

## Design an Embedded Web Server for Road Traffic Monitoring

Sahar Lazim Kadoory

College of Electronics Engineering, University of Mosul

### Abstract

This paper proposes a prototype system for road traffic monitoring purposes using Wireless sensor networks (WSN) which consists of relatively inexpensive sensor nodes capable of sensing and wireless communication. A Web server is embedded into WSN nodes to provide a flexible remote monitoring based on web browser. WSN nodes and Embedded Web Server (EWS) have been implemented using UBICOM IP2022 network processor. The reference design includes complete source code written in C-language. The web pages which are required for the web server have been developed using HTML together with CGI scripts by implementing them as C functions within the web server. A practical implementation of the system has been tested and its operation is validated through simulation using OPNET 11.5. The simulation results have declared that the wireless LAN IEEE.802.11b is very suitable for the system's environment and the utilization of EWS' CPU is 55%. This means that EWS can afford the network load without dropping. The results also indicates that the average throughput, delay, and response times are in the reasonable rang.

**Keywords:** Embedded Web Server (EWS), Wireless Sensor Network (WSN), Traffic Monitoring, Network Processor, UBICOM IP2022, CGI Program.

### تصميم خادم ويب مضمّن لمراقبة حركة النقل البري

سحر لازم قدوري

كلية هندسة الالكترونيات، جامعة الموصل

### الملخص

في هذا البحث تم اقتراح نظام لأغراض مراقبة حركة النقل البري باستخدام شبكات المتحسسات اللاسلكية (WSN) والتي تضم مجموعة من المتحسسات الرخيصة نسبيا القادرة على التحسس والاتصال اللاسلكي. حيث تم تضمين خادم الويب داخل احد المتحسسات اللاسلكية لاجل تزويده بالمراقبة عن بعد باستخدام متصفح الويب. المتحسس اللاسلكي وخادم الويب المضمّن تم تنفيذهم باستخدام معالج الشبكة (UBICOM IP2022) وان عملية التصميم باستخدام هذا المعالج تضمنت كتابة برنامج بلغة السي، اما صفحات الويب المستخدمة لخادم الويب فانها كتبت باستخدام لغة (الإنش تي إم إل) وبرامج آلية تنفيذ معالجة المعلومات (CGI) وهذه البرامج كتبت على شكل دوال بلغة السي داخل خادم الويب. تم فحص النظام المقترح عمليا والتأكد من مدى فعاليته بأسلوب المحاكاة باستخدام برنامج ال OPNET 11.5 . والنتائج العملية اظهرت كفاءة استخدام الشبكة اللاسلكية IEEE.802.11b وان استخدامية الخادم المضمّن كانت 55% وهي كافية لتحمل حمل الشبكة، وكما ان الانتاجية وزمن التأخير وزمن الاستجابة كانوا ضمن الحدود المقبولة.

## 1. Introduction:

Wireless sensor networks (WSN) are new integrative technologies arising from the improvement of wireless communication and small sensors. WSN is a kind of monitoring networks consisting of a large number of low-cost, power-saving, highly integrative and self-organized sensor nodes. WSNs own broad and valued application outlook including military, inner-city management, biomedical treatment, environmental monitoring and remote monitoring of dangerous areas [1].

In the past, there have been recent studies on cities that used good traffic monitoring system to solve the issues of traffic congestion among vehicles and roads. Congestion reduces efficiency of transportation structure and increases travel time, air pollution, and fuel consumption [2].

To solve these issues, a WSN is made to monitor the transport vehicles and design a traffic monitoring system based on a WSN that is applicable to all types of environment. It's ready to install over large-scale traffic areas and monitor roads in a much fine way. Where, WSN nodes can count vehicles and detect their speed, direction and types. WSNs are the combination of embedded system and wireless communication [1, 2, 3].

In the Last few years there has been a major demand for remote monitoring and control of embedded systems. The most appropriate way to do this is to connect the embedded system to the internet and allowing the use of aware Web browsers from any device based on web browsing. The most direct way to achieve this is to embed a Web server into embedded system [4, 5].

The embedded system can be employed to attend the embedded web documents, including static and dynamic data about embedded systems, to web browsers. This type of web server is called an Embedded Web Server (EWS). By using EWS, administrators can format and display the data of traffic monitoring with HTML pages and any standard browser [4, 6].

In this paper, a traffic monitoring system is proposed using WSN. The rest of the paper is organized as follow: Section 2 covers the related works of traffic monitoring system using WSN. Section 3 displays the overview of UBICOM platform. Section 4 consists of design of the proposed traffic monitoring system. Section 5 demonstrates the implementation of EWS and section 6 shows the testing and operation of it. Also, section 7 includes the simulation framework of the proposed traffic monitoring system. Finally, section 8 declares the conclusions of this paper.

## 2. Related Works:

Over the last decade, many WSN systems have been comprehensively established and studied for numerous applications, including environmental monitoring and traffic monitoring [3]. In road traffic monitoring applications, the following significant papers are reviewed in this section. In 2006, Cheung et al. [7] built a traffic surveillance technology system based on wireless sensors. Their system was deployed in freeways and at intersections for traffic measurements such as vehicle count, occupancy, speed, and vehicle classification which can't be obtained from standard inductive loops. The experiments showed that deploying WSN for traffic monitoring provides %99 of detection rate in real time. In 2007, Malik Tubaishat el at. [8] proposed a traffic control system using WSNs. The new decentralized system depended on the traffic information collected from the WSN to achieve a real time adaptive traffic control. The advantages of information collected from the wireless sensor network had two folds: 1) improve the localized flow model in the intersection and 2) improve the coordination among the neighbor traffic lights. In 2008, Ivan Corredor el at. [9] presented a new system of traffic

## Kadoory: Design an Embedded Web Server for Road Traffic Monitoring

control based on WSN. The advantages of this new technology applied to the traffic control allowed designing and developing systems with a high level of autonomy and intelligent. The capacities of these kinds of system to manage and get acknowledge about traffic were huge. In 2009, Manohar Bathula et al. [10] presented the design and implementation of a sensor network system for monitoring the flow of traffic through temporary construction work zones. Also, they implemented software architecture in TinyOS/nesC for collecting a variety of traffic statistics, such as flow, density, vehicle trajectories, etc. The design for a sensor network-based system for monitoring traffic was (a) inexpensive, (b) rapidly deployable, and (c) requires minimal maintenance. In 2011, V. Vanithaa et al. [11] proposed an algorithm for preprocessing the traffic monitoring system that provided set of services requests that were submitted simultaneously, like road traffic conditions, vehicle statistics, accident zone detection etc. The experimental results indicated that the proposed solution eliminated redundancies among similar requests efficiently and effectively thus reduced communication cost and increased network lifetime. In 2012, Brittany Crowell [2] explained the connection of a wireless sensor that was capable of estimating the density of traffic. The sensor was able to detect motion from infrared and then transmitted binary numbers as data. The wireless sensor that was created utilized Wi-Fi. Each part of the sensor had its specific role that had the ability to help detect car movement. The data was then transmitted to a webpage for further calculations.

### 3. UBICOM IP2022 Network Processor Overview:

Embedded systems are integrated hardware and software systems implementing dedicated functions, such as, electronic gadgets, communication devices, household appliances and medical instruments [4]. In this paper, embedded UBICOM IP2022 platform was adopted to implement Wireless Sensor Network and Embedded Web Server (EWS) for traffic monitoring purposes.

UBICOM IP2022 is a network processor produced by UBICOM Company; UBICOM provides the whole solution as a fully integrated platform - the Real Time Operating System (RTOS), the protocol stack, and the necessary hardware. UBICOM's IP2022 chip embeds some basic hardware, but it permits combining it with on-chip software to support the most prevalent protocols. The key to this approach is Software System on Chip (SOC) technology.

UBICOM's hardware includes the following components as shown in Fig. 1 [12].

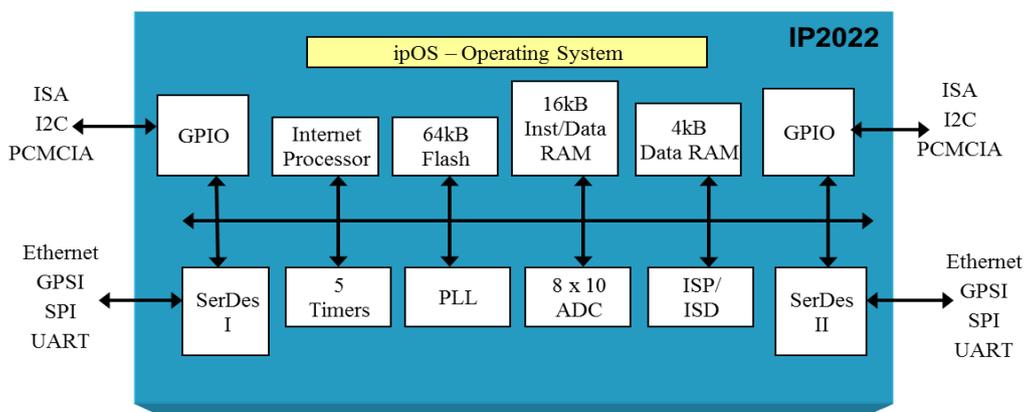


Fig. 1 UBICOM IP2022 Block Diagram

- 64KB (32K x 16) Flash program memory
- 16KB (8K x 16) SRAM data/program memory
- 4KB (4K x 8) SRAM data memory
- Two SerDes communication blocks supporting common PHYs (Ethernet, USB, UARTs, Bluetooth, Wireless IEEE 802.11, and so on.) and bridging applications
- 120MHz RISC processor
- Supports software implementation of traditional hardware functions
- In-system reprogrammable and run time self-programmable

The basic components for UBICOM's complete development environment can separate to three essential components [13]:

- 1) UBICOM's Software Development Kit (SDK): UBICOM supports many software packages, like ipOS™, ipStack™, ipHAL™, ipModule™ and etc.
- 2) Red Hat GNUPro tools which consist of GCC ANSI C compiler, Assembler, Linker, and GNU debugger.
- 3) UBICOM's Configuration Tool Integrated tool to support rapid development efforts.

UBICOM's Unity Integrated Development Environment (IDE) contains Editor, project manager, graphic user interface to GNU debugger, device programmer.

#### **4. System Design:**

To solve the issues of road traffic flow and to supply the driver of any car with the necessary knowledge about the status of the road, this system is proposed. It was implemented using WSN because it has low cost, small size, easy to install, consumes low power and Embedded Web Server (EWS) which provides the remote access. The components of the proposed system are as follow:

- Management server
- EWS at each road or intersection
- WSN into each vehicle
- PC or any device based on web browsing

The operation of the proposed system is explained as follows:

When, the vehicle entered to the specific road or intersection, this sensor will communicate with EWS of that road by sending a special message to inform EWS about its status, position, speed, etc. The communication between them is based on wireless communication protocol. However, EWS collects and stores the messages from each vehicle that flows nearby that road to analyze the messages and adds them to the current statistics of that road. The statistics are formatted and displayed as a dynamic web page to allow the client to review them on its web browser. Where, web browser has become the default standard user interface to a range of applications. It has become universal and it can run on many platforms right from enormous work stations to even almost hand phones. It allows client to access web server from any location to review the statistic of any road traffic. The communication between clients and web server is based on HTTP protocol. Afterword, EWS may be received configuration file from management server for reprogramming. Management server is a specific PC which is used to reprogramming EWS over wireless network by sending the configuration file with new web page, after the statistics information are received from it. Figure 2 exhibits the components of the proposed traffic monitoring system and their flows (more details about this system will be explained in section 7).

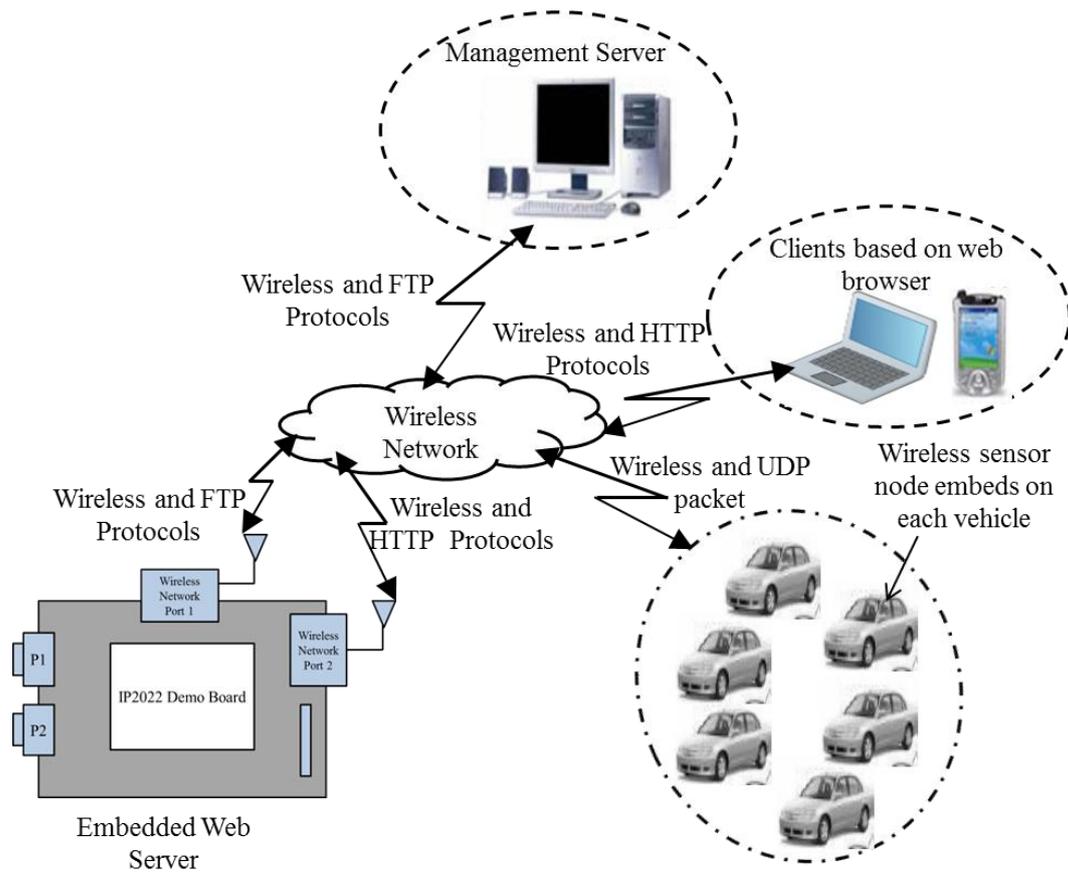


Fig. 2 The overall traffic monitoring system architecture in the remote access

### 5. Implementing the Embedded Web Server:

In the proposed system, the central function unit to get access on WSN that is located at any road traffic via Web browser is the embedded web server. Where, such EWS is used to bring the desired HTML and pictures over the worldwide Internet or a local network to the Web browser. In this section, the implementation of such EWS will be explained with the configuration details.

Before that, it will be revealed important feature about UBICOM platform. Where, UBICOM platform can be programmable over Internet or local network via wireless network port. This feature is significant when the administrator wants to reprogrammable EWS and adds new web page or knows the operation efficiency through control messages that are sent to the management server via the same port without needing to go to the location that EWS is placed in it.

The program in UBICOM's platform is called project, and the process constructing the project passes by two stages. The first stage adds the necessary packages for the project and configured them using the configuration tools and the output of this stage is a file extension (.lpj) holds inside it all the configuration's values of the program. The following packages are the SDK ipModules which are used when constructing the project:

- ❖ **ipOS:** manages resources -- the CPU, memory, and I/O of an IP2000 system and provides utility services (such as random number generation) for other modules.

- ❖ **ipStack:** provides an implementation of the Internet Protocol and the Internet Control Message Protocol. These are the two cores IPv4 networking protocols on top of which transport layer protocols such as TCP, UDP, and others may be built.
- ❖ **ipWeb:** provides an embedded Hypertext Transfer Protocol (HTTP) server for use with ipStack and provides an embedded web-server which uses the HTTP server layer.
- ❖ **ipFile:** provides a read-only flash file system. It provides application access to file data at system runtime, such as web pages to be served by a webserver.
- ❖ **ipWireless:** provides the wireless related functionality that is independent of any radio card and common to any project that wants to use 802.11 wireless.

The second stage is writing the program code within the editor of C language using Integrated Development Environment (IDE) for UBICOM platform, which is called Unity. Three basic files are created using C editor

- **main.c:** it contains the block configuration of IP2022 and initialization of the operating system, the stack for TCP/IP protocol, and our application such as HTTP protocol and DHCP client. Figure 3 shows the flowchart of the common operation of EWS.
- **TCP\_analysis.c:** it contains the following functions (the details about this file will be explained in section 5.1):
  - Functions for TCP/IP connection protocol.
  - Functions HTTP protocol connection.
  - A Function is used to call the static web pages that are stored in the internal flash memory.
  - A Function is used to create a Dynamic Web page based on the current statistical that are passed from main program.
- **UDP\_analysis.c:** it includes the function that is counting the messages that are received from the wireless sensors of the vehicle to find the number of vehicles nearby that road, compute the average of speed and find other information by analyzing the data in the messages (the details about this file will be explained in section 5.2).

After the construction of the project, the translation and debugging of the program are performed, and as a result of these two operations, the file extension is converted to the executive (elf) extension to be uploaded to UBICOM processor.

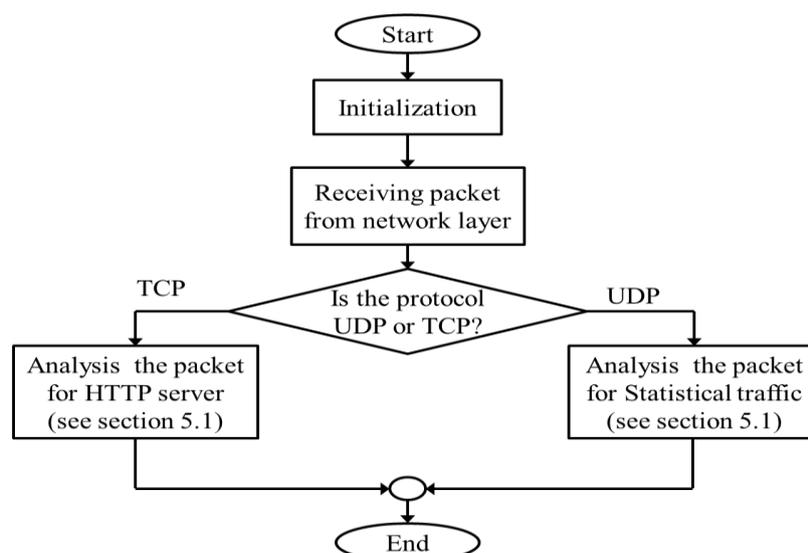


Fig. 3 flowchart of common operation of EWS

### 5.1 Web Page Design:

EWS should use the HTTP protocol to transmit Web pages to the device based on web browsing. HTTP server and client communicate via a TCP connection (the default TCP port value is 80). EWS works completely passive and it waits for a request of a client. The web page content is stored in local memory as static pages. In such case, any request from a client will always result in the client receiving the page as it is stored at the server. To make the web server dynamically updatable to new data that comes from WNS nodes, a Common Gateway Interface (CGI) program is used. This CGI program generates a web page of any current new data and passes that information into the stream of HTML data to be assembled as a web page by the remote client. Figure 4 shows the components of the embedded web page.

To protect the web server from any attacks or intruders or unshared persons who want to stop or destroy the web server or change the information and statistics on it. HTTP authentication is added on the embedded web server to prevent entering the web server by any unshared person or unauthorized person, username and password are setting for the authorized or administrator persons. HTTP Authentication can be enabled using the configuration tool of ipWeb package. A login name and password are required before any resources are sent.

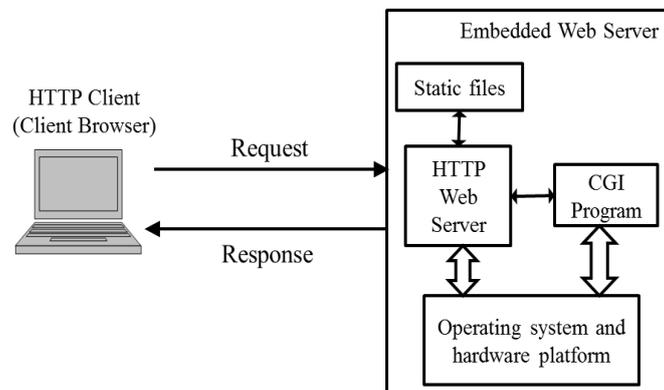


Fig.4 Embedded web page components

### 5.2 Statistical Analysis of Information from WSN of the Vehicles:

The second function of UBICOM platform is used to analyze the data packet received from WSN that embedded into each vehicle in order to pass this information to CGI program to update the web page of that road. This operation is performed as follows (as shown in Fig. 5):

- EWS can compute the number of vehicles entered that road by computing the number of received UDP packet (assumed the vehicle sends one packet when entered that road). To compute the number of vehicles at different resolution time (daily, hourly, and per minute), it must use a timer to specify the resolution time. Where, UBICOM platform is provided with special type of timer which is called oneshot timer. Oneshot timers are the primary mechanism for causing actions to take place after a specific time interval has elapsed. Each oneshot timer has the following key attributes:
  - Time duration before the timer expires.
  - A callback function which will be called when the timer expires.
  - A callback argument which will be passed to the callback function when it is called.
  - A flag that indicates whether the timer is free-running or synchronized to the system clock.

The default frequency of the periodic timer is 1000 Hz, but it is adjustable through the TICK\_RATE configuration parameter, where, the default value for TICK\_RATE is 1000 giving the timer a resolution of 1 millisecond. In order to set the time duration for one minute, the TICK\_RATE is multiplied by 60000, while the time duration for one hour is set through multiply the TICK\_RATE by 3600000, and the TICK\_RATE is multiplied by 86400000 to the time duration for one day.

- To compute the average of speed for that road, the same oneshot timers are used to compute the number of vehicles at one time and one hour. Where, the speed of each vehicle is sent as data in UDP packet.

Afterward, the statistics of day is assembled (the time duration of the oneshot timer for one day is expired) and sent as UDP packet (of size 104Bytes as shown in Fig.6) to the management server in order to configure new programming file with new web page. It is impossible to store all the traffic statistics to use them into CGI program because the memory of UBICOM platform is limited. The following statistics are passed to CGI program to update the web page on EWS:

- No. of vehicles within the last minute and hour is declared in the web page.
- Average speed within the last minute and hour is also pointed out in the web page.

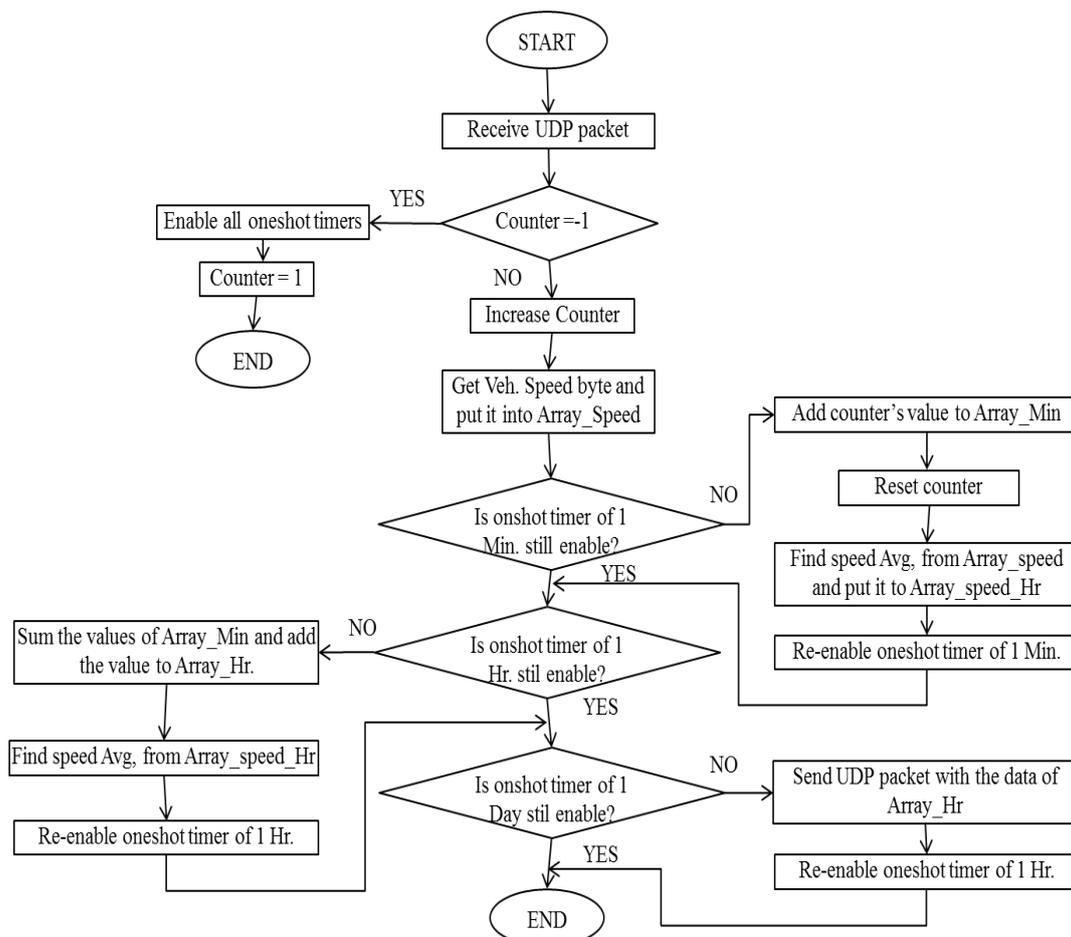


Fig. 5 flowchart of statistical analysis into EWS

## Kadoory: Design an Embedded Web Server for Road Traffic Monitoring

Wireless header (30Bytes)	Network header (20Bytes)	UDP header (6Bytes)	Veh.No. at Hr. 1 (2Bytes)	Veh.No. at Hr. 2 (2Bytes)	Veh. No. at Hr. 3 (2bytes)	.....	Veh.No. at Hr. 24 (2bytes)
------------------------------	-----------------------------	------------------------	------------------------------	------------------------------	-------------------------------	-------	-------------------------------

Fig.6 the format of UDP packet directed to management server

### 6. Experimental Testing:

Basically, the web server enables users to fetch web pages and display them on their own browsers in a platform-independent manner. Moreover, information coming from WSN nodes can be used to generate dynamic HTML documents by common gateway interface (CGI) scripts. Figure 7 shows the embedded web server, responding to the ping command made by the client by typing the IP address of the server in the client's browser. The user has to enter '10.1.1.26' IP address to access the server. This request is taken by the operating system of the client to its WLAN controller which sends the request to the network device that processes and checks for the system connected to the network with the particular IP address. Figure 8 shows the client entering the IP address of the server and the server send the html page to the requested clients. If the IP address entered is correct and matches to that of the server, a request is sent to the WLAN controller of the server and a session is established and a TCP/IP connection is establishes and the server starts sending the HTTP authentication web pages to the client (as shown in Fig. 9).

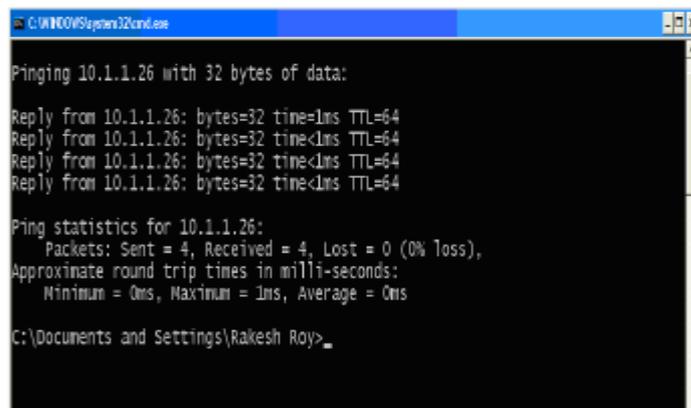


Fig. 7 response ping of the embedded web server over network

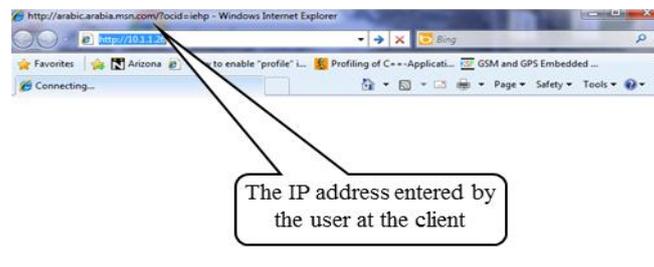


Fig. 8 user entering IP address of embedded web server

After the correct username and password is entered, the traffic statistics web page will be sent from EWS to the requested clients. Figure 10 displays the web page form of the proposed traffic monitoring system if it is located in one of the crowded streets. The proposed system provides rich information layout to their users. It can be noted that nearly real time traffic data

is assumed here for different time resolutions (yearly, monthly, weekly, daily, hourly and even per 1 minute).

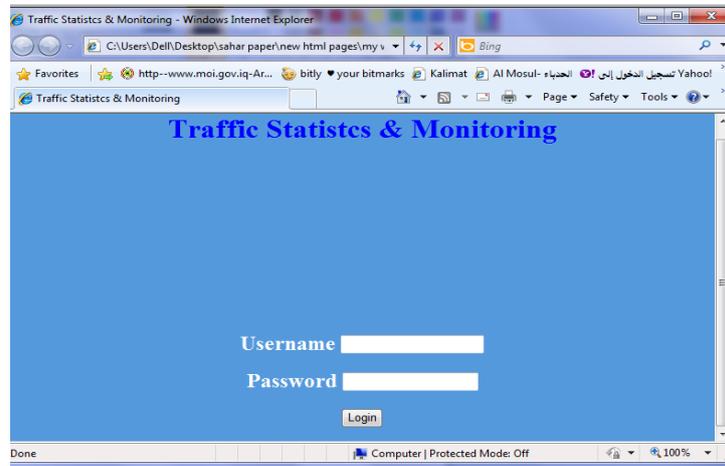


Fig.9 web page of the HTTP authentication

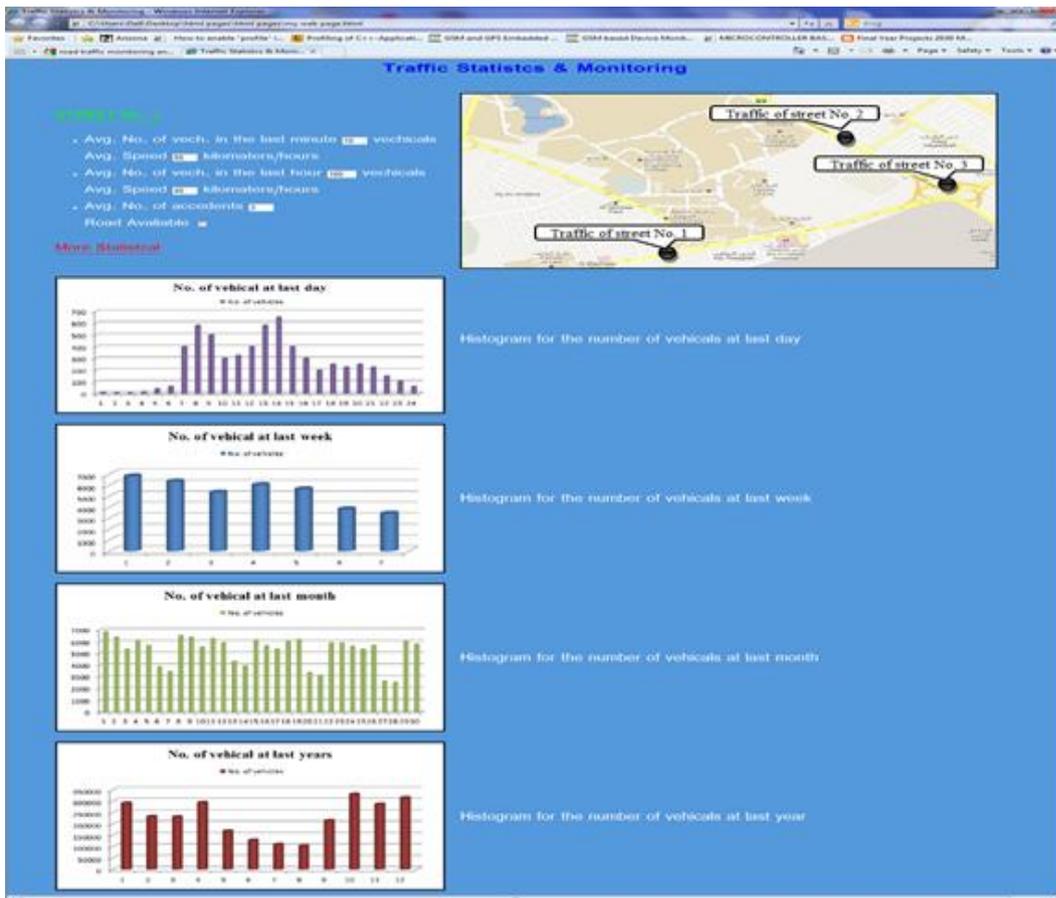


Fig.10 web page of the proposed traffic monitoring system

### 7. Simulation Framework:

The efficiency of the proposed system was evaluated through comprehensive simulations. The simulation analysis involves of different components which will be linked together using an IP network.

In this work, detailed simulation models of the system components are built using network simulation package OPNET11.5 which developed an efficient way to simulate field devices. The Wi-Fi networking model is built by creating a projects and work on the model at the OPNET's network layer.

#### 7.1 Simulation of the Proposed System:

The proposed system described in section 4 is simulated as shown in Fig.11. The simulation scenario consists of the following components:

- Two clients (client1 & client2) which are PCs or any device based on web browsing are located anywhere of the wireless network.
- One Embedded Web Server (EWeb\_Server) for a specific road: which collects and stores the messages from each vehicle to analyze and add them to the current statistic of that road.
- Ten vehicles equipped with WSN (average number of vehicles through one minute): which inform the embedded web server of that road about their status like speed by sending a special message.

The assumptions which are deliberated during the simulation are:

- The Wi-Fi network model is assumed to work with IEEE.802.11b standard with a maximum speed of 11Mbps.
- All the WSN nodes into each vehicle are fixed.
- Embedded web server and clients are also fixed (immobile).
- The dimension of the network is 300\*300 m which supposed to be realistic area for one roads of the city.
- The packet processing speed of EWS is 2000 packet/sec (which reflects CPU speed of  $\approx 120$  MHz that simulates the frequency of UBICOM network processor).

For well investigation, some applications are designed (or created) to examine the behavior of the proposed system. The flow actions of the applications are as follows:

- ❖ *Phase1 (Message Phase):* WSN Sensors into each vehicle send a UDP packet of size 64Bytes when they flow nearby a specific road to EWS of that road.
- ❖ *Phase2 (Webpage Phase):* If any driver or person wants to review the status of a specific road, he will request the webpage from EWS of that road. It will reply to the person or driver by sending an html web page as a file of 10 Kbytes using HTTP protocol (assume that the client request the web page at each 15 minutes).
- ❖ *Phase3 (Statistic Phase):* after the EWS assembled the traffic information for one day, it will send them as UDP packet of size 100Bytes to management server to used it in the new configuration file.
- ❖ *Phase4 (Programming Phase):* When the UDP packet at phase3 is received by management server, it uses this packet to configure the new web page. After that, the new configuration file is sent of size 50Kbytes using FTP protocol to reprogramming the EWS of that road.

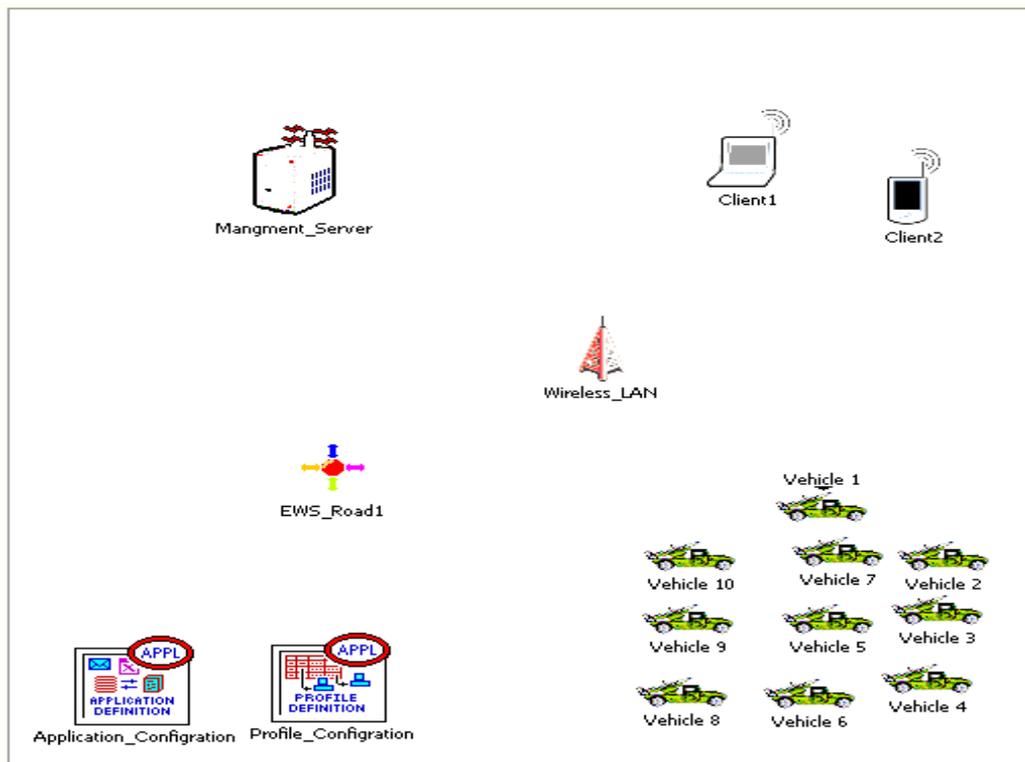


Fig.11.Simulation of the proposed System

**7.2 Simulation Results**

Depending on the assumptions mentioned in the previous subsection, the effect of non-real time data on the system is tested by assuming that two client requesting a webpage of 50Kbyte size from the EWS of that road using HTTP protocol. It will send a UDP packet for one day statistics management server, and receive a configuration file from it. So, the scenario simulation is running for one day and the results (as shown in Table 1 and Fig. 12) are declared and that the utilization of the EWS’s CPU is equal to 55%.

Table 1 important result for each phase

	Average Throughput (bit/sec)	Average Delay (sec)	Average Response Time (sec)
Phase 1	580	0.0027	---
Phase 2	As shown in Fig. 12	0.007	0.044
Phase 3	2250	0.0032	---
Phase 4	280	0.056	0.21

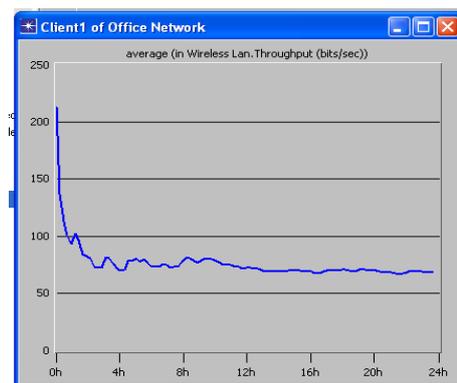


Fig. 12 average throughput of client requesting web page

The results are as expected for efficiency system, where, it can be noted the following points:

- Wireless LAN IEEE.802.11b with speed 11Mbps was very suitable for the system's environment.
- There is no dropping in any packet and no load in the system.
- The average response time for different applications was in the acceptable range.
- The average delay for the phases was relatively low, as, the average delay for overall system was equal to 0.0031 sec which is believable.
- The average throughput for the phases was good, as, the average throughput for overall system was equal to 8000 bits/sec this value was very respectable.
- The utilization of EWS' CPU appeared that EWS can afford the network load without dropping.

### 8. Conclusions:

This paper proposes a traffic monitoring system using Wireless sensor networks which are composed of tens or even hundreds of small, cheap sensor nodes which communicate with one another wirelessly. Web server is embedded into WSN nodes to convey the state information of WSN nodes. WSN nodes and the embedded web server are employed using UBICOM IP2022 which proves its high efficiency through employment and operation each of them. The software implementation of the embedded web server is written using C language. The web pages which are required for the web server were developed using HTML. This embedded web server is tested for its working, using a PC hosted over a network of PC's. Moreover, Utilization of dynamic monitoring web based on CGI scripts improves the response capability and brings convenience for complex monitoring web design. A practical implementation of the system was tested and its operation is validated through the simulation scenario using OPNET 11.5. The simulation results were declared that the wireless LAN IEEE.802.11b was very suitable for the system's environment, the utilization of EWS' CPU is 55% this means that EWS can afford the network load without dropping, and the average throughput, delay, and response times for different applications were in the reasonable rang.

### 9. References:

- [1]. Wen Y., Pan J., Le J., "Survey on Application of Wireless Sensor Networks for Traffic Monitoring", International Conference on Transportation Engineering (ICTE), pp. 2079–2084, 2007.
- [2]. Brittany C., "Wireless Sensor Utilizing Wi-Fi that is Capable of Estimating the Density of Traffic", Research paper, Department of Computer Science Jackson State University, 2012.
- [3]. Vongsagon B., Jurarat E., Kulwadee B., Somsak K., "xbee Wireless Sensor Networks for Temperature Monitoring", Research paper, Sirindhorn International Institute of Technology, 2011.
- [4]. Muhammad Sh., "Embedded Web Server", thesis of the Master Degree of Electrical Engineering, Electrical & Electronic Engineering department, Kolej University Technology TW1 Hussein Onn, October, 2004.
- [5]. Hong-Taek J., Mi-Joung Ch., James W., "An efficient and lightweight embedded Web server for Web-based network element management", International Journal of Network Management, pp. 261 – 275, Oct 2000.

- [6]. V. Billy, Sanket D., S. G. Shiva, “Design and Development of ARM Processor Based Web Server”, Poster paper, International Journal of Recent Trends in Engineering, Vol. 1, No. 4, pp. 94 – 98, May, 2009.
- [7]. Sing Y., Sinem C., Baris D., Sumitra G., Chin-Woo T., Pravin V., “Traffic measurement and vehicle classification with a single magnetic sensor”, Journal of the Transportation Research Board, February, 2006.
- [8]. Malik T., Yi Sh., Hongchi Sh., “Adaptive Traffic Light Control with Wireless Sensor Networks”, Research paper, Indian institute of science, IEEE, pp. 187 –191, 2007.
- [9]. Iván C., Ana-B G., José-F M., Pedro L., “Wireless Sensor Network-based system for measuring and monitoring road traffic”, Research paper, collector Iberoamérica, 2008.
- [10]. Manohar B., Mehrdad R., Ishu P., Nilesh P., Joe G., Nigamanth S., “A Sensor Network System for Measuring Traffic in Short-Term Construction Work Zones”, Research paper, Springer-Verlag Berlin Heidelberg, pp. 216 –230, 2009.
- [11]. V. Vanithaa, V. Palanisamy, K. Baskaran, “Service Merging for Service Oriented Wireless Sensor Networks”, European Journal of Scientific Research, Vol.50 No.2, pp. 263-269, 2011.
- [12]. “IP2022 Wireless Network Processor Features and Performance Optimized for Network Connectivity IP2022 Data Sheet”, UBICOM, Inc., 22 Jan. 2009, Web Site: <http://www.ubicom.com>.
- [13]. “IP2022 Internet Processor Product Brief”, UBICOM, Inc., 2001, Web Site: <http://www.ubicom.com>.