Toward Ubiquitous Networking: QoS-aware Residential Gateway with Embedded ZigBee-based Network

Pei-Chen Tseng1, Rung-Shiang Cheng2, Yi-Cheng Chang1, Wen-Shyang Hwang3

1 Department of Information Engineering and Informatics, Tzu Chi College of Technology
Hualien 970, Taiwan
{peichen, s9743028}@tccn.edu.tw

2 Department of Computer and Communication, Kun Shan University
Tainan 710, Taiwan
rscheng@mail.ksu.edu.tw

3 Department of Electrical Engineering, National Kaohsiung University of Applied Sciences
Kaohsiung 807, Taiwan
wshwang@mail.ee.kuas.edu.tw

Received 6 May 2012; Revised 27 May 2012; Accepted 30 May 2012

Abstract. Society is trending toward ubiquitous networking. This study thus presents a prototype but fully functional system which, in theory, could be expanded worldwide, expediting development of the Internet of Things. Our earlier QoS-aware residential gateway (EmQRG) for real-time class-based queuing bandwidth management is reviewed and experimentally demonstrated in a novel embodiment which includes a FDIXP425-DevPlatform integrating the EmQRG with a wireless M2M ZigBee-based temperature/humidity monitoring network (FT-6250 + FT-6251’s), which is treated experimentally as a fire alarm system. When any temperature/humidity module exceeds a preset value, the composite system activates a warning light bulb and sends warning messages to designated recipients. This emergency signal has top EmQRG transmission priority. Tests in the context of streaming video and simulated background internet traffic under light to heavy network congestion and bottlenecking show consistently good QoS for both the alarm and the streaming media. The emergency alerts are received immediately under all conditions. The warning light bulb turns off when temperature falls below the threshold value. Discussion shows that the EmQRG network can contain other embedded EmQRG networks and be embedded within higher EmQRG-based networks. The presented system is cost-effective, easy-to-use, easy-to-implement and completely implementable with available hardware and software.

Keywords: embedded system, QoS, M2M, IoT, home network, wireless sensor network

1 Introduction

In pre-technological times, the future was viewed as a recurrence of a standard pattern, i.e. a phoenix-type scenario where death is followed by rebirth to the same form. History was a flat line with little societal change. Contrastingly, the recent several centuries have seen such rapid scientific/technological progress that a new type of fiction has been created, namely science fiction. For example, the 19th century author Jules Verne launched a spaceship to the moon by use of a large cannon (“From the Earth to the Moon,” 1865). Science fiction involves imagining the future from an evolving technological viewpoint. It has become one of the most popular forms of contemporary media.

Modern-day science fiction commonly envisions a future in which the everyday environment is ubiquitously networked. In such a future, wired and wireless networking and smart wireless-capable devices are embedded in most common devices and sub-components in complex machine-to-machine (M2M) networks similar to the anticipated Internet of Things (IoT) [1]. As such, it foresees that Fig. 1’s wireless networking of dissimilar and composite networks becomes increasingly ubiquitous. Internet-capable uniquely identifiable objects (things) and their virtual representations interact more and more, gradually forming an IoT that may one day include every object on the planet [2]. Such wireless sensing devices are networked to form complicated systems of systems (Fig. 2) [3], [4]. Such systems need to operate for many years, reliably performing their functions in the context of real world communication. Among their primary functions are to monitor and report the current status, location, general performance and failure of the mechanisms or sub-components to which they are affixed or in which they are embedded, e.g. nuclear power stations, airplanes, automobiles, air conditioners, cellphones, jog-
Tseng et al.: Toward Ubiquitous Networking: QoS-aware Residential Gateway with Embedded ZigBee-based Network

ging shoes, fish bowls and even the human body. Many of the systems are unattended for large periods of time, often performing very important tasks. However, the ubiquitously networked future has not yet arrived. At present, it is under development. The urgency of this critical mission is exacerbating our present need for high confidence wireless sensing networks (WSN) with high inter-compatibility and high ease-of-use.

Fig. 1. Wireless network technology

Table 1. Comparison in wireless network technology

<table>
<thead>
<tr>
<th></th>
<th>ZigBee</th>
<th>Active RFID</th>
<th>Bluetooth</th>
<th>Wi-Fi</th>
<th>Ultra Wide Band</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target Market</strong></td>
<td>Monitoring and control, sensor network</td>
<td>Object identification and management</td>
<td>Cable replacement</td>
<td>Internet application</td>
<td>Short range high data rate application</td>
</tr>
<tr>
<td><strong>System Resource Requirement</strong></td>
<td>Low</td>
<td>Proprietary</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>$3&lt;</td>
<td>$4+</td>
<td>$5</td>
<td>$6-$10</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Typical current absorbed</strong></td>
<td>18 mA&lt;</td>
<td>5 mA&lt;</td>
<td>30 mA&lt;</td>
<td>100-350 mA</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Max. Bandwidth</strong></td>
<td>250 Kbps</td>
<td>N/A</td>
<td>2.1 Mbps</td>
<td>Up to 54 Mbps</td>
<td>100 Mbps +</td>
</tr>
<tr>
<td><strong>Nodes/ Master</strong></td>
<td>64000</td>
<td>Mesh</td>
<td>7</td>
<td>32</td>
<td>Mesh</td>
</tr>
<tr>
<td><strong>Nominal Transmission range</strong></td>
<td>10-100 m</td>
<td>5-100 m</td>
<td>10 m</td>
<td>100 m</td>
<td>10 m</td>
</tr>
<tr>
<td><strong>Application Focus</strong></td>
<td>Remote control, battery-operated products, sensors</td>
<td>Electronic card, object tracking, presence monitoring</td>
<td>Wireless USB, PDAs, handsets, headsets</td>
<td>High speed wireless Ethernet access</td>
<td>Real-time multimedia data transmitting</td>
</tr>
</tbody>
</table>
WSNs are different than traditional wireless or wired networks because of sensor and actuator based interaction with the environment. Such networks are of interest for various applications, e.g. search and rescue, disaster relief, surveillance, target tracking and smart environments. In sensor networks, a node’s location is an important part of the node’s state. Node location is used [5], [6], [7], [8]: (i) to identify the physical space from which sensor readings originate, e.g. identifying target position during tracking, locating survivors under rubble, locate and coordinate smart vehicles; (ii) in geographically-based (as opposed to ID-based) communication protocols; (iii) in other location-based services, e.g. in smart hospitals to monitor current patient body status and location and to provide immediate patient records to medical personnel. Continued WSN research will generate very sophisticated location-aware devices and services. At present, many localization algorithms have been proposed to provide per-node location information. Presently, there are two general categories for proposed location mechanisms, one being the range-based approaches, the other being the range-free approaches. The range-based approaches use point-to-point distance estimates (range) or angle estimates to calculate location. The latter disregards range data when calculating location. Because the range-based approaches require additional hardware, the range-free approaches are considered more cost-effective.

One of the wireless network systems under current development is ZigBee-based. ZigBee is a home-area network designed to replace the proliferation of individual remote controls. ZigBee was created to provide a cost-effective, standards-based wireless network that supports low data rates, low power consumption, security and reliability. Accordingly, the ZigBee Alliance is developing standardized application software on top of the IEEE 802.15.4 wireless standard [3], [9]. The ZigBee Alliance is working with IEEE to ensure an integrated, complete and interoperable network. For example, the Zigbee Alliance will provide interoperability certification testing of 802.15.4 systems that include the ZigBee software layer. At present, ZigBee is the only standards-based technology that addresses the needs of most remote monitoring, control and sensory network applications. Expected applications of network-type ZigBee-based systems such as the RCM 3700 network system [10] focus largely on M2M applications, including consumer electronics, energy management systems, security systems, inventory management systems, body-monitoring health care systems, home automation systems, automated buildings, industrial automation systems, etc. Interest is expected grow rapidly as the devices become cheaper and more sophisticated.

Many of the functions performed by a Zigbee network will require extensive modular replication. Examples include measuring temperatures over large areas, vibration monitoring in large structures, monitoring the progressive location of a moving object, storing variable amounts of information from a variable number of static and/or moving monitors, integrating and controlling/monitoring large numbers of diverse devices in a smart building or an ocean-going vessel, etc. Adequate system QoS will depend significantly on the degree of geographic replication. This is somewhat analogous to locally-acting organisms (e.g. bacteria, bees or humans) cooperating to achieve tangible effects on a large scale.

Human-to-human (H2H), human-to-machine (H2M) and M2M composite systems are increasingly common [1]. Whether wireless or hardwired, whether ad hoc, LAN or Internet, such services are among the forefront of development. Market interest is intense. The great variety of current and pending applications, platforms and connection modalities is driving research intended to solve interoperability issues, which are among the major
concerns in any composite system. Interoperability issues are increasing rapidly in importance with the daily increase in the variety of available network-capable devices/systems which can in turn be embedded within larger systems, for example a closed-community control system which combines security camera systems, gate control, license plate recognition, biometric-based entrance controls, lawn care systems, pet monitoring systems, fire monitors, solar collector systems, etc. It is difficult to design efficient and flexible architecture for systems which exchange data between significantly different sub-systems.

Whereas earlier technological generations were content with wireless text-level data transmission speeds, multimedia and other large-volume high-speed data sharing applications are becoming standard. A wide variety of light-weight low-power microprocessor-embedded network-capable devices are evolving rapidly. Users of multiple sophisticated and diverse wireless modalities now spend most of their time in increasingly information-networked environments such as smart homes, offices, schools, shopping malls, highways, cellphone-service zones and WiFi-service zones. At present, however, communication standards, software languages and connection specifications remain diverse. Nevertheless, high QoS remains a critical requirement of such systems, an issue which presents daunting challenges to designers.

Our prior work presented an embedded QoS-aware residential gateway (EmQRG) [11] which focused on the home network. It offered a basic Internet-LAN gateway and server for the home which could coordinate and optimize data traffic flow over a network with bandwidth-limited resources. The system optimized traffic by identifying and classifying forwarded traffic according to priorities which could be set by the user. Data-intensive streaming such as video and online gaming were given high priority so that good and continued QoS could be maintained. Less critical traffic such as email and ftp transfer were given lower priority. Top priority was given to critical signals such as fire alarms, security alerts and remote sensor/system links which required fastest transmission times for proper function.

In general, we anticipate that ZigBee systems will be peripheral to or embedded in PC-based systems and will tend to be used for M2M functions. Thus, this current study extends our earlier EmQRG system to integrate with M2M-type wireless ZigBee systems [12]. First we review and demonstrate experimentally EmQRG for real-time class-based queuing (CBQ) bandwidth management in a DiffServ-capable CBQ-capable network. Second, we demonstrate and verify EmQRG’s ability to deliver high QoS in conjunction with a ZigBee M2M system. This is done experimentally using the EmQRG to integrate a ZigBee-based temperature/humidity monitoring system (FT-6250 + FT-6251’s) with ongoing but varying levels of streaming multimedia demand. Third, we discuss the issues involved in expanding this prototype system to larger and more complex environments.

This rest of this paper is organized as follows. An overview of related work is presented in section 2. Section 3 presents the DiffServ-QoS mechanism and the application structure of the FT-6250/FT-6251 ZigBee network. Implementation and experimental results are in sections 4 and 5. Discussion is presented in section 6. Conclusions and future work are summarized in section 7.

2 Related Work

Much literature has been dedicated to wireless technologies for short- and medium-range communications between electronic devices [5], [6], [13], [14], [15], [16], [17], [18]. Considerable further work has been dedicated to developing their ability of these technologies to support multiple applications and improving their cost efficiency [5], [14], [15]. Such technologies are becoming significantly synonymous with home, manufacturing automation and very-large-scale smart systems.

ZigBee is playing a significant role, already covering many applications for home, military and generic WSNs. ZigBee is based on the IEEE802.15.4 standard, which supports physical and MAC layers as well as approved ZigBee Alliance network and application layers. Being a symbiosis of two organizational standards, ZigBee devices are being built on open platforms. Many companies are utilizing open software and hardware to manufacture ZigBee platforms [10], [14], [18]. This technology attracts users by its simplicity, cost efficiency and low power consumption, adding important impetus to the trend of “smartinization.”

ZigBee devices are a good choice for many applications. They can work years on the same battery, support transmission rates of 20 Kbps to 250 Kbps, and are very economical for indoors use over distances ranging from 10 to 100 meters [19], [20], [21]. The majority of modern WSNs are built in frequency bands where ZigBee is standardized. ZigBee-based WSNs are becoming common elements in instrumentation and other industries. Accordingly, the WSN market and the ZigBee-based sub-market are developing rapidly due to high demand in industry, with the special boost of high volume deployment by the military [6], [16]. Nevertheless, as seen in Fig. 1 and Table 1 [4], [18], it should be recalled that ZigBee is competitive with Bluetooth, Ultra Wideband and 802.11 techniques. Although none of competing technologies so far has demonstrated a unique superiority, an increasing need for intercompatibility is verified by the fact that vendors frequently try to build chipsets that support several competing technologies.
In keeping with this issue, our EmQRG system is designed to integrate, coordinate and optimize data traffic between a wide variety of technologies and systems (Fig. 3). EmQRG, as will be detailed in the next section, was designed in its basic form to serve as a smart-home control center which can integrate safety/security tasks, coordinate information sharing of digital content, translate communication protocols among the various devices, work as a gateway to external networks, and also perform the DiffServ-QoS traffic classification protocol detailed in the next section.

![Fig. 3. Network architecture of EmQRG-embedded ZigBee home network](image)

This paper adds a FDIXP425-Dev Platform [22] to our existing system so as to demonstrate and verify intercompatibility with a M2M multi-sensor ZigBee-based wireless sensor network. The FDIXP425-Dev Platform supports many network and communication applications, e.g. the embedded network video server machine development platform, the embedded network development platform, the embedded communication development platform, the 802.11a/b/g application development platform, the SOHO router, internet gateways, WLAN AP, network fireproof wall development, VoIP, etc. The manufacture claims that using this platform allows customers to easily finish development tasks.

Wireless communication in the following is accomplished according to the X-10 standard [17], which is an international and open industry standard for communication among electronic devices. Availability and simplicity have made X-10 the best-known home automation standard. It enables plug-and-play operation with any home appliance and requires no special knowledge to configure and operate a network for control of various devices. It was designed primarily for home automation, using power line wiring for signaling and control, where the signals involve brief radio frequency bursts representing digital information. The X-10 PRO code format is the de facto standard for power line carrier transmission. X-10 transmissions are synchronized to the zero-crossing point of the AC power line. X-10 commands enable control of the status of a device, for example controlling the status of a lamp unit (on, off, dim, bright).

In the following, a number of FT-6251 High Power ZigBee Sensor Boards [4] are used to transmit temperature and humidity data over the ZigBee-based WSN network to a central coordinator (ZigBee gateway), i.e. a FT-6250 High Power ZigBee Base Board. A star network topology is formed as seen in Fig. 4, wherein all the end devices communicate only with the coordinator. The FT-6250 coordinator communicates via an RS232C link with IXP425 embedded system which is connected to the FDIXP425-Dev Platform embedded in the EmQRG system.

The coordinator starts the network for the other sensor boards to join. The coordinator receives sensor data from various sensor boards and outputs the readings and the corresponded network addresses to the UART. The coordinator sends the control data to the end device via indirect transmission. The end device periodically takes readings from the modular temperature and humidity sensors, then transmits the data to the coordinator. During consecutive data transmission, the end device steps into sleep mode to conserve energy. The end devices send the data to the coordinator via direct transmission. The coordinator saves network information in a flash memory and retrieves it to maintain an established network after power failure or reboot. The resulting ZigBee system is capable of:

- 65,536 network (client) nodes.
- Optimized for timing-critical applications and power management
  - Time to Join Network: <30ms
  - Sleeping to active: <15ms
3 DiffServ-QoS mechanism and application structure of the FT-6250/FT-6251 ZigBee network

3.1 DiffServ-QoS Mechanism

The DiffServ-QoS service architecture is based on a methodology whereby the EmQRG classifies forwarded traffic, marking each data packet for special treatment according to preset (user adjustable) behavior aggregates. Within a DiffServ-capable network, packets are forwarded according to the per-hop behavior (PHB) associated with the DiffServ codepoint contained in the packet header. This mechanism utilizes a special type of RAW socket called divert (IPPROTO_DIVERT) that allows receiving and sending just like regular sockets. The difference is that a divert socket is bound to a specific system port, so the IP firewall can be instructed to send only certain packets, i.e. the IP firewall can filter out packets for the divert socket. The IP firewall is used to implement the so-called DSME (DiffServ Marker Engine) (Fig. 5) within the EmQRG system. Under Linux kernel version 2.4.18_eixp425, iptables are developed to support the IP firewall. DSME is the mechanism which is core to the DiffServ-QoS abilities. The following details the DSME packet processing scenario for traffic forwarded via the IP Firewall in Fig. 5 [23]:

1. Traffic (packet) forwarding through the EmQRG is intercepted by the IP firewall.
2. DSME requests the IP firewall to redirect matched packets (i.e. IP packets that have the same destination address and the same port number) to the specific system port in conformance with the firewall rules.
3. Once the packet is redirected to the specific system port, it does not travel directly forward.
4. DSME adopts divert socket to bind the specific port, then reads raw packets (the whole IP packet including the IP header). The TOS (type of service) fields of the IP header of raw packets are modified by the DiffServ-TOS conversion values in DSME. The modified packets are sent back to divert socket.
5. The modified packets are sent back to the firewall.
6. The modified packets are sent to the appropriate network application for further processing if needed. Otherwise, they go directly to step 8.
7. The modified packets are sent to the firewall again.
8. Finally, the modified packets continue to travel forward.

EmQRG classifies forward traffic flow into classes such as safety traffic, multimedia traffic, ICMP traffic, file transfer traffic, web traffic, interactive traffic and BE (best effort for uncategorized traffic) traffic (traffic types listed in Table 2). The EmQRG then assigns each class of traffic a preset DiffServ-QoS value. This is particularly important in busy multi-service composite systems. For example, email traffic and FTP file transfer do not require immediate and prompt delivery for functionality, and can thus be assigned a low priority. Streaming...
video and online video games require prompt and timely delivery to maintain reasonable customer satisfaction. These applications are assigned higher priority. Emergency medical alarms and similar services demand the fastest response and are given the highest priority. As mentioned, these priorities are adjustable according to the demands of the user/administrator.

![Packet processing scenario by DSME via IP firewall](image)

**Fig. 5.** Packet processing scenario by DSME via IP firewall

<table>
<thead>
<tr>
<th>Traffic class</th>
<th>Protocol categories</th>
<th>Divert port</th>
<th>DiffServ setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety traffic (HTTP:8077)</td>
<td>TCP:8077</td>
<td>9977</td>
<td>EF</td>
</tr>
<tr>
<td>Web traffic (HTTP:80)</td>
<td>TCP:80</td>
<td>9980</td>
<td>AF22</td>
</tr>
<tr>
<td>File transfer traffic (FTP)</td>
<td>TCP:21, TCP:20</td>
<td>9921, 9920</td>
<td>AF31</td>
</tr>
<tr>
<td>Interactive traffic (SSH, TELNET)</td>
<td>TCP:22, TCP:25</td>
<td>9922, 9925</td>
<td>AF21</td>
</tr>
<tr>
<td>Multimedia traffic (UDP)</td>
<td>UDP</td>
<td>9990</td>
<td>EF</td>
</tr>
<tr>
<td>Ping (ICMP)</td>
<td>ICMP</td>
<td>9910</td>
<td>BE</td>
</tr>
</tbody>
</table>

**Table 2. Traffic classification**

3.2 Application Structure in FT-6250/ FT-6251 ZigBee Sensor Network

The function AppColdStart(), which acts as the entry point of the application, is called by the boot loader every time after there is a reboot or a wake up from a sleep mode without memory hold. The AppColdStart() first calls the function vWUART_Init() to perform the following tasks [4]:

- Initialize stack and hardware interface. Application Queue API is used, so only the upward queues are monitored.
- Initialize the system state and flags.
- Set PAN ID and short address of the coordinator.
The coordinator operates in any of these system states:

```c
/* System states */
typedef enum
{
  E_STATE_INIT,
  E_STATE_START_ENERGY_SCAN,
  E_STATE_ENERGY_SCANNING,
  E_STATE_START_COORDINATOR,
  E_STATE_RUNNING_UART_APP
} teState;
```

The first four states create a network (Fig. 6). Then, the coordinator stays at E_STATE_RUNNING_UART_APP for normal operation. End devices can join the network by performing Active Scan.

Direct transmission is used to transfer data from end devices to the coordinator, which is “power on” at all times. MAC layer Acknowledgement ensures successful data reception. The end device goes into sleep mode after successful data transmission. If the end device assumes network connection is lost, it reboots to attempt to rejoin the network. Indirect transmission is adopted to transfer data from coordinator to end devices, which poll after every wake-up to check if data is pending from the coordinator.

The coordinator saves network information to flash memory, so the network is maintained after coordinator reboot or power failure. Major network information sCoordData are stored in flash memory by calling vSaveContext(). sCoordData are restored by calling vRestoreContext(), which reads data out of flash memory and saves network information into the MAC hardware register by calling vStartCoordinator().
In the application, a short press of the Button 1 (Fig. 7) can save the network information. A long press of Button 1 (until the toggle of LED 0) will restore the network information. During network operation, both LEDs on coordinator and end devices are ON in the initialization stage. After coordinator/end-device connection, the end device sends data periodically with a flash of LED 1. When LED 1 is on, the coordinator also toggles to indicate reception of data. When Button 0 is pressed, LED 0 toggles and the LED 0 of all end devices will also toggle, after a delay.

4 System Implementation

A verification and demonstration EmQRG system with a M2M ZigBee-based home-type network is implemented. The EmQRG is embedded in an Intel mainboard with a Linux OS. Figs. 9(a) and 9(c) show our experimental network with a ZigBee-based temperature/humidity monitoring system and three PCs. The temp/humid system includes the IXP425, Host, a ZigBee gateway (coordinator) and a set of ZigBee sensor nodes as end devices. The Internet is simulated by two Cisco 7204 VXR and two 3845 routers. The EmQRG functions as the heart of the network, interconnecting the various devices. Every device in the network has its own IP address.

The EmQRG interface is based on a FudanTech High-Speed Network Development Platform using an Intel IXP425: FDIXP425-Dev Platform. The FDIXP425-Dev Platform provides: 32MByte Flash, 128MByte SDRAM, three independent 10/100M Ethernet interfaces, two 32bit 33MHz~66MHz PCI Slots, one MiniPCI slot for a WLAN card for 802.11a/b/g applications, six high-speed UART interfaces, two 32bit 33MHz~66MHz PCI Slots, two 32bit 33MHz~66MHz UART interfaces, two high-speed USB HOST interfaces, one USB device interface, RTC, GPIO and a JTAG debug interface [22].

The coordinator is based on a FT-6250 card, which is a development board embedded with a high power ZigBee wireless communication module interfacing with GPIO, UART, ADC, DAC or comparator systems. Embedded with the world’s first ZigBee system single chip, the JN-5121, the FT-6250 contains a high-performance 32-bit microprocessor, has a wide range of signal coverage and is designed for long-distance wireless sensor applications, permitting point-to-point transmission distances as long as 700 meters in an open outdoor environment. Indoor transmission is considerably shorter and is strongly dependent on the local environment, but compared to earlier ZigBee modules under similar conditions, the range is three to five times longer [4]. The FT-6250/FT-6251 system, using an external IPEX connector antenna, allows use of high-gain or directional antennae for special working environments.

The individual ZigBee-based temp/humid sensors employ a FT-6251 card, which is a development board similar to the FT-6250. We embed each FT-6251 with a digital temperature sensor (0° ~ 70 °C) and humidity sensor (5 ~ 95%).

![Fig. 8. FT-6250/FT-6251 network formation flowchart](image-url)
A flowchart for the FT-6250/FT-6251 network is shown in Fig. 8, where the coordinator (FT-6250) is connected to the IXP425 embedded system by an RS232C link with a 115200 baud rate. The data is updated every 500 ms.

State 1: Start sampling temperature data by calling vHTSstartReadTemp();
State 2: Get the temperature result by calling u16HTSreadTempResult();
State 3: Start sampling humidity data by calling vHTSstartReadHumidity();
State 4: Get the humidity result by calling u16HTSreadHumidityResult();
State 5: Store the temperature and humidity results to the parameter pu8Payload by calling vWUART_TxData();
State 6: Output the temperature and humidity results to UART by calling vProcessIncomingData().

```c
/* Copy frame data to serial buffer for output on UART */
for (i = 1; i < psFrame->u8SduLength; i++)
{
    vSerial.TxChar(psFrame->au8Sdu[i]);
}
```

The ZigBee-based temperature/humidity network requires initial setup. Threshold values must be set for each sensor. A database containing the static node topology must be established. We also establish a database whereby authorized users can access the home network [11]. We must also set up and test the implemented special functions of the specific system. In our test case, we have designed the temperature system to serve as a high-priority fire alarm which turns on a warning light bulb and can be preset to alert suitable persons or systems via SMS, FTP, make-a-call, email services, etc. The system response is, of course, adjustable by the user and depends significantly on the peripheral equipment of the system. In the case of room temperature monitoring, the system might send a command to turn on/off or adjust an air conditioner. In the case of monitoring the body temperature of a hospital patient, it might send a message to a specific nursing station.

5 Experimental Results

The fundamental feasibility of the proposed system is verified and demonstrated by real-time laboratory experiment. The schematic layout and the physical installation are shown as Figs. 9(a) and 9(b). Fig. 9(a) shows the DiffServ-capable network uses four Cisco series routers, 7204-1, 7204-2, 3845-1 and 3845-2, configured as DiffServ routers. They police and forward data according to the DiffServ field markings. A network processor IXP425 embedded system is the EmQRG. Three PCs function in the system, one as Client, one as traffic generator H1, one as a traffic receiver called Server. EmQRG and H1 are each connected by a 100 Mbps Ethernet link to Cisco router 7204-1. The link between the two DiffServ routers is a 10 Mbps Ethernet link, i.e. a bottleneck intentionally designed in to challenge EmQRG QoS capabilities. Cisco router 7204-2 then connects to the Server over a 100 Mbps Ethernet link. The Cisco router 7204-1 with CBQ traffic control of bandwidth/traffic management is for optimized use of bounded network resources. CBQ provides fine granularity bandwidth sharing and traffic priority control, including session level, thereby enabling service level guarantees for individual flows and aggregate traffic. The bandwidth allocation for each DiffServ-QoS traffic classification at the Cisco router 7204-1 is composed of 5 isolated channels with the non-sharable bandwidths, 3 of 1 Mbps for AF (assured forwarding) traffic, 1 of 3 Mbps for BE traffic and 1 of 4 Mbps for EF (expedited forwarding) traffic (10 Mbps bandwidth total). The EmQRG assigns forwarded traffic to the classes in Table 2. Security data gets highest priority followed by multimedia and other real-time applications, although these priorities are user-adjustable. For laboratory test, the Server is located so as to simulate an Internet or large LAN environment both with/without background traffic and with/without the above-mentioned bottleneck.

47
Experiment 1. Experiment 1 tests and verifies the EmQRG system in its primary context, i.e. as a basic Internet-LAN gateway and server for the home, office or small local network, coordinating and optimizing data traffic flow over a network with limited bandwidth. In keeping with modern real-world user trends, our testing
focuses on high-priority data-intensive video streaming, which presents the most severe challenges to network QoS and, thusly, consumer satisfaction. Less critical traffic such as email and ftp transfer are given lower priority. Testing is performed both without and with network congestion and bottlenecking. UDP video is delivered by vlc from Client and H1 to the Server, which contains a different IP for each sender. Congested conditions are even under congested and bottlenecked conditions. Embedded within the EmQRG, the experimental fire alarms of the ZigBee M2M system were received almost immediately under all experimental conditions, even maximally congested and bottlenecked.

Results of statistical analysis of experiment 1 are obtained by the command “sh interfaces accounting” at Cisco router 7204-1, showing: The total traffic has 148298 packets (104907880 bytes) with 48569 packet drops to be sent. BE Traffic is 134164 packets (85713908 bytes) and 2304 packet drops. EF traffic is 14134 packets (19193972 bytes) and 0 packet drops. Thus the server is showing the film with a 32.75% packet drop rate. Video captured during network congestion can be seen at the Server in the upper and lower video frames in Fig. 10(a).

In experiment 2, first, the X-10 power line network, Host, IXP425 and the FT-6250/FT-6251 ZigBee sensor network are integrated as shown in Figs. 11(a), 11(b) and 11(d). Then, as initialization, the emergency temperature threshold is set to 28°C. In the initial quiescent state, the warning light bulb of the X-10 power line network is off because the pre-set alarm temperature threshold is higher than the current temperature of any of the temperature sensors. An alarm condition is generated by touching a single temperature sensor with a human index finger (Fig. 11(b)). This raises the temperature/humidity to 28°C/76%, as displayed on the home page of the FT-6200 sensor system Fig. 11(c), i.e. it raises the sensor temperature to equal or greater than the emergency alarm/alert threshold. In consequence, the light bulb of the X-10 power line network turns on. Because the system is modeled as a fire alarm, the system also sends an emergency fire message to the user or administration center by communication modalities pre-set by the user. Even after an initial triggering, the light bulb turns on once more (Fig. 11(d)) when the temperature falls below the threshold value. Notably, the system continues to monitor the temperature and humidity of the sensors at all times, making it trivial to collect this data in a file and to print it out as an over-time temperature/humidity history of the each sensor’s local environment. However, this continually monitored data is of low priority, whereas the active fire alarm data is of top priority.

In fact, during real time experiment, there was no observable effect on streaming video performance as a consequence of the simultaneously functioning ZigBee wireless sensor network under normal or emergency alarm conditions, regardless of the level of network congestion.

6 Discussion

The above 2 experiments reasonably represent a composite H2H, H2M and M2M “home” as seen in Figs. 2 and 3, i.e. a home-type network or corner office network of dissimilar devices with dissimilar communication protocols, dissimilar data transfer speeds and different requirements for customer satisfaction. The experiments were implemented on real hardware. The QoS of the basic EmQRG home network was demonstrated to function well, even under congested and bottlenecked conditions. Embedded within the EmQRG, the experimental fire alarms of the ZigBee M2M system were received almost immediately under all experimental conditions, even maximum congestion and bottlenecking.
Chaining our SOA (service-oriented architecture)-based EmQRG as parallel or embedded networks presents no special new challenges. Note that SOA implementation of our basic EmQRG system is not included in this paper’s discussion, although details of the SOA-EmQRG are available in [24]. Chaining means that the basic hardware must be duplicated for each new network. The biggest issue is the specific set-up of new components in each network. The SOA nature of the architecture simplifies much of this issue, making new component systems relatively easy to install. It also makes it easy for larger community networks to contain and protect smaller home networks.

It is expected that Fig. 1’s wireless networking of dissimilar and composite networks will become increasingly ubiquitous. It is likewise expected that the need for fast high-volume data traffic will increase. QoS requirements will similarly increase, with collateral need for automatic system adaptability to network overload. Additionally, the prevalence of M2M ZigBee-based and other embedded networks is likely to increase in the home/office/community/etc networks of the evolving IoT world [2]. The proposed system offers and expedites all these features, using only presently available hardware and software. We consider that the presented system offers a practical core to the solution of these issues.
The warning light bulb of the X-10 power line network is off when the current temperature/humidity is less than the alarm/alert threshold.

The warning light bulb of the X-10 power line network turns on when the sensor reaches/exceeds the alarm/alert threshold; the system also sends an emergency fire alarm message to the user or administration center.

The temperature/humidity has reached the alarm/alert threshold, which is displayed on the home page of the FT-6200 sensor system.

The warning light bulb turns off again when the temperature/humidity falls below the threshold value.

Fig. 11. Results of experiment 2
7 Conclusions and Future Work

This study has considered the foreseeable future as increasingly ubiquitously networked. High intercompatibility, high QoS and both super- and sub-embedability are demanded in a ubiquitously networked world. Toward the realization of this vision, this study has presented an extension of our previously published EmQRG QoS-aware residential gateway. The basic EmQRG system and its special mechanisms have been summarized. The experimental results have confirmed that EmQRG maintains QoS high standards for multimedia, even during experimentally bottlenecked and congested network conditions. Wireless M2M networking capability in conjunction with normal EmQRG function is demonstrated by embedding within the EmQRG a wireless ZigBee-based multi-module temperature/humidity network to monitor the temperature and humidity of the network’s environment. Laboratory experiments consider the temperature/humidity network as a fire alarm, triggering when the temperature sensor exceeds a designated value. The EmQRG system then activates a warning light bulb and sends emergency messages to designated recipients by designated modalities. Experimental results verify that the emergency alarm is generated and received “immediately,” regardless of congestion of the basic EmQRG network, at the same time maintaining high QoS streaming video. Discussion has shown that the presented system is easily expandable to extensive networks containing many parallel systems and/or in conjunction with many embedded sub/super-systems. The presented system is cost-effective, easy-to-use, easy-to-implement and can be implemented with available hardware and software. The presented system thus constitutes a practical example of and approach to expedited implementation of the ubiquitously networked world.

Acknowledgement

This research was supported by National Science Council of Taiwan under project number NSC 100-2221-E-277-001.

References

[3] ZigBee Alliance, on http://www.zigbee.org


