View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by CiteSeerX

Sensors & Transducers, Vol. 153, Issue 6, June 2013, pp. 209-218

www.sensorsportal.com

Sensors & Transducers

© 2013 by IFSA

http://www.sensorsportal.com

A Web Service Interface for a Distributed Measurement System Based on Decentralized Sharing Network

[1] Fabrizio Ciancetta, [1] Edoardo Fiorucci,
[2] Daniele Gallo, [2] Carmine Landi, [2] Mario Luiso
[1] Dipartimento di Ingegneria Industriale e dell'Informazione e di Economia, Università dell'Aquila
Via G. Gronchi 18 – Pile, 67100 L'Aquila, Italy
[2] Dipartimento di Ingegneria Industriale e dell'Informazione, Seconda Università degli Studi di Napoli
Via Roma, 29 - 81031 Aversa (CE), Italy
Tel.: +39-0815010375, fax: +39-0815037042
E-mail: [1] {fabrizio.ciancetta, edoardo.fiorucci}@univaq.it,
[2] {daniele.gallo, carmine.landi, mario.luiso}@unina2.it

Received: 2 May 2013 /Accepted: 14 June 2013 /Published: 25 June 2013

Abstract: The Web Service technology has increased in importance in these years. Accessing to remote resources without knowledge about physical implementation and with a reduced hardware requirement is the main goal of new portable device. The use of Web Service technology allows clients to create a standard interface to access to the measurement service published by the server. In low cost multipoint distributed measurement systems, the measurement services are provided by Smart Web Sensors. A new concept of distributed measurement system arises from the possibility to fuse all services with the same functionality in a single user-transparent service: the response of distributed services network to a user request gives a complete vision of the service by collecting results from any Smart Web Sensor in the network. So, in this paper the development of a multipoint distributed measurement system, based on the peer-to-peer Gnutella network with Web Service interface, is presented. *Copyright © 2013 IFSA.*

Keywords: Smart sensor, Distributed measurement system, Peer-to-peer, Web service, Gnutella.

---

1. Introduction

Several applications require a distributed measurement system able to measure the same or different parameters at different points in a wide area. In recent years, a wide variety of solutions have been proposed for the remote measurement and data transmission.

Distributed systems based on smart web sensors represent the best solution to many different measurement problems. By adopting these sensors, a client can receive the measure of a particular physical quantity with a browser or an application, developed to receive information from the web server [1]-[2]. Distributed systems can be basically grouped in two categories.

The first, widely adopted, approach is based on a number of smart measuring devices, linked with a centralized system, a central server keeping a list of users and shared resources. The primary advantage of centralized systems is their simplicity. Because all data is concentrated in one place, centralized systems are easily managed and have no problems of data consistency or coherence. During a search, every client sends a request to the central server that consults its lists providing results of IP user addresses. The file downloading happens between the two interested users from outside-centralized network. So, the server does not keep up any files. Each system is an independent server and must be selectively interrogated by the clients. The client needs to know the server position on the network (IP address) before starting the operations.

The second approach, a decentralized system, is always based on a number of smart measuring systems, but presents the advantage to make easier the interrogation by the clients. This gives more extensibility to the network in which any node can join the network and instantly make new files available to the whole network. Another important feature of decentralized networks is that the failure or shutdown of any particular node does not influence the rest of the system. On the other hand, the intrinsic nature of this network gives two problems: the difficulties to manage the network because all the nodes have the same hierarchic level and the possibility to have packet loop that causes useless traffic. The user can search all the measurement related information available on the network using dedicated (special) services. Then he can ask to transfer the needed data from one or more measuring systems able to carry out the required measurement. There is no need to know any information related to the server (address list) before starting the search. A hybrid schema is also possible combining centralized and decentralized systems (Fig. 1). Decentralization contributes to the extensibility, fault-tolerance, and lawsuit-proofing of the system, while the partial centralization makes the system more coherent than a purely decentralized system [3].

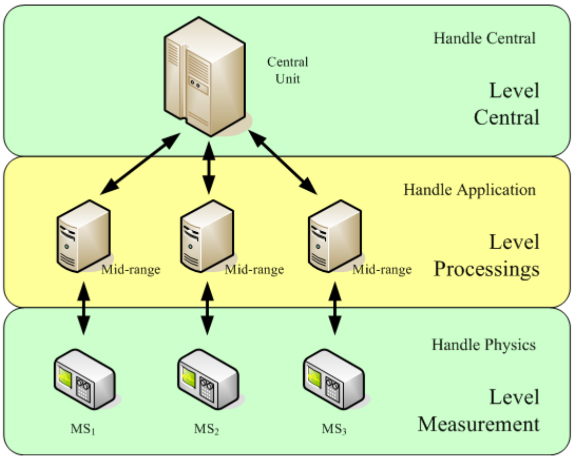


Fig. 1. Hybrid measurement network.

From developer side, smart web sensors present always a closed approach to interact with them, so Web services are adopted in order to give a standard approach in developing a Service Oriented Architecture [4]**Error! Reference source not found.**. From network side, Internet is a widely adopted network where any user is uniquely identified with its IP address. So, to implement a distributed measurement system, it is necessary to use 1) a common and open communication protocol to exchange information and 2) a methodology to auto-configure any smart sensor that is linked to the network [5].

Peer-to-peer networks allow individual computers to communicate directly with each other and to share information and resources without using specialized servers. A common characteristic of this new breed of applications is that they build, at the application level, a virtual network with its own routing mechanisms. The topology of this virtual network and the adopted routing mechanisms has a significant influence on the application properties such as performance and reliability [6].

Significant advantages can be gained using a freeware and widely adopted technology, such as that we adopted, the Gnutella. Our choice has been motivated by its large diffusion (more than 2.2 million users), even if for different kind of applications, such as file sharing [7].

Our aim is to make possible the use of this technology to share measurement information supplied by smart sensors, instead of files, with the same straightforwardness and reduced costs. To analyze the performance of this system, a prototype for measuring environmental parameters has been implemented and discussed in the following.

2. The Implemented Distributed Architecture

In a wide distributed measurement network, the main disadvantages are the difficulties to access single data from a Smart Web Sensor and the necessity to use proprietary technology to obtain information and to communicate through the network.

When a client computer asks for a measurement task to network devices, it has to found some information: the IP address of the servers that can perform the measurement, the list of measurement tasks and sensors available, the server geographic position and the network routing.

In our proposal, we adopted the “Web Service” that gives a new degree of freedom in developing a measurement network, because based on standard communication protocol and software interface (see Fig. 2). Web Service is the fundamental building block in the move to distributed computing on the Internet. Open standards and the focus on communication and collaboration among people and applications have created an environment where Web Service is becoming the platform for application integration. All Web Service communications are done through messaging mechanism of SOAP (Simple Object Access Protocol) that is based on XML that is a Standard Generalized Markup Language (SGML): a language that facilitates the structuring of data in documents. In addition to XML documents, information about name and attributes of each data element are provided, therefore each client can extract, from a message, only the information needed. This, of course, gives to users a great deal of freedom [8].

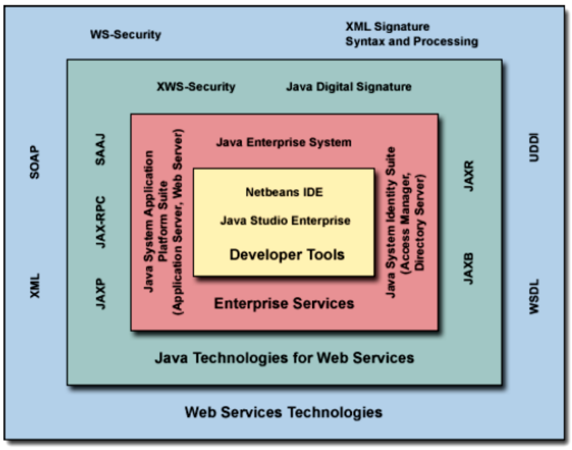


Fig. 2. The Web Service Technology.

As previously introduced, the proposed architecture uses the Gnutella network (see Fig. 3**Error! Reference source not found.**), a network that allows linked hosts to share arbitrary resources [7]. This is a decentralized P2P system, consisting of hosts connected to one another using TCP/IP. In this network a client request for a measurement application is addressed to a computer which performs a particular Web service (Gnutella Web Service). This computer uses the Gnutella network to search all the users able to perform the specific measurement, called Gnutella Embedded Clients (GECs) as reported in Fig. 4. The name client for GEC is because it is a client of the Gnutella network. As shown in Fig. 5, the GEC is revealed by common Gnutella sharing software (Phex).

To execute the user search the request (query message) is repeated to all the Gnutella network computers (Fig. 6). When the suitable user is found, this network sends back the GEC address to the client. At this point, the client can download the measures directly from the GEC, without overloading the Gnutella network [8].

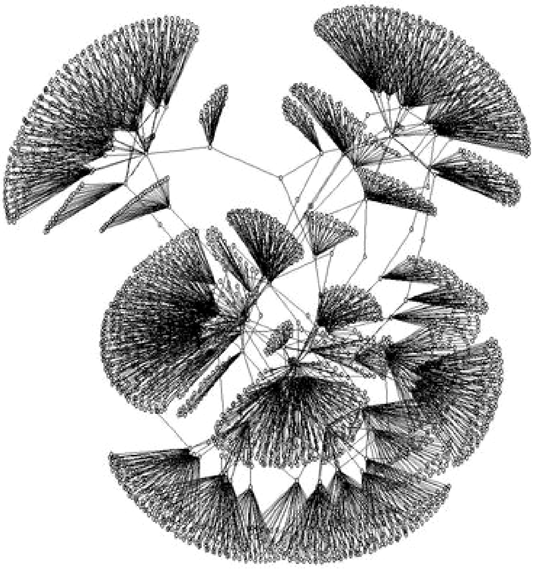


Fig. 3. The Gnutella Network.

In our proposal the measurement points are the GECs. Each GEC can perform special measurements depending on the kind of sensors embodied. When a measurement operation is asked, GEC sends the results to the Gnutella Web Service (GWS).

One of the advantages of the proposed solution is the simplification of the activities to search and locate the measurement systems (GEC). This network creates an Internet over-structure from which all clients can perform a free access without external configuration and the GECs are visible without special operations [9].

In order to implement this kind of system, a special Gnutella Web Service, a kind of interface between the client and the Gnutella network, has been implemented (Fig. 7). The need for this implementation is because the current implementations, referring exclusively on files sharing, cannot support a measurement process.

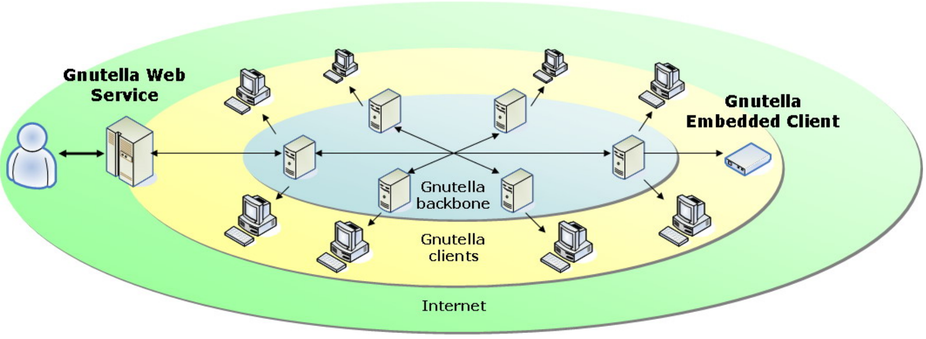
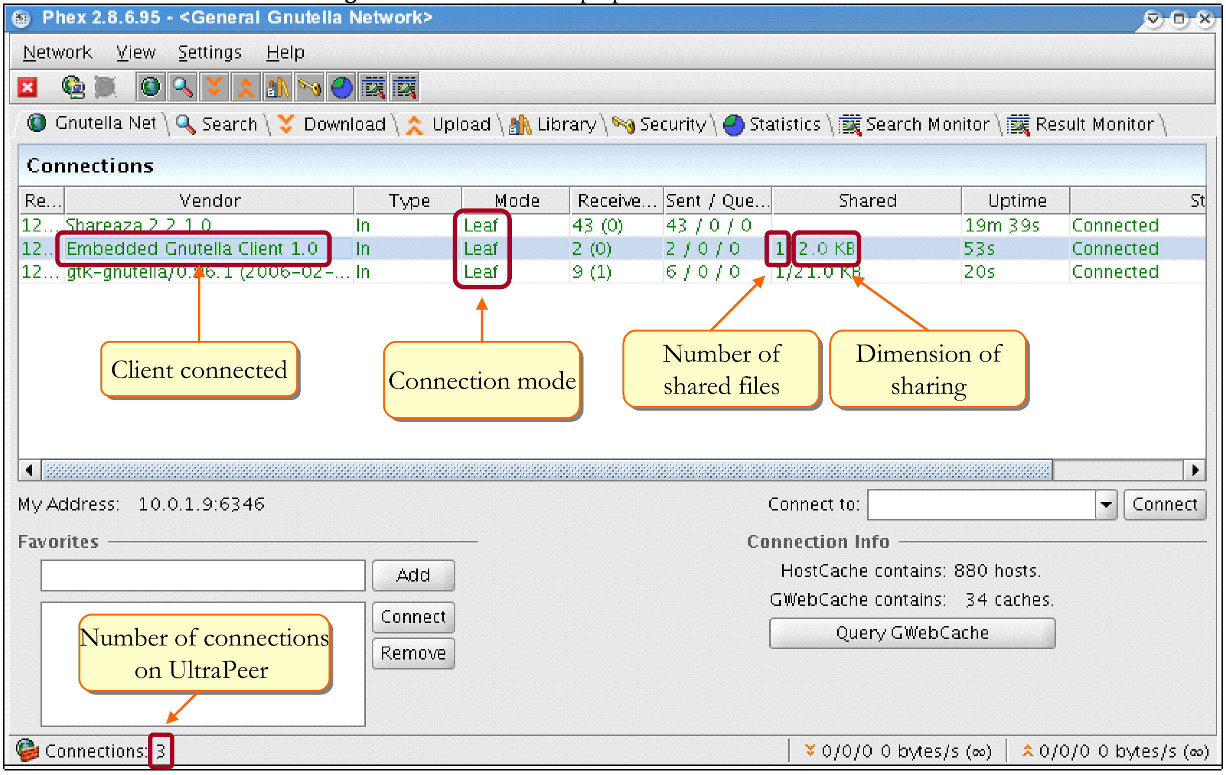


Fig. 4. Architecture of the proposed measurement network



Re...	Vendor	Type	Mode	Receive...	Sent / Que...	Shared	Uptime	St
12...	Shareaza 2.2.1.0	In	Leaf	43 (0)	43 / 0 / 0		19m 39s	Connected
12...	Embedded Gnutella Client 1.0	In	Leaf	2 (0)	2 / 0 / 0	1 / 2.0 KB	53s	Connected
12...	gtk-gnutella/0.96.1 (2006-02-...	In	Leaf	9 (1)	6 / 0 / 0	1/21.0 KB	20s	Connected

My Address: 10.0.1.9:6346

Connect to:

HostCache contains: 880 hosts.

GWebCache contains: 34 caches.

Connections: 3

0/0/0 0 bytes/s (∞)

0/0/0 0 bytes/s (∞)

Fig. 5. GEC presents on the list of Gnutella client in common Gnutella sharing software.

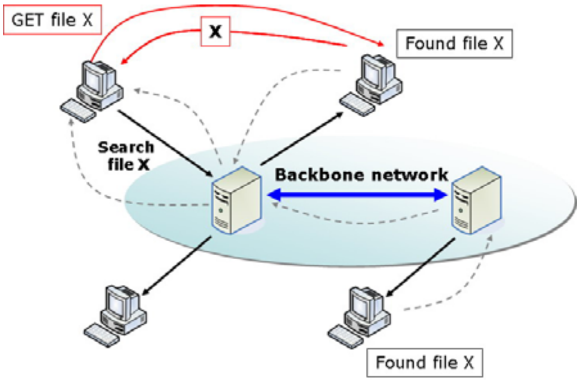


Fig. 6. The measurement server search, route and download.

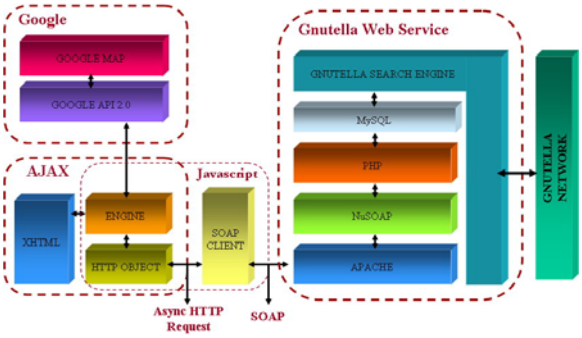


Fig. 7. Architecture of the implemented Web Service and Operator Interface.

3. Web Service TEDS

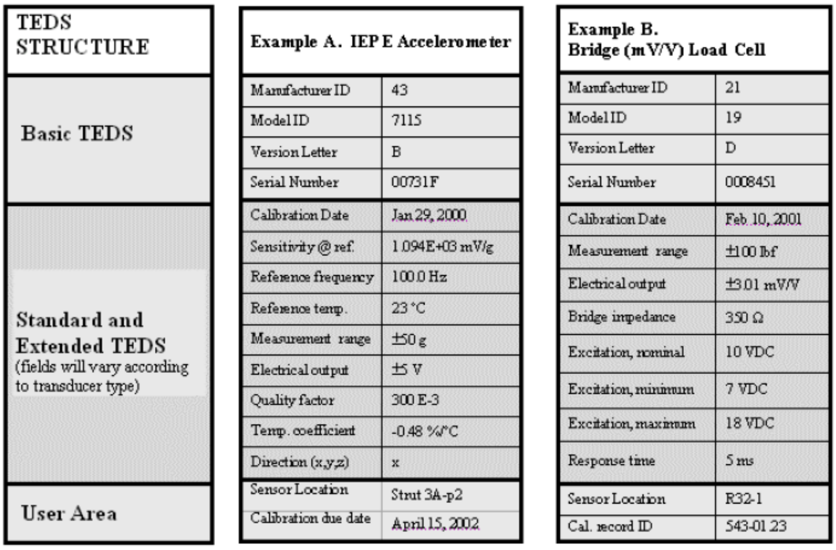
Interfacing transducers to all communication networks and supporting the wide variety of protocols is time-consuming and costly for manufacturers. To simplify this problem a standardized connection methods to interface smart transducers to the existing control networking technology has been proposed by the IEEE 1451 family of standards [9].

The heart of the IEEE 1451.4 standard is the definition of the TEDS (Transducer Electronic Data Sheets), the information structure that contains the critical sensor information to enable plug-and-play operation. The TEDS typically resides in an EEPROM embedded in the sensor and is accessed by the measurement system via a simple low-cost serial interface.

IEEE 1451.4 defines the TEDS structure to be very compact yet flexible and extensible enough to handle a wide range of sensor types and requirements. The TEDS information is divided into several key sections as reported in Fig. 8.

The first part, the Basic TEDS, contains the required sensor identification information, including manufacturer, model and serial number. The Basic TEDS may be followed by an IEEE standard TEDS that contains the specific ‘data sheet’ information – typically the data needed to properly configure the electrical interface and convert the measurement data into engineering units. Typical TEDS parameters include measurement range, electrical output range, sensitivity, power requirements and calibration data. The standard TEDS section describes everything needed to make a measurement using the sensors.

This standard specifies a collection of TEDS formats, defined as templates, for different sensor types. The templates provide the means for the measurement system to convert the binary data stored on a smart TEDS sensor's EEPROM into meaningful specifications. The collection of templates includes IEPE (Integrated Electronic Piezo-Electric) accelerometers and microphones, IEPE pressure sensors, Wheatstone bridge sensors, strain gauges, load and force transducers, thermocouples, RTDs, thermistors, LVDT/RVDT, resistive sensors and amplified sensors (any type) with voltage or current output.



TEDS STRUCTURE

Basic TEDS

Standard and Extended TEDS
(fields will vary according to transducer type)

User Area

Example A. IEPE Accelerometer	
Manufacturer ID	43
Model ID	7115
Version Letter	B
Serial Number	00731F
Calibration Date	Jan 29, 2000
Sensitivity @ ref.	1.094E+03 mV/g
Reference frequency	100.0 Hz
Reference temp.	23 °C
Measurement range	±50 g
Electrical output	±5 V
Quality factor	300 E-3
Temp. coefficient	-0.48 %/°C
Direction (x,y,z)	x
Sensor Location	Strut 3A-p2
Calibration due date	April 15, 2002

Example B. Bridge (mV/V) Load Cell	
Manufacturer ID	21
Model ID	19
Version Letter	D
Serial Number	0008451
Calibration Date	Feb. 10, 2001
Measurement range	±100 lbf
Electrical output	±3.01 mV/V
Bridge impedance	350 Ω
Excitation, nominal	10 VDC
Excitation, minimum	7 VDC
Excitation, maximum	18 VDC
Response time	5 ms
Sensor Location	R32-1
Cal. record ID	543-01 23

Fig. 8. Structure of the IEEE 1451.4 TEDS, with Example Templates for IEPE Accelerometers and Bridge-Based Load Cells.

The IEEE 1451.4 standard defines a Mixed-Mode Interface for Smart Transducers, adding a self-identification technology to traditional analog-mode sensors and actuators to store in TEDS sensor's EEPROM. In some applications the sensor operating conditions prevent the use of any electronics, such as EEPROMs. For these sensors without EEPROM embedded, IEEE 1451.4 contemplates the adoption of Virtual TEDS file. A Virtual TEDS file (Fig. 9) is a file stored on a local computer or a web-accessible database that reports all TEDS information of sensors instead of an embedded EEPROM. This enables the huge installed base of legacy, analog sensors to realize the benefits of TEDS without being retrofitted with an embedded EEPROM.

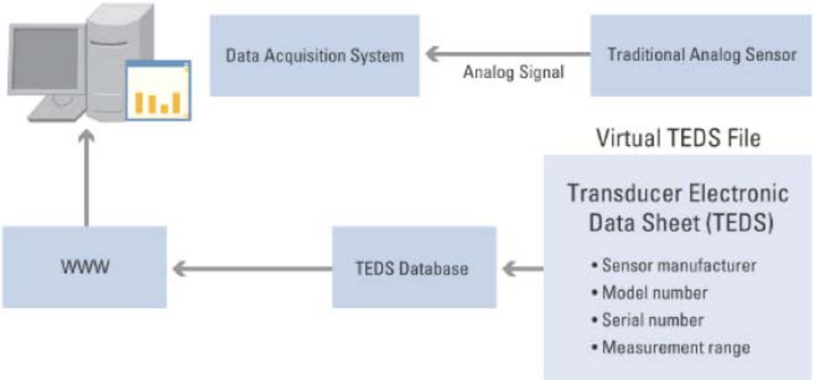


Fig. 9. Virtual TEDS Accessible via a Web Interface.

In the proposed network, we save the information related to the sensors in a binary file stored on the Gnutella Embedded Client Flash memory using the structure reported in Fig. 8. We have developed another Web Service that works with the Gnutella Web Service to require data information. Using the methods exported by this new Web Service, it makes possible to receive and to work on TEDS information downloaded from the GECs. Moreover, the main feature introduced with the new Web Service is presented in Fig. 10 in which the Web Service static class diagram is reported. The Web Service creates new software objects that it allows developers to use the TEDS information without knowing the TEDS structure and so without parsing the TEDS binary file stored on GECs. The main class is the Station1451 that is a collection of Sensor1451 objects.

Every object is composed by: a TEDS Basic object in which the TEDS Basic information are reported, partially in the template objects present in the Template package and partially in the calibration objects present in the Calibration package. This complex structure can be adopted in any Windows Form that references the Web Service making easier the working on TEDS information.

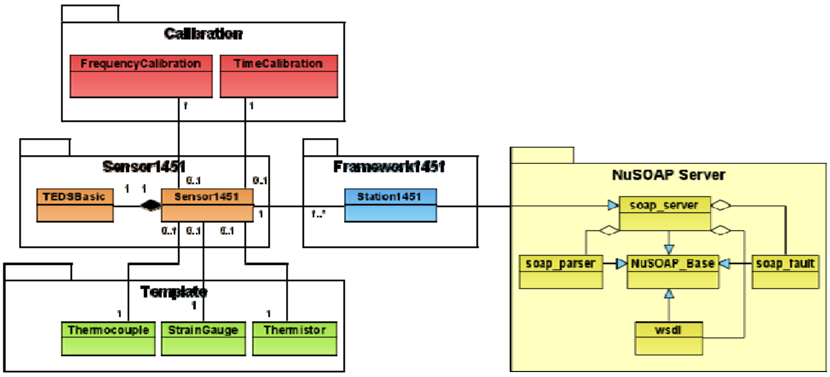


Fig. 10. Web Service class diagram.

4. The Developed Remote Measurement System

4.1. The Authentication Problem

Even if the whole system adopts Gnutella network to exchange information between GECs and GWS, some problems are present for the developing of a remote measurement system. During a GWS search, all the involved GECs respond to the Web Service with the Gnutella packet ([10]) of Fig. 11, without specifying if the supplied data have been locally generated by a measurement system or downloaded from a different client.

To control if the downloaded file corresponds to the right client, we use the HASHID item. Gnutella network was developed to allow users to share files. So, everyone can access to the services present on the GEC for the downloading of measurement information. In fact, in GEC the downloading of a file corresponds to a creation of a XML streaming, reporting the measurement information.

When a user downloads the file, it makes it present in his shared directory, accessible to everyone and, in particular, to the GWS during a searching. Therefore, the problem is that this information is incongruous, so the GWS has to check if the file was generated form GEC or not. The HASHID item contains a hashing of the IP GEC that generated the XML file, using the Message Digest algorithm 5 (MD5) [11]. Therefore, if the file was downloaded from a no GEC client, the hash of Gnutella user IP address mismatches with the HASHID item and the GWS can rejects the downloaded file.

Because the GWS knows the GEC IP address and the file name matched in the searching, when it begins the file downloading out of the Gnutella network, the following procedure is applied:

- Check if the extension file is XML;
- Check the existence of a HASHID item in the XML file;

Control of the HASHID item.

4.2. The Hardware Architecture

In our network the GEC is an embedded system able to perform measurement tasks and to communicate over the Gnutella network. As reported in Fig. 12, it is composed of three sections:

- Input section: transducer and the signal conditioner;
- Micro controller: data acquisition and preprocessing task performed by a dsPic microcontroller;
- Network Embedded System section: processing and network front-end in which a micro Gnutella kernel has been developed to communicate with the Gnutella network using a networking embedded system (Fox Board).

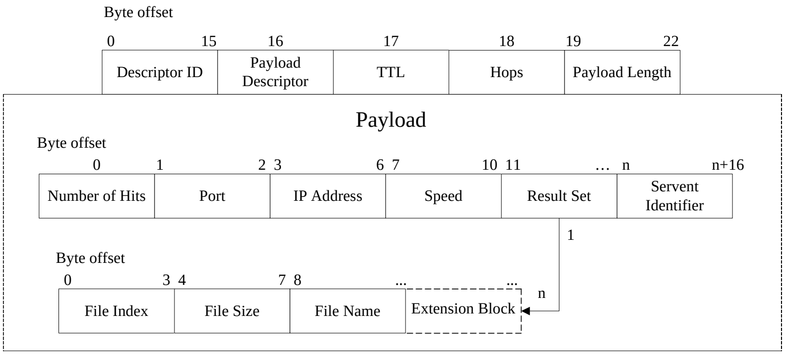


Fig. 11. Hybrid measurement network. Gnutella response packet during a search.

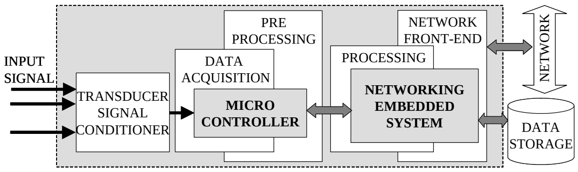


Fig. 12. GEC hardware architecture.

Moreover, the GEC generally can locally store the measurement results and supply a measurement history. In particular the policy chosen is to store any new measurement in a database (DB) to allow remote access.

Each new day, all the data accumulated during the past day are processed to reduce the DB size. The operations performed by the GEC are the evaluation of maximum, minimum and average value. Next, these values are stored in another DB that keeps the history trend.

To develop the Gnutella Web Service we adopted LAMP: an acronym of the technologies that represent the open source Internet platform (Linux + Apache + MySQL + PHP | Perl | Python). As a multipurpose operating system, Linux is used for a wide variety of purposes including networking, software development, as well as an end-user platform. Apache is an implementation of an HTTP server and is the principal Web server in use today. MySQL is an implementation of a DB server that is known for its speed and reliability. PHP is a general-purpose scripting language that is particularly suited to Internet-based system development and is the most widely used Apache module. The implementation of GEC is reported in Fig. 13.

4.3. The Gnutella Web Service

In order to evaluate the feature of the proposed architecture, we implemented a monitoring application able to measure atmospheric values, developing a remote measurement system (GEC), a GWS and a web interface between it and the operator (Fig. 7).

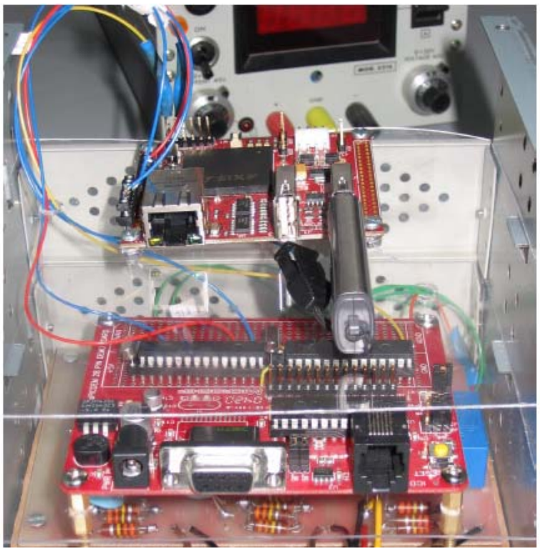


Fig. 13. GEC hardware implementation.

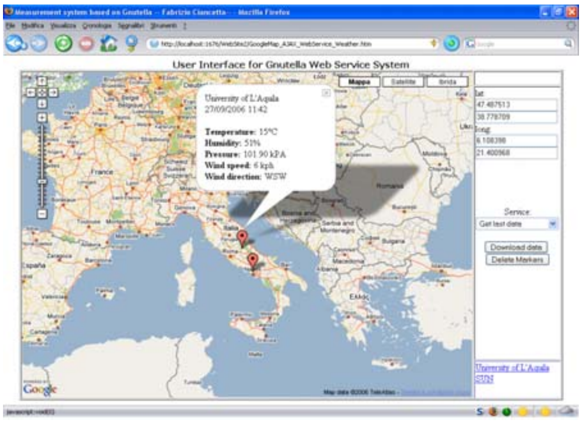
The GWS provides a particular implementation of typical Gnutella software, developing an ad-hoc Gnutella Search Engine. The results of the searching are processed by a PHP script for the responding of a SOAP request parsed by NuSOAP class. The GWS exports some methods to allow the user to access the Gnutella network. The methods are specifically developed for a measurement application; in particular the exported methods are:

- *GetStations*: to obtain information about the stations present in a limited geographic area defined by GPS coordinates, in order to restrict the searching. The output of the method gives an array of stations in which everyone reports the station name, the GPS position, the service list, other generic information and its HASHID.
- *GetCurrentData*: the user calls the method passing the HASHID of the remote station and the service request to obtain the current data.
- *GetHistoryData*: is similar to *GetCurrentData*, but is to access stored DB data.

The Gnutella network is time consuming during the searching. In order to reduce this time, we adopted a caching system: at the end of a search, the authenticated stations are cached and their IP address stored in a DB for a limited period. Therefore, to obtain some information from a particular station, it is not necessary to start a new search, but it is possible to directly perform the download.

4.4. The Web User Interface

The web user interface has been implemented as a XHTM page that sends a request to Web Service and displays the results using Google Map (Fig. 14).



University of L'Aquila
27/09/2006 11:42

Temperature: 15°C
Humidity: 51%
Pressure: 101.90 kPA
Wind speed: 6 kph
Wind direction: WSW

lat: 47.487513, 38.778709
long: 6.108398, 21.400968

Fig. 14. Screenshot of web user interface.

The web user interface gives a more degree of freedom of the whole system, allowing the user to access measurement information directly with a common browser. It uses Ajax (Asynchronous JavaScript and XML) technology to create interactive web applications.

The XHTML page sends asynchronous requests to the Web Service and installs a callback function on the XMLHttpRequest. All the management of the function is done in JavaScript. To interface the XHTML-JavaScript page with GWS, we adopted a SOAP client: a JavaScript class able to receive/create XML data form XHTML page and create/receive SOAP packet to GWS.

In particular, on the remote station we implemented the services: temperature, humidity, pressure, wind direction and speed as shown in the Google Map Balloon accessible directly on the map.

5. Results

In Fig. 7 we reported the structure of the Gnutella Web Service (the interface to Gnutella network) and in Fig. 14 the web user interface of the system. To provide a more powerful mode to represent data from Gnutella Embedded Client, we developed a Windows Form user interface. Based on Framework .NET 2.0, the user interface is divided in two parts: the first part, placed on the right side of the Windows Form, in which the user can: i) list the GECs present in the geographic area limited by the GPS coordinates; ii) select a station, looking at the available services and its GPS coordinates; iii) see a geographic view of all the station involved in the search.

On the left Windows Form side there are two panels, reporting the downloaded data. In the Current Data Panel (Fig. 15) we have a current view of the station with the last stored data acquired by the Gnutella Embedded Client and a graphical view of all the data of the current day from the 0:00 to the current hour retrieved from the GEC DB. In the History Data Panel (Fig. 16) we can perform a direct access to the Gnutella Embedded Client DB, downloading the data. In this example, we reported temperature and wind speed. All data are accessible directly to the GEC, without passing through Gnutella network to reduce the traffic. In order to reduce space there are two DB: one for the values accumulated during the day and another for an historical trend of the measurements.

6. Conclusions

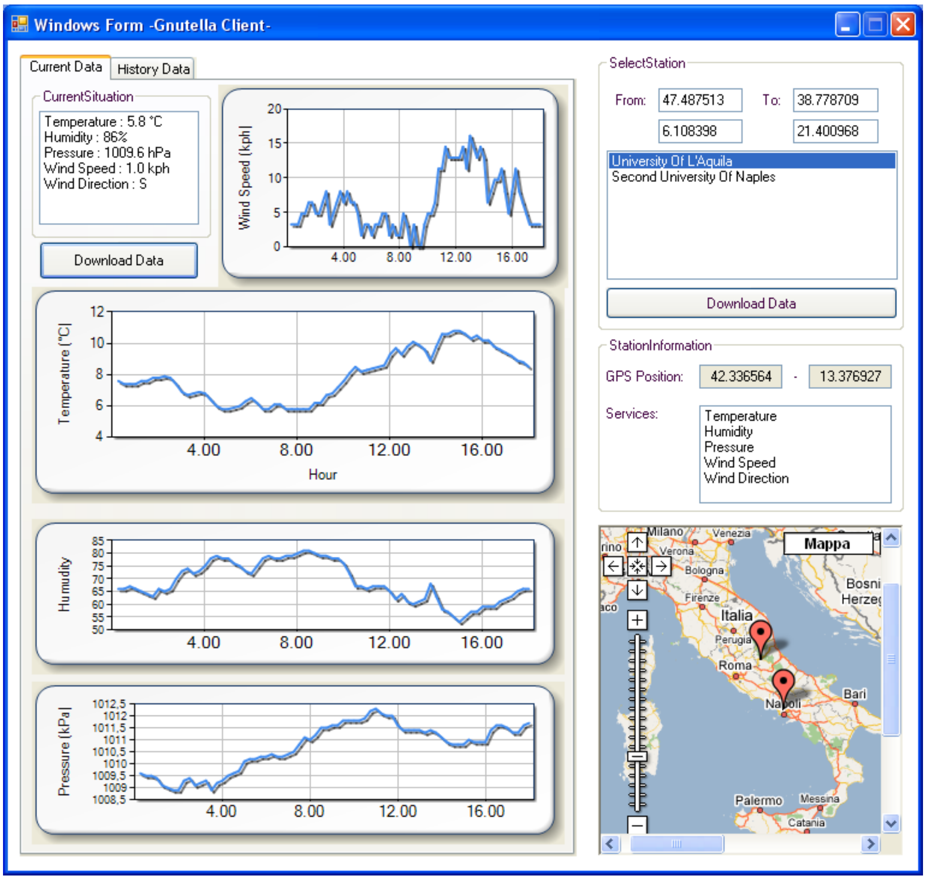
In this paper, a peer-to-peer distributed system for multipoint measurements is presented. The system uses the Gnutella network to route the information through the embedded client. These clients access to Gnutella network too, so they embody a micro kernel Gnutella to receive/send Gnutella packets. The distributed architecture adopts a Web Service to interface the Gnutella network with the users, so the adopted communication protocol (SOAP) is standard and allows a more degree of freedom of the whole system.

The particular application chosen to test the architecture is a weather monitoring station.

Two GECs have been developed with temperature, humidity, pressure, wind speed and direction sensors. One is placed at the University of L'Aquila and the other at the Second University of Naples. The GECs embody a local DB to store measurement trends. We developed a user interface to access the distributed architecture with a VB.NET Windows Form, using the Framework .NET 2.0.

The first obtained results show that the whole system is able to exchange information with GECs and to access the local DBs.

Further works will be oriented to measure the system performance such as the required searching time and the data transfer time.



Windows Form -Gnutella Client-

CurrentSituation

Temperature : 5.8 °C
Humidity : 86%
Pressure : 1009.6 hPa
Wind Speed : 1.0 kph
Wind Direction : S

SelectStation

From: 47.487513 To: 38.778709
6.108398 21.400968

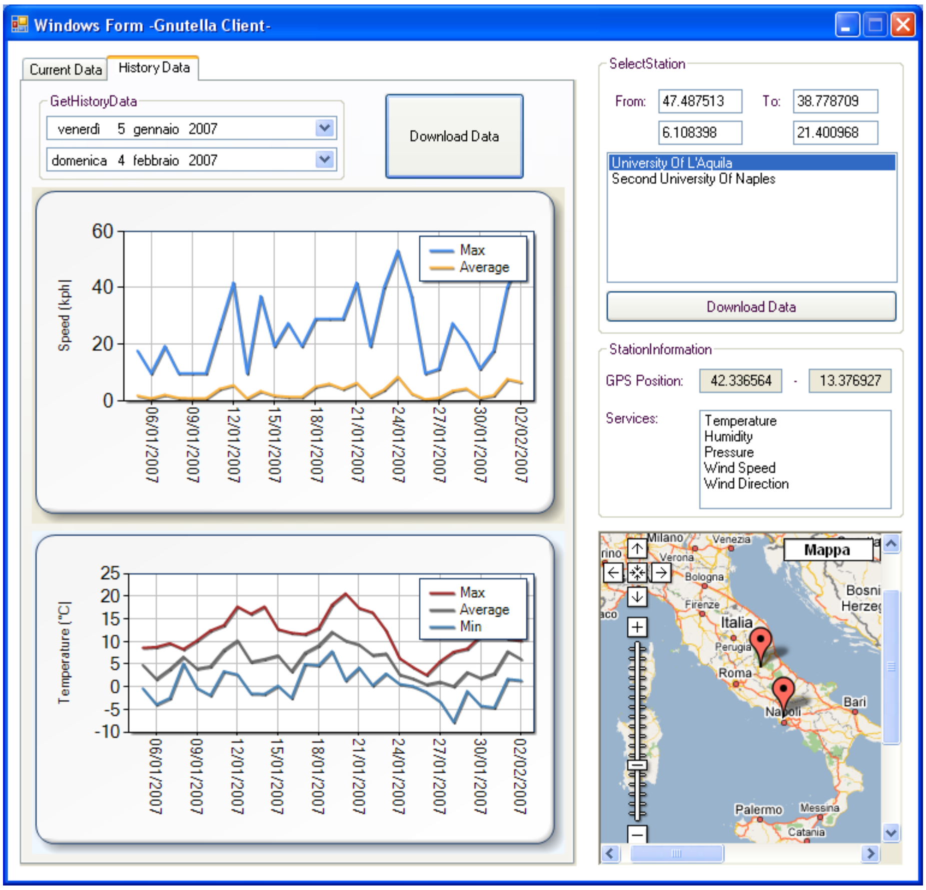
University Of L'Aquila
Second University Of Naples

StationInformation

GPS Position: 42.336564 - 13.376927

Services:
Temperature
Humidity
Pressure
Wind Speed
Wind Direction

Fig. 15. Windows Form Current Data Panel.



Windows Form -Gnutella Client-

GetHistoryData

venerdì 5 gennaio 2007
domenica 4 febbraio 2007

SelectStation

From: 47.487513 To: 38.778709
6.108398 21.400968

University Of L'Aquila
Second University Of Naples

StationInformation

GPS Position: 42.336564 - 13.376927

Services:
Temperature
Humidity
Pressure
Wind Speed
Wind Direction

Fig. 16. Windows Form History Data Panel.

References

[1]. F. Ciancetta, G. Bucci, B. D'Apice, C. Di Nucci, M. T. Todisco, A distributed measurement system based on Peer-To-Peer architecture for River Monitoring, in *Proceedings of the IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems Ostuni (VECIMS 2007)*, Italy, 25-27 June 2007.

[2]. F. Ciancetta, B. D'Apice, D. Gallo, C. Landi, Plug-n-Play Smart Sensor Based on Web Service, *IEEE Sensors Journal*, Vol. 7, Issue 5, May 2007, pp. 882-889.

[3]. J. E. Berkes, Decentralized Peer-to-Peer Network Architecture: Gnutella and Freenet, *University of Manitoba*, Winnipeg, Manitoba, Canada, April 2003.

[4]. F. Ciancetta, B. D'Apice, D. Gallo, C. Landi, Architecture for Distributed Monitoring based on Smart Sensor and Web Service, in *Proceedings of the Instrumentation and Measurement Technology Conference (IMTC'06)*, Sorrento, Italy, 24-27 April 2006.

[5]. Cristaldi, L., Ferrero, A., Muscas, C., Salicone, S., Tinarelli, R., The impact of Internet transmission on the uncertainty in the electric power quality estimation by means of a distributed measurement system, *IEEE Transactions on Instrumentation and Measurement*, Vol. 52, Issue 4, 2003, pp 1073–1078.

[6]. Matei Ripeanu, Peer-to-Peer Architecture Case Study: Gnutella Network Analysis, in *Proceedings of the 1st International Conference in Peer-to-Peer Networks*, Linkpings Universitet, Sweden, Aug. 2001.

[7]. http://en.wikipedia.org/wiki/Gnutella

[8]. G. Bucci, F. Ciancetta, E. Fiorucci, Daniele Gallo, Carmine Landi, A low cost embedded Web Services for measurements on power system, in *Proceedings of the IEEE VECIMS 2005*, Giardini Naxos, Italy, 18-20 July 2005.

[9]. F. Ciancetta, G. Bucci, C. Landi, Taxonomies of distributed measurement systems via UDDI registry, in *Proceedings of the Instrumentation and Measurement Technology Conference (IMTC'08)*, Victoria, British Columbia, Canada, 12-15 May, 2008, pp 1316- 1321.

[10]. IEEE Standard for a Smart Transducer Interface for Sensors and Actuators, *IEEE Std*, 1451.1-4, 1997, http://ieee1451.nist.gov/

[11]. http://rfc-gnutella.sourceforge.net/

[12]. http://tools.ietf.org/html/rfc1321

[13]. Gray, A. J. G., Sadler, J., Kit, O., Kyzirakos, K., Karpathiotakis, M., Calbimonte, J.-P., Page, K., García-Castro, R., Frazer, A., Galpin, I., Fernandes, A. A. A., Paton, N. W., Corcho, O., Koubarakis, M., Roure, D. D., Martinez, K., Gómez-Pérez, A., A Semantic Sensor Web for Environmental Decision, Support Applications, *Sensors*, 11, 2011, pp. 8855-8887.

[14]. Bröring, A., Maué, P., Janowicz, K., Nüst, D., Malewski, C. Semantically-Enabled Sensor Plug & Play for the Sensor Web, *Sensors*, 11, 2011, pp. 7568-7605.

[15]. Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S., Lemmens, R., New Generation Sensor Web Enablement, *Sensors*, 11, 2011, pp. 2652-2699.

[16]. Douglas, J., Usländer, T., Schimak, G., Esteban, J. F., Denzer, R. An Open Distributed Architecture for Sensor Networks for Risk Management, *Sensors*, 8, 2008, pp. 1755-1773.

2013 Copyright ©, International Frequency Sensor Association (IFSA). All rights reserved.
(http://www.sensorsportal.com)

Promoted by IFSA

MEMS for Cell Phones & Tablets Report up to 2017

Market dynamics, technical trends, key players, market forecasts for accelerometers, gyroscopes, magnetometers, combos, pressure sensors, microphones, BAW filters, duplexers, switches and variable capacitors, oscillators / resonators and micromirrors.

Order online:
http://www.sensorsportal.com/HTML/MEMS_for_Cell_Phones_and_Tablets.htm