked with the Water Development Board of Bangladesh to work on a practical solution across the river banks.

We hope to deploy a working prototype in several locations of the river network of Bangladesh and provide long term evaluations. We hope to extend the idea in countries that are challenged in the same ways.

Acknowledgment

The project is funded currently by an internal funding of North South University.

References

[1] Hughes D, Thoelen K, Horré W, Matthys N, Del Cid J, Michiels S, et al. LooCI: a loosely-coupled component infrastructure for networked embedded systems. In: proceedings of the 7th international conference on advances in mobile computing and multimedia (MoMM '09). New York, NY, USA: ACM; 2009. p. 195–203.
[2] List of Flash Flood, Web link: http://en.wikipedia.org/wiki/List_of_flash_floods (Last accessed on March 2014).
[3] Ahmed N. Gradient Sensor Middleware System design for Flash Flood prone River banks in Bangladesh. Grace Hopper Celebration Women Comput 2013.
[4] Basha EA, Ravela S, Rus D. Model-based monitoring for early warning flood detection. In: proceedings of the 6th acm conference on embedded network sensor systems (SenSys '08). New York, NY, USA: ACM; 2008. p. 295–308.
[5] Stoianov I, Nachman L, Whittle A, Madden S, Kling R. Sensor networks for monitoring water supply and sewer systems: Lessons from boston. In: Water distribution systems analysis symposium 2006; 2008. p. 1–17.
[6] Jirapon S, Ootamakorn C. Real-time flood monitoring and warning system. Songklanakarin J Sci Technol (SJST) 2011;33(2):227–35 Mar-Apr. 2011.
[7] Ahmed N, Khan MR, Syrus MA. Shonabondhu: a sensing middleware to handle flash flood. ACM Sensys; 2015.
[8] Islam M, Islam T, Syrus M, Ahmed N. Implementation of flash flood monitoring system based on wireless sensor network in Bangladesh. International conference of informatics, electronics and vision. ICIEV; 2014.
[9] Ahmed N, Azad AK, Khan MR, Habib A, Ghosh S, Shahid S. ShonaBondhu: a cloud based system to handle flash flood. In: Networking systems and security (NSysS), 2016 international conference on. IEEE; January 2016. p. 1–6.
[10] Hughes D, Greenwood P, Coulson G, Blair G. Gridstix: Supporting flood prediction using embedded hardware and next generation grid middleware. In: Proceedings of the 2006 international Symposium on on world of wireless, mobile and multimedia networks. IEEE Computer Society; June 2006. p. 621–6.
[11] Rekhis S, Ellouze N, Boudriga N. A wireless sensor network based water monitoring system. In: Proceedings of the 8 h ACM symposium on QoS and security for wireless and mobile networks (Q2SWinet '12). New York, NY, USA: ACM; 2012. p. 33–40.
[12] Chen D, Liu Z, Wang L, Dou M, Chen J, Hui Li. Natural disaster monitoring with wireless sensor networks: a case study of data-intensive applications upon low-cost scalable systems. Mob Netw Appl 2013;18(5):651–63 October 2013.
[13] Seeger C, Buchmann A, Laerhoven KV. Wireless sensor networks in the wild: three practical issues after a middleware deployment. In: Proceedings of the 6th international workshop on middleware tools, services and run-time support for networked embedded systems (MidSens '11). NY, USA: ACM, New York; 2011 Article 1, 6 pages.
[14] Park C, Kim S-H, Telci IT, Aral MM. Designing optimal water quality monitoring network for river systems and application to a hypothetical river. In: proceedings of the winter simulation conference (WSC '10), björn Johansson, sanjay Jain, and jairo montoya-torres (Eds.). Winter Simulation Conference; 2010. p. 3506–13.
[15] Ye F, Zhong G, Lu S, Zhang L. Gradient broadcast: a robust data delivery protocol for large scale sensor networks. Wireless Netw 2005;11(3):285–98.
[16] Faruque J, Psounis K, Helmy A. Analysis of gradient-based routing protocols in sensor networks. In: International conference on distributed computing in sensor systems. Springer Berlin Heidelberg; 2005. p. 258–75.
[17] Lin H, Lu M, Milosavljevic N, Gao J, Guibas LJ. Composable information gradients in wireless sensor networks. In: Proceedings of the 7th international conference on Information processing in sensor networks. IEEE Computer Society; April 2008. p. 121–32.
[18] Chen YYM, Accardi A, Kiciman E, Patterson DA, A. Fox, Brewer EA. (2004). Path-based failure and evolution management.
[19] Ammons G, Larus JR. Improving data-flow analysis with path profiles. ACM SIGPLAN Notices May 1998;33(5):72–84.
[20] Ahmed N, Kumar R, French RS, Ramachandran U. RF2ID: a reliable middleware framework for RFID deployment. In: Parallel and distributed processing symposium, 2007. IPDPS 2007. IEEE international. IEEE; March 2007. p. 1–10.
[21] Buyya R, Ranjan R, Calheiros RN. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: challenges and opportunities. In: Int. conf. high performance computing & simulation. Germany: Leipzig; June; 2009. p. 1–11.
[22] Sarker AA, Rashid M. Landslide and flash flood in Bangladesh. In: Disaster risk reduction in Bangladesh. Springer; 2013. p. 165–89. Chapter 8.
[23] Arduino, Weblink: http://www.arduino.cc/ (Last accessed on March 2014).
[24] Ultrasonic Rangile Module, Weblink: http://www.micropik.com/PDF/HCSR04.pdf (Last accessed on March 2014).
[25] Honeywell, Liquid Level Sensor, Weblink: http://sensing.honeywell.com/index.php?ci_id=4128&la_id=1&Ntk=si_all_products&N=1235&Ntt=ll103101 (Last accessed on March 2014).

Nova Ahmed: Nova Ahmed is an Assistant Professor of North South University, Bangladesh. She is leading the project named Shonabondhu over a period of four years along with her talented students of the same institute. Her research interest lies in Distributed Computing, Low Cost Sensing, Computing for Good and Human Centered Computing.

Shuvashis Ghosh: Shuvashis worked on Shonabondhu project as an undergraduate student of North South University, Bangladesh. He has presented the work in prestigious international venues like MobiSys, 2016 and local venues NSys 2016. He has worked on the project when it was in its infancy.

Rifat Ahmed Hassan: Rifat worked in the team of three students on Shonabondhu and has presented the work at Mobisys 2016. He worked on the project as an undergraduate student of North South University, Bangladesh. He has been contributing to the project as a Research Assistant.

Sian Iftekhar Galib: Galib worked as an undergraduate student of North South University, Bangladesh.

A.K. Azad: Azad worked on the project as an MS student of North South University, Bangladesh. He contributed on the Cloud based system of this work.

Minhaz Ahmed Syrus: Syrus one of the founding member of the Shonabondhu project who worked as an undergraduate student at North South University, Bangladesh. His contribution lied in the baseline networked system of Shonabondhu. He is an emerging researcher having published in venues such as Sensys, Ubicomp and Compsac working on developing country centric problems.