Embedded System for an Automated, Heterogeneous Flood Forecasting System

Submitted by: Subhajit Sahu Supervised by: Debiprasad Priyabrata Acharya

Project Report

Embedded System for an Automated, Heterogeneous Flood Forecasting System

## Introduction

In June 2013, a multi-day cloudburst centered on Uttarakhand, and caused devastating floods and landslides in India as the worst natural disaster since the 2004 Indian Ocean tsunami. According to the figures provided by the Uttarakhand government, more than 5,700 people were “presumed dead”, which included 934 local residents. Destruction of bridges and roads left about 100,000 pilgrims and tourists trapped in the valleys, leading to three of the four Hindu Chota Char Dham pilgrimage sites. The Indian Air Force, the Indian Army, and paramilitary troops evacuated more than 110,000 people from the flood ravaged area.

Fig 1: Soldiers rescuing people in Uttarakhand

Floods are the most common form of natural disaster in India, and cause huge loss of life, and property. The Central Water Commission (CWC), India has installed around 175 flood forecasting stations in India, with only 3 forecasting sites in the Uttarakhand state. Due to this insufficiency of flood forecasting stations, CWC was only able to forecast a medium-level flood at Rishikesh and Haridwar on June 18, 2013. This led to a huge loss of life, and expenditure of around ₹1100 crore.

## Requirement of an Automated Flood Forecasting System

Today, there exist quite a number of flood forecasting algorithms, developed by researchers. However, we mostly lag behind in the availability of state of the art infrastructure required for flood forecasting. Most of the flood forecasting systems currently installed at various sites in India are manual. Thus, the overall flood prediction system is deprived of a proper integration, which is required for accurate modeling and forecasting. Again, there is no available system that allows people to know about the flood danger level of a location in real-time (such a system is available for trains). In this project, we propose to work on the design of an automated flood forecasting system that is capable of collecting data from installed sites continuously, and making it readily available to the experts for flood forecasting.

## Overview of a Flood Forecasting system

A forecasting system works by monitoring a system, and accordingly predicting the next state of the system. In case of a flood forecasting system, the parameters which can cause a flood, are continuously monitored. These parameters mostly include river water level, rainfall, ground saturation, glacier temperature, wind and humidity. These parameters are generally collected at various critical points, mostly near rivers, and are called flood forecasting sites. Data collected from these sites are provided to experts, which help them understand the state of the river system or surrounding. A flood prediction model is then applied to predict the future state, and thus detecting any chances of occurrence of flood.

## Overview of the Proposed System

As mentioned before, the current system in use is mostly manual, inefficient, and requires automation. We propose to design wireless flood forecasting sensor systems that can be installed at various flood forecasting sites. These sensor systems would then communicate with a set of centralized servers that would monitor and manage the overall system. Easy access to the data collected on the servers would be provided to both flood forecasting organizations and the general public through a direct interface, or a standard web interface. Fig. 2 shows as an example the setup of automated and heterogeneous flood forecasting sites in River Mahanadi.

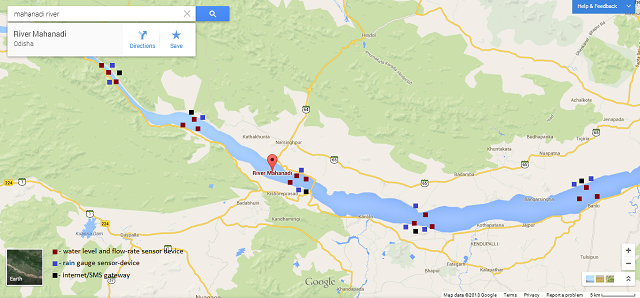


Fig. 2: Wireless sensor systems setup at various sites in River Mahanadi, as an example

## Data collection from sensor systems

The wireless sensor networks, which would be placed at strategic points in a river’s drainage basin, would consist of a number of wireless sensor-devices and one or more internet gateways. Fig. 2 shows as an example, the wireless sensor systems setup at various sites of river Mahanadi. All the sensor-devices and internet gateways would be battery powered, and would also include an energy harvesting unit for continued operations. Each sensor-device would have one or more sensors connected to it, and would collect data from each sensor at certain intervals of time. While no data is being collected, the sensor-device would power-down itself, in order to save energy. Similarly, if the data collected is repetitive in nature, then it would not be sent, so as to save power. Multiple sensor-devices would also be used to detect faulty sensors, and report it to the application server for hardware repair. All sensor-devices would forward their data to an internet gateway, whose job is to appropriately forward the data to an application server. The internet gateway would also act as a router for the local sensor-device network, managing the transmission of data packets from one sensor-device to another or an application server, or vice versa.

## Data Storage, Access and System management

Application servers, which would receive data from all sensor-devices store and manage the collected data. These servers would also allow various clients, including web servers, to obtain collected data, process it, stream some new (generated) data, or send messages to the sensor-devices. These servers would cooperatively work with each other, if required, in order to complete a requested operation. It is also the task of these servers to ensure security at every level of the system, starting from sensor-devices, to the clients. Direct access platforms (software libraries), for communicating with application servers, would be designed for languages such as C/C++, C#, Java and MATLAB to allow experts to easily load collected data directly from the application servers, to their program, which can be used by them for implementing automated flood forecasting models. Web servers would be designed, which would provide an easy to use platform to access the flood forecasting system, to both the experts, and the general public. Fig. 3 shows the overall network structure of the automated flood forecasting system.

Once the entire system is ready, it would be possible to:

1. Collect data from a large number of forecasting sites, without having to manually visit any site.
2. Run automated flood forecasting programs from anywhere in the world.
3. Monitor and install firmware upgrades to the entire forecasting system, by a single click.
4. Report the status to all other experts and to the general public in real-time.

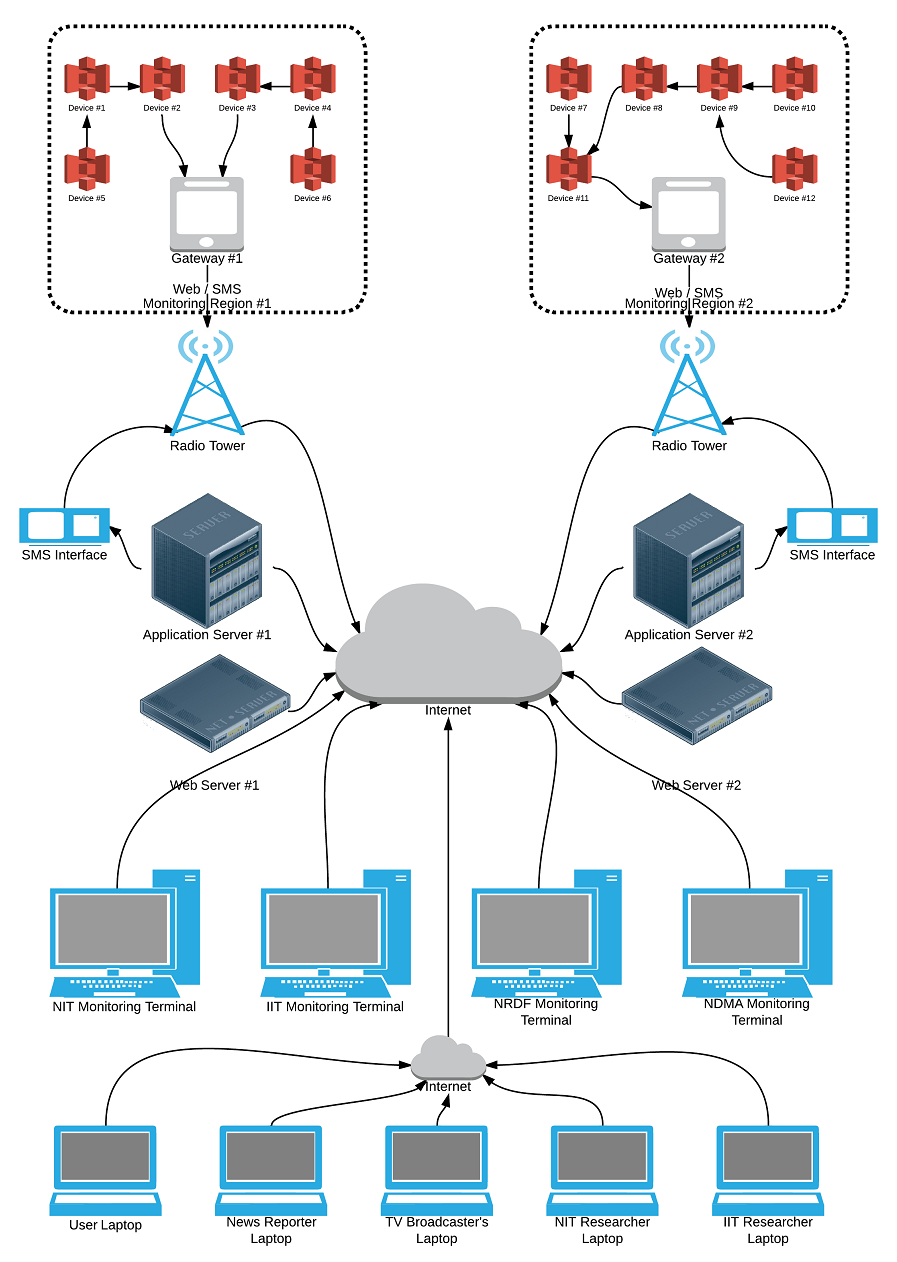


Fig. 3: Network structure of the automated flood forecasting system

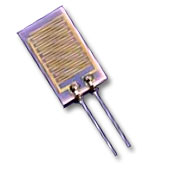
## Scope of the Project

The aim of this project is to design and implement an Embedded System for an Automated, Heterogeneous Flood Forecasting System. This includes the design of low-power, cost-effective wireless sensor-devices, internet gateways, application servers, web servers, and direct data access platforms. Implementation of flood forecasting algorithms is however, not a part of this project.

## Hardware Architecture

## Parameter measurement using Electronic Sensors

Flood forecasting requires the measurement of certain parameters at various locations in the drainage basin of a river. Automated measurement of these parameters can be performed through the use of electronic sensors that are capable of measuring the physical value of a parameter, and converting it to an equivalent electrical signal, in analog or digital form. Fluid pressure sensors and Ultrasonic range sensors can be used for measuring the river water level at a site. Additionally, the water flow rate at the location can be measured by a pair of pressure sensors. Digital rain gauges can be used to measure the rainfall at a location, Soil wetness sensor can be used for finding saturation level of the ground, Infra-Red temperature sensors can be used for measuring glacier temperature, humidity sensor for measuring humidity, and a DC fan pair for measuring wind speed and direction. Fig. 4 shows various electronics sensors that would be used for measuring the parameters.



Wind sensor

Humidity sensor

Soil wetness sensor

Temperature sensor

Digital rain gauge

Fluid pressure sensor

Fig. 4: Electronic sensors used for measuring parameters required for flood forecasting

## Sensor-device for data collection

One or more of the sensors mentioned above would be connected to a sensor-device. The core of each sensor-device is a low-power microcontroller which controls the operations of the device. It obtains data from the sensors connected to it, manages power supply to all parts of the device through a power supply unit (with the objective of minimizing power consumption) and communicates with other sensor-devices and internet gateways through an RF unit.

## Power Supply unit

A Lithium-ion or a lithium-polymer rechargeable battery is used for power the sensor device. The 3.7V output voltage of the battery lowered to 3.3V for powering up the microcontroller. The output voltage is also raised to 5V, which is required by most electronic sensors for their operation. The power supply unit which does this task of voltage regulation also allows the microcontroller to turn off parts of the sensor-device, when not in use. Power supply unit also enables the microcontroller to monitor the amount of charge left in the battery, which allows it to make intelligent decisions regarding data collection, storage and transmission. It also enables the device to report its battery status to the application servers. Fig. 5 show the block diagram of a sensor-device.

GPRS modem

Sensor 2

Sensor 1

Microcontroller

Sensor 4

Sensor 3

Battery

Power supply module

Fig. 5: Overall block diagram of a sensor-device (the microcontroller has an internal RTC)

## Energy harvesting unit

Each sensor device also includes an energy harvesting unit, which allows the device to generate power from the surroundings. This is required for prolonged and uninterrupted operation of the device and helps lower maintenance costs. The energy harvesting unit is connected to the power supply unit, which up-converts the generated voltage to a value which can be used to recharge the battery.

## Wireless Communication unit

The sensor device has a wireless communication unit, which could be a standard 434MHz RF unit, a standard 2.4GHz ISM unit (such as CC2500), or a 2.4GHz wireless PAN unit (such as XBee 802.14.5 / ZigBee). The wireless unit allows the microcontroller to send the collected data to the core of the flood forecasting system. Making use of a wireless communication unit for each sensor device allows for easy setup, maintenance, upgrades and replacement. Since transmission and reception of data wirelessly is slow, and consumes much power, the microcontroller wisely decides when to turn on the RF unit, when to send data, and which data to send. All sensor nodes however require an internet gateway in near vicinity, which helps the local group of nodes to set up a local network, and communicate with each other over a multi-hop network, just like the internet.

## Other Required Modules

Each sensor-device also has a Real Time Clock (RTC) which helps the microcontroller to keep track of time, and record the data collection time of every sample. Optionally, each sensor device may also have an SD card slot for data storage. Each sensor node also has a USB programming port, which allows the firmware of the device to be updated local (on-site). The sensor-device would however be pre-programmed with a boot-loader, which would enable the device to be programmed wirelessly. Since the internet gateway connects the sensor-devices to the application servers, which can be accessed from anywhere, the sensor-devices’ firmware can be upgraded from anywhere in the world. The sensor device would have a set of LEDs which allow the microcontroller to display its state of operation.

## Internet Gateway for data transmission

The internet gateway, whose hardware architecture is very similar to that of sensor-device (however, it may not have any attached sensor), serves as a local network manager, and allows them to easily communicate with application servers, or to any other local sensor-device. It also has a GSM/GPRS or Wi-Fi unit that allows it to act as a “gateway” for the sensor-devices to the internet.

## Network Configuration

In cases, where the internet gateway is close enough to a Wi-Fi network, the gateway is made to have a Wi-Fi unit. However, in most cases, since Wi-Fi network may not be available at the usage sites, the GSM/GPRS unit is used for internet access. In order for the GSM/GPRS unit to work, and SIM card is attached to it and is configured with appropriate GPRS settings required to access internet.

## Network Operation

The gateway establishes a TCP connection to an application server, whenever it is required to send data. It also connects with an application server to receive any available notifications for any device on its local network, if any. The internet gateway however requires a high-capacity and high-power battery, or a direct power connection, because it performs a lot of networking activities through the GSM/GPRS unit or the Wi-Fi unit, both of which require large amount input supply current, for transmission or reception.

High capacity Battery

RF unit

Sensor 2

Sensor 1

Power supply unit

Microcontroller

Energy harvesting unit

GSM/GPRS or Wi-Fi unit

Optional direct AC power supply

Fig. 6: Overall block diagram of an internet gateway

## Application Server for System management

The application servers, which would be located far away from the sensor systems, in an office, would do the job of managing the entire flood-forecasting system. All the data collected by sensor systems are sent to these servers, which cooperatively collect and manage the stored data. These systems must have a high-speed network connection, huge storage space, and must be able to support a large number of simultaneous connections.

## Web Server for easy access

There would be web servers that would provide a standard easy-to-use web interface for monitoring and controlling the entire system. Both the application server, and the web server must have a public IP address so as to accessible to the sensor systems and user computers.

## Software Architecture

Unlike hardware, which is visible to us, software is the invisible soul which drives the hardware. A well designed software can make a system work like magic, and a bad one can bring about a catastrophic failure. When building a system, it is often a requirement that various software must work with each other to accomplish a particular task. In such a situation, operational and communication standards / protocols must be well defined in order to ensure that each software knows how to cooperate with another in all possible cases. Setting up a unified model for the operation of the system makes it simpler, thus enhancing its usability.

## Entity-oriented model for the system

We propose the automated flood forecasting system to be modelled as an entity-oriented model. According to this model, the system is a set of entities, with each entity having its own data, and a set of associated operations. Each individual sensor, be it physical or logical, is considered to be an entity in this system.

## Logical sensors

A logical sensor means that the sensor does not physically exist, but provides a sense of its existence. As an example, if a sensor-device has an instantaneous voltage and current sensor, both of which are physical, values obtained from both these sensors can be used to calculate instantaneous power, an important piece of information, which is considered to have been measured by a logical sensor (that does not exist physically).

## Members of an Entity

Now, each entity is of a particular type, and according to its type, has a set of data-fields and associated functions. As an example, there could be multiple temperature sensors, each measuring temperature at a different location, but all of these sensors are of the same type and therefore must be having similar associated properties. Each entity, based upon its type, would also have a set of associated functions which allow it to perform a specific operation such as storing data on a server, or retrieving the last message sent from its owner.

## Similarity of this model to the Object-oriented model

This model is very similar to the object oriented programming model used in programming languages, where an entity is similar to an object, and its type is similar to an object’s class. These associated properties of an entity are similar to that of the data-members of a class, and the same goes with the associated functions.

## What Entity models, and its usage

Apart from a single sensor, a sensor-device as a whole is also considered an entity. A group of sensors, or a group of sensor devices are also considered as entities. Groups of groups are also considered as entities. Functions invoked on a group (by the owner of the group) causes that function to be invoked on all members of the group which have that function available. An example of this could be upgrading the firmware of all sensor-devices in a group, and retrieving their current battery level.

## Entity Identifiers

We propose to identify each entity with a unique 250-bit Entity ID. Also each entity type is given a unique 123-bit Type ID, and each function is given a unique 123-bit Function ID (properties can only be accessed through functions). However, since full identifiers may not always be required for identification, short 7-bit/14-bit/29-bit/60-bit/123-bit identifiers would be supported by the system.

## Entity Access protection feature

In order to provide protection from data loss, corruption, or illegal access, each entity is only allowed to access a set of properties of its subordinates. This feature of entities can be used to create read-only views of certain fields of any entity. This is equivalent to assigning users with different access rights. For example, weather monitoring stations can group all the rainfall sensors together as an entity, and then create a read-only entity out of it to allow the general public to have access to real-time rainfall data, but disallow them from reprogramming the firmware loaded into their sensor-devices.

## Flexible Number format for the system

In order to allow for the use of short identifiers within the entire system, which is required for reducing communication overload, a number format has been proposed. According to this format, the lowest bits are reserved for identifying the size of the number value being used. The number of 1’s, starting from the least significant bit, to the higher significant bits is used to indicate to size of the number value in bits. The size of the entire number structure is bits, where n is the number of 1’s, and the size of the number value is bits. Hence, this number format can be used to represent number values of sizes 7, 14, 29, 60, 123, 250 bits and many more. As a result, this number format provides a good flexibility in representing numbers. All identifiers and numbers mentioned afterwards shall be using this format.

**1**

**1**

**1**

**1**

**0**

**Number**

The number of 1’s indicate the size of the number. No 1’s indicate an 8-1 = 7bit number. Here 4 1’s represent a 128-5 = 123bit number.

The number value, which is obtained by shifting right the number by appropriate number of bits. In this case, the number must be right-shifted by 5 to obtain the number value.

0 is used as a stop bit

Fig. 7: Flexible number format that is proposed to be used with this system

**References**

* Xiang Yang Li, K. W. Chau, Chun Tian Cheng, Y. S. Li. A Web-based flood forecasting system for Shuangpai region.Advances in Engineering Software 37 (2006), 146-158.
* Bartosz Balis, Marek Kasztelnik, Marian Bubak, Tomasz Bartynski, Tomasz Gubala, Piotr Nowakowski, Jeroen Broekhuijsen. The UrbanFlood Common Information Space for Early Warning Systems. Internaltional Conference on Computational Science, ICCS 2011.
* Ibrahim Demir, Witold F. Krajewski. Towards an Integrated Flood Information System: Centralized data access, analysis, and visualization. Environmental Modeling & Software 50 (2013), 77-84.
* V.V. Krzhizhanovskaya, G.S. Shirshov, N.B. Melnikova, R.G. Belleman, F.I. Rusadi, B.J. Broekhuijsen, B.P. Gouldby, J. Lhomme, B. Balis, M. Bubak, A.L. Pyayt, I.I. Mokhov, A.V. Ozhigin, B. Lang, R.J. Meijer. Flood early warning system: design, implementation and computational modules. International Conference on Computational Science, ICCS 2011.
* W. Al-Sabhan, M. Mulligan, G.A. Blackburn. A real-time hydrological model for flood prediction using GIS and the WWW. Computers, Environment and Urban Systems 27 (2003), 9-32.
* Design and FPGA Prototyping of a Flood Prediction System. Sandeep Gandla, Waleed K. Al-Assadi, Sahra Sedigh and Raghu A. R. Rao.