

EcoPlex: Empowering Compact Wireless Sensor Platforms via Roaming and Interoperability Support

Chung-Yi Ke, Nai-Yuan Ko, Chih-Hsiang Hsueh, Chih-Hsuan Lee

¹Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 30013

Pai H. Chou^{1,2}

²Center for Embedded Computer Systems
University of California, Irvine
Irvine, CA 92697-2625 USA

Abstract—EcoPlex is an infrastructure that enables simple wireless platforms to participate seamlessly in a feature-rich, wireless ad hoc network. EcoPlex consists of gateways that are responsible for handoff support for mobility and high data rate without burdening the simple nodes to implement multi-hop protocols. The gateways also create virtual identities for simpler nodes to enable their participation in the feature-rich network without adding complexity to them. We demonstrate the feasibility of this idea with the ultra-compact wireless sensor platform called Eco to participate as virtual nodes in a fully general ZigBee network. Experimental results show EcoPlex to be efficient and scalable. The enhanced mobility and interoperability are added to the Eco platform at the infrastructure level, all with minimal node complexity.

I. INTRODUCTION

Low-cost, *ultra-compact* wireless systems are becoming an increasingly important part of mobile and ubiquitous computing. By ultra-compact, we mean a complete wireless system on the order of 1 cm³ or smaller including battery, antenna, microcontroller unit (MCU), radio, I/O, and possibly storage. This is in contrast to conventional handheld devices (cell phone, MP3, and motes), which are 2–3 orders of magnitude larger. Table I shows a size comparison. The small size means they are less intrusive to deploy, while low cost means they are amenable to many more applications. However, to be truly low cost and small, they must be resource-constrained and thus can afford to implement only bare-essential features.

Interoperability with full-featured standard protocols is often sacrificed as a non-essential feature, especially because it often imposes extra burden on the hardware and software. Interoperability is desirable and powerful, because it enables the user to leverage and compose existing systems without having to re-invent the solutions. For example, it would be very powerful if every wireless sensor node implemented a TCP/IP stack, but this is simply not the case due to high overhead. To achieve interoperability without burdening the compact nodes, we propose adding support at the infrastructure level. To demonstrate the feasibility of this idea, we developed EcoPlex, a network of gateway nodes to enable these compact nodes

This work was sponsored in part by the National Science Foundation (USA) Grants CNS-0448668 (CAREER) and CNS-0721926, the National Science Council (Taiwan) Grant NSC 96-2218-E-007-009, and Ministry of Economy (Taiwan) Grant 96-EC-17-A-04-S1-044.

TABLE I
SIZE COMPARISON OF MOBILE AND UBIQUITOUS DEVICES. ALL INCLUDE BATTERY.

System Name	Size (cm ³)	prog. size or flash	features included
Eco	1	4 KB	antenna, accelerometer
μ Part	2–4	64 words	sensor, no antenna
Bluetooth ear.	1.9	n/a	earphone, microphone, ant.
iPod Shuffle	6.17	4 GB	music player, no RF, no ant.
Telos rev.B	~38	48 KB	sensor, chip antenna
Mica2DOT	~10	128 KB	dep. on sensor module
Mica2	≥50	128 KB	dep. on sensor module

to participate in full-featured networks by creating *virtual identities* and supporting *roaming*.

A. Case Study: ZigBee

ZigBee is a standard for wireless sensor networks (WSNs). It is possibly the most popular among researchers due to its support for ad hoc mesh networking. ZigBee is a networking layer on top of IEEE 802.15.4, which is a medium access control (MAC) protocol based on carrier sense, multiple access with collision avoidance (CSMA/CA). Although 802.15.4 is indeed relatively simple, ZigBee is quite feature-rich. ZigBee defines *profiles* for several application classes, including lighting control, media control, security alarm, automated meter reading, healthcare, etc. The typical code size of a ZigBee stack is 64–100 KB, which might seem trivial for general-purpose computers, but it is quite high or even prohibitive for many cost-sensitive applications. In fact, a typical “ZigBee-ready” platform such as Telos Revision B [1] contains 48 KB of program memory, barely enough for the ZigBee stack, let alone any operating system or application code. As a result, researchers either implement only a subset of these protocols or sacrifice interoperability by building their own, lighter-weight stacks.

Besides interoperability on homogeneous hardware, interoperability between heterogeneous platforms may be important or necessary. On the high end, one may want ZigBee to be integrated into IP networks. At the personal area network level, interoperability with Bluetooth or Z-Wave may be desirable. For example, a ZigBee application may need to input data from a Bluetooth-only device or to output commands to a Z-

TABLE II
COMPARISON OF WIRELESS PAN TECHNOLOGIES

Type	Stack size	Network	Target application class
Wibree	4–16KB	Pt-to-Pt, Star	low-energy Bluetooth
Z-Wave	10–32KB	Mesh	home automation/control
ZigBee	64–128KB	Mesh, Star	wireless sensor network
Bluetooth	≫128KB	Star (piconet)	PC/phone peripheral

Wave-only device for home control and automation. Besides Bluetooth and Z-Wave, it may be more effective for a ZigBee network to incorporate data from other non-ZigBee platforms with better size, data rate, or power considerations. For example, μ Part [2] and Eco [3] are two ultra-compact wireless sensor platforms on the order of 1–4 cm³ including battery and antenna, but their respective program memory sizes are 64 words (not KB) and 4 KB, or 1–2 orders of magnitude smaller than ZigBee. The small form factor enables new applications such as infant monitoring, user interface devices, consumer behavior tracking, etc. End users care about using the right solution that satisfy their constraints, and most would not care about the actual protocol as long as the platforms are interoperable.

B. Contributions and Paper Organization

The main contribution of this work is to demonstrate the effectiveness of the infrastructure approach that supports interoperability and roaming. We choose ZigBee as the complex standard and Eco as the compact platform for our study. EcoPlex extends ZigBee networks by adding not only the link layer to Eco nodes but also presenting them logically as nodes in the ZigBee network. For the nodes to work with EcoPlex, we develop EcoMAC, a lightweight, pulling style protocol that is suitable for highly resource-constrained sensor nodes such as Eco. The roaming support adds mobility to relatively simple nodes within the coverage area and can enable complex nodes to perform high-bandwidth communication that would otherwise be infeasible if relayed.

The rest of this paper is organized as follows. Section II describes the related works. Section III provides the background on an Ethernet-ZigBee gateway we have previously built, an overview of the ZigBee protocol, and a description of Eco. Section IV details our system design. Section V presents the experimental results. Section VI concludes with a discussion of future work.

II. RELATED WORKS

A. Wireless PAN Technologies

Table II shows a comparison of some popular wireless personal area network (PAN) technologies available for mobile and ubiquitous computing. ZigBee and Z-Wave are two similar wireless technologies with overlapping target applications. They are both mesh networking protocols: nodes in a Z-Wave network are divided into Controllers, Routing Slaves, and Slaves, which correspond to Coordinators, Routers, and End Devices in ZigBee. Z-Wave targets home automation and

control applications; ZigBee was initially motivated by WSNs but ended up also including home control and automation as a target. The code size of a Z-Wave stack is about 10–35 KB on 8-bit MCUs, about 1/2 to 1/3 the size of ZigBee ones for the same type of MCU (e.g., 8051). Due to the simpler protocol stack and smaller code size, Z-Wave chips that integrate the MCU, RF, and I/O can be made smaller and lower cost than ZigBee ones.

Bluetooth is the most well established among them and the most heavyweight. Bluetooth can be found in most cell phones, laptop computers, keyboards, and earpieces. Bluetooth is a master-slave protocol, where nodes form a star network called *piconet* with the master, and multiple piconets can form a *scatternet*. Bluetooth modules are often implemented with a 32-bit embedded processor. To be competitive in the mobile and ubiquitous space, Bluetooth Low Energy Technology, also known as Wibree, has been proposed as a simple subset of the upcoming Bluetooth 3.0 standard. The hardware will consume lower energy, deliver higher data rate of up to 2 Mbps, and the protocol stack complexity will be a fraction of Z-Wave.

Interoperability of these protocols with other protocols has primarily been TCP/IP, in the form of 6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks) [4] over 802.15.4, and IP over Bluetooth. However, currently there is no interoperability among these wireless protocols except between Wibree and Bluetooth 3.0. IP interoperability would be highly desirable, as each node would be made accessible by any computing system that speaks TCP/IP. Cooper [5] and Shon [6] each implemented a subset of TCP/IP protocol stack to allow TCP/IP communication with embedded devices, while two embedded TCP/IP stacks, LwIP and uIP, are proposed by Dunkels [7]. However, implementation of a TCP/IP protocol stack comes at a high price for these resource-constrained nodes, which find even the ZigBee stack to be too heavyweight. To enable interoperability without burdening the nodes, we propose to implement the interoperability feature at the infrastructure level.

B. Infrastructure

Infrastructure support is an important part of many wireless system deployments, even for mesh networks that are supposed to operate without centralized control. For instance, in the WSN literature, the Extremes Scaling Network [8], the James Reserve Extensible Sensing System [9], and the Great Duck Island [10] deployments are examples where tiered network organization enabled greater spatial scale and increased network capability. Rhee et al [11] describes the design of a tiered network for increasing the lifetime of the entire network. Lynch et al [12] discusses tiered networks for structural health monitoring. The Tenet architecture by Gnawali et al [13] simplifies application development and results in a generic mote tier networking subsystem that can be reused for a variety of applications, all without significant loss of overall system efficiency. These tiered networks effectively provide infrastructure support that enables nodes to be simpler, but the nodes are homogeneous in terms of wireless protocol in a

given tier, and they assume the nodes to be stationary rather than mobile.

C. Heterogeneous Networks

Previous works have considered *vertical interoperability* where a higher-tier network provides a bridge to the fully general network, but few if any work considers *horizontal interoperability* within a given tier. Dunkels et al proposes an adaptive communication architecture for wireless sensor networks [14]. They notice the same problem that various sensor network protocols have been proposed, but existing communication architectures for sensor networks are not designed for this heterogeneity. They design an adaptive communication architecture that enables users to write the applications without knowing the underlying protocols. However, the adaptive communication architecture acts as a virtual machine that runs on top of the sensor nodes. Moreover, it achieves interoperability by increasing node complexity and thus may be difficult for highly resource-constrained nodes.

D. Roaming Support

Roaming support has been proposed and is in use in many wireless networks, including Global System for Mobile Communications (GSM), Wi-Fi, and Bluetooth. Most published works on handoff algorithms have been done for Bluetooth. In George et al [15], the authors discuss three soft handoff methods. In these methods, the base stations can measure the relative received signal strength (RSSI). When the RSSI drops below the threshold, the base station initiates the handoff procedure. Mobile Bluetooth Public Access (mBPAC) handoff protocol [16] is a hard handoff algorithm, where the base station and Bluetooth devices periodically poll each other to check the existence of the link. If the link is disconnected, then the base station requests the nearby base stations to page the Bluetooth device through a wired network. ZigBee relies on ad hoc networking capability to obviate handoff support, but it does so adaptively without planning or consideration for continued communication, and therefore the latency or interruption may be long.

III. BACKGROUND AND PROBLEM

This section first provides a background on the system platforms that we have constructed as a basis for the proposed EcoPlex system, including an Ethernet-ZigBee gateway called EZ-Gate and the ultra-compact wireless sensor node called Eco. Based on these platforms, we give the problem statement for EcoPlex, including the design objectives and metrics for evaluation.

A. EZ-Gate

EZ-Gate [17] is an Ethernet-to-ZigBee gateway that we have developed. As the name implies, it bridges between a ZigBee PAN and TCP/IP over Ethernet. The Gateway has an architecture similar to a consumer-grade Wi-Fi/Ethernet router, except that it supports ZigBee instead. EZ-Gate is an embedded Linux computer with ample flash and RAM for

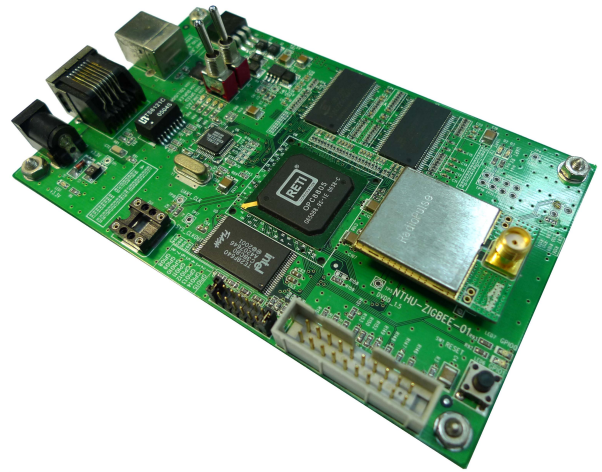


Fig. 1. The fully assembled EZ-Gate board.

running user applications. The hardware consists of the gateway main board and the RadioPulse LM-2400 module. The gateway main board contains a RETI ARM9-based broadband network processor with an integrated 10/100-base/T Ethernet port, a PCI host bridge, ATA133 IDE controller, USB 2.0 root hub, a security cryptographic co-processor engine, and several multimedia I/O interfaces. It has 64 MB of DDR RAM and 256 MB of flash memory for the file system. For the radio module, the RadioPulse LM-2400 module contains an integrated 8051 MCU+RF system-on-chip. We programmed it with a full ZigBee Coordinator stack, and thus it serves as a ZigBee coprocessor to the ARM9. In addition, we have developed graphical user interface (GUI) tools for the users to manage EZ-Gates and ZigBee nodes. A photo of the EZ-Gate board is shown in Fig. 1.

A ZigBee network consists of three types of participants: ZigBee Coordinators (ZC), Routers (ZR), and End Devices (ZED). EZ-Gate implements the Coordinator and Router roles. A Coordinator serves as the root of the PAN and may bridge to other PANs. A Router acts as an intermediary node that routes data from other devices within the same network. An End Device is a leaf node and can communicate only with its parent node, which may be either the Coordinator or a Router. Messages from an End Device are relayed by its parent node to another End Device. In addition, ZigBee devices must agree on a profile in order to communicate with each other.

B. Eco Wireless Sensor Platform

Eco is an ultra-compact wireless sensor platform and consists of Eco nodes and base stations. An *Eco node* is 1 cm³ in volume and weighs under 2 grams, including a custom Lithium-polymer battery and a chip antenna. The battery can be charged in the system. An *Eco base station* is a transceiver module that provides an uplink for the Eco nodes to the host computer or a gateway. Several versions of the base station are available, including one with a Fast Ethernet interface, USB 2.0 interface, and UART interface. For the purpose of this

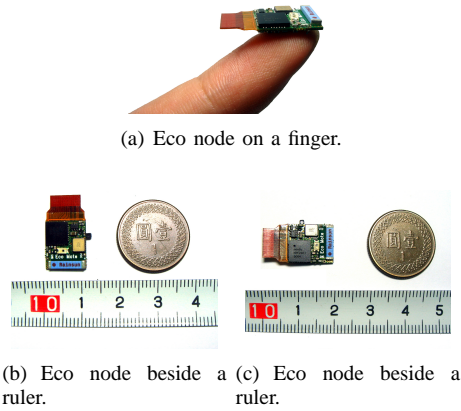


Fig. 2. Eco wireless sensor node (a), (b), and (c).

work, we chose the USB version. Photos of the Eco platform are shown in Fig. 2.

Both the Eco node and base station use integrated MCU+RF SoCs based on Nordic’s ShockBurst protocol in the 2.4 GHz band, though it is incompatible with IEEE 802.15.4 used by ZigBee. The base station can be programmed to run autonomously without step-by-step intervention from the host computer. However, the small form factor means resources are constrained. The Eco node contains 4 KB of RAM shared between program and data, while the base station contains 2 KB of data RAM and 16 KB of program flash. The scarcity of resource means these nodes cannot afford to implement complex networking protocols, as most of the available memory should be devoted to the application.

C. Problem Statement

The problem statement of this work is to develop infrastructure support for two purposes: Eco-ZigBee interoperability and roaming. The first is to enable Eco nodes to appear in a ZigBee network purely by infrastructure support. The only modifications are to augment the EZ-Gates with Eco base station modules and to modify EZ-Gate and base station software. Each Eco node is to be connected to a base station in a star topology, while ZigBee nodes should be able to discover them, communicate, and form mesh networks with them just like any other ZigBee End Devices. To enable roaming of Eco nodes, the EZ-Gates are to negotiate and perform handoff for the best possible link quality.

The objectives are as follows. First, the interoperability support should incur minimal or no modifications to the hardware and software of Eco and ZigBee nodes, while the base station and EZ-Gate can afford to implement most or all of the protocol complexity. Second, overhead incurred by handoff support should be minimized. Another way of stating the second objective is to maximize the available, sustainable bandwidth for user payload, even during the handoff process, as well as minimizing the latency of communication.

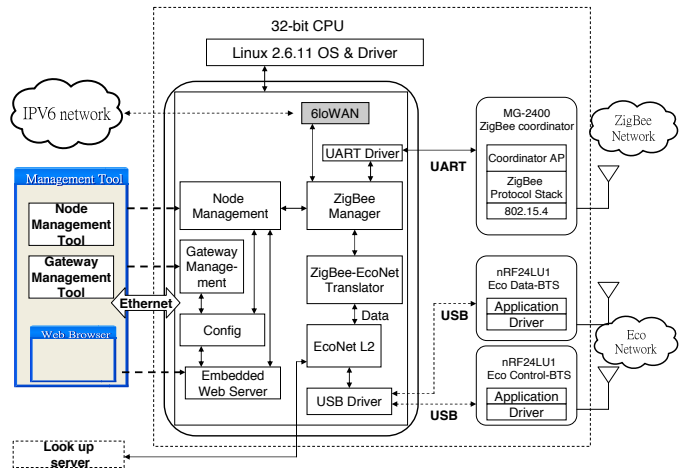


Fig. 3. System structure.

IV. TECHNICAL APPROACH

This section describes our technical approach to enabling simple nodes to participate in full-featured networks and roaming via infrastructure support. Our approach is divided into several levels of abstraction: multi-radio gateway, data link and media access control, tiered network organization, creation and bookkeeping of virtual identities, and handoff support.

A. Multi-Radio Gateway

In our terminology, a *gateway* refers to an embedded computer that performs routing and handles complex protocols, whereas a *base station* is a simpler concept that transmits or receives wireless packets on behalf of another computer, possibly a gateway. Therefore, a gateway may actually contain multiple base stations of same or different types. At the hardware level, the first step towards interoperability is to add the radio transceivers needed. Our EZ-Gate platform already includes the RadioPulse module for supporting ZigBee protocol processing, 802.15.4 MAC, and RF. To support Eco, we attach two Eco base stations to EZ-Gate’s USB port. One common reason for two transceivers instead of one is that each radio is half duplex, and two radios make it full duplex. In this case, our reason for using two transceivers per gateway is to dedicate one public channel to shared control of nodes between gateways and one private channel for intra-gateway use with its nodes. The architecture of the gateway is shown in Fig. 3.

B. Link Layer: Pulling

Our approach to keeping the resource-constrained nodes as simple as possible is to adapt a *pulling*¹ protocol named EcoMAC. That is, simple nodes act as passive *thin servers*,

¹Some readers may ask if we mean *polling* instead of *pulling*. The purpose of polling is to check the status of a device or a flag in shared memory as a condition for executing an action. Pulling, on the other hand, is a communication style where the client requests and the server replies.

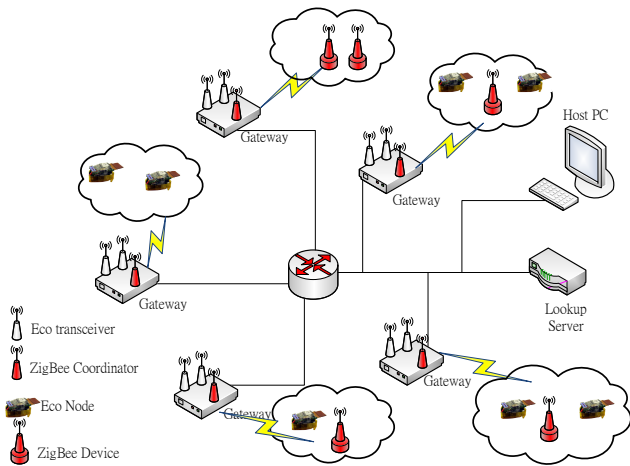


Fig. 4. Network structure of EcoPlex.

while larger nodes such as gateways act as *fat clients* that actively pull data from them. This is in contrast to *pushing* style protocols, where nodes actively transmit data either during assigned time slots or after checking channel availability. Pushing may seem more intuitive, as it allows nodes to make their own decisions on communication and power management; in perfect communication, it uses only the necessary bandwidth and nothing extra. However, uncoordinated multiple access with pushing can result in collision. Moreover, if reliable communication is required, then nodes would still need two-way traffic by returning an acknowledgment message. On the other hand, pulling is very simple to implement, requires no time synchronization, and can implement reliable transmission, since a pulling message can be used effectively as a NACK – by pulling the previous packet again. EcoMAC is based on the EcoDAQ [18] protocol.

C. Tiered Network Organization

As a superset of a ZigBee network, EcoPlex generalized the tiered network structure. The lower-tier network consists of ZEDs and Eco nodes over wireless links. They communicate with each other or with EZ-Gates, which form the mid-tier network. The top-tier network of EcoPlex consist of *Lookup Servers*, which are analogous to a dynamic *domain name server* (DNS) for the Internet, for tracking the association of nodes with gateways. Lookup Servers are used for routing data messages and handoff. Although the current implementation assumes one Lookup Server, there can be multiple Lookup Servers, making it possible to do distributed processing and thus be scalable to large networks without a central bottleneck. The top-tier and mid-tier networks are assumed to be two separate but bridged Ethernet networks. The network structure of EcoPlex is shown in Fig. 4.

D. Virtual Identity

To enable simple nodes to participate in the full-featured network, our approach is for the infrastructure to create and maintain virtual identities for them as well as routing messages

for them. For each Eco node that joins the EcoPlex network, we allocate a unique ZigBee address. However, an identity is more than just a unique network ID, because each ZigBee node can implement several *endpoints* or logical devices. A typical ZigBee node therefore maintains a list of ZigBee-defined data structures called *descriptors*, each of which specifies the node’s capabilities and roles in a given ZigBee profile (i.e., application class). To reduce burden on the Eco nodes, we make each Eco node maintain only a bitmap, rather than the descriptors themselves, to indicate which endpoints that it implements. The gateway then expands the bitmap into the full ZigBee descriptor on behalf of the Eco node. All other states pertaining to each Eco node are maintained in a database by the Lookup Server.

An Eco node is almost always passive, except when it needs to join the EcoPlex network. In that case, it will actively transmit a *Join-Request* message on the control channel. Those gateways that hear this request will synchronize with the Lookup Server, which will allocate a new node ID if one has not already been assigned to the node previously and appoint one of the gateways to be the initial owner. During normal operation, each gateway operates in promiscuous mode to snoop and receive packets that could be destined to any of its Eco nodes or ZigBee nodes, and it routes the packets to the destined node. Packets generated by an Eco node and destined to a ZigBee device are translated by the node’s owner gateway and transmitted on behalf of the node, and vice versa.

E. Roaming Support

Our approach to roaming support can be divided into state maintenance, link-quality tracking, and handoff coordination. Roaming requires the infrastructure to keep track of the gateways involved. To decide whether to perform a handoff, the infrastructure keeps track of the packet loss statistics. In case multiple nodes need to undergo the handoff process, the infrastructure enforces serialization, both for consistency and to prevent collision. To decide which of the candidate gateways will be selected, the gateways coordinate with each other based on link quality, load balancing, bandwidth availability, and other considerations. Then, they synchronize their decision with the Lookup Server. The steps of the handoff process are as follows.

First, in normal state, each gateway periodically measures the link quality. As the link quality deteriorates, the gateway asks the node to increase its transmission power if possible. When the link quality drops below the threshold, the gateway sends a *Handoff-Request* message to the Lookup Server to initiate the handoff. In the mean time, the node continues its regularly scheduled data transmission. After the gateway receives a *Handoff-Response* message from the Lookup Server, it sends a *Handoff-Start* message to the Eco node to start the handoff process. The Eco node may also lose connection with this gateway, in which case it would attempt rejoining any gateway that can hear it. The Lookup Server selects the gateway that has the best link quality from a list of candidates.

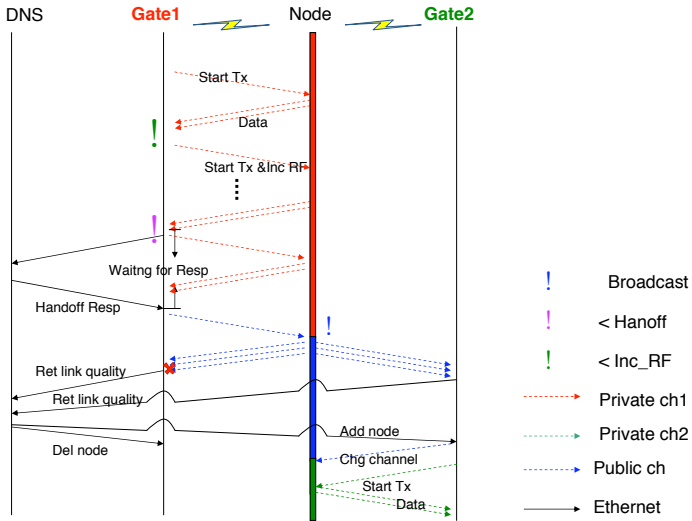


Fig. 5. Handoff Process.

V. EVALUATION

This section presents experimental results to show effectiveness of EcoPlex. Not only should EcoPlex work correctly for handoff and virtual identity features, but it should also perform well in terms of four objectives: data throughput, handoff latency, load balancing, and code size. We also discuss potential issues with the current implementation and possible solutions.

A. Experimental Setup

Our experimental setup consists of both Eco and ZigBee sensor nodes and EZ-Gates. Due to the limited number of EZ-Gates available at the time of this writing, we created substitute EZ-Gates with laptop computers running Linux 2.6.15 with attached Eco and ZigBee transceivers in the same way they are attached to EZ-Gates. For all practical purposes, these laptops perform identically to EZ-Gates. One of the laptops runs the Lookup Server. We configure the default RF transmission for -10 dBm, rather than the usual 0 dBm. The reason is that a lower power level enables EcoPlex to be much more sensitive to the relative distance of the Eco nodes to the gateways. To measure the timing accurately at each stage, we use the Tektronix TDS2024B oscilloscope [19] that features 200 MHz bandwidth, 4 channels, and a sampling rate of up to 2.0GS/s in real time.

B. Throughput and Latency

1) *Stationary Nodes*: Fig. 7 shows the average data throughput while nodes are stationary. The payload length of one RF packet is 27 bytes, of which 23 bytes are used as the actual data field for user applications. Therefore, we individually discuss the data throughput in terms of 23 bytes and 27 bytes. We vary the number of reply packets per pull and measure the results in a highly controlled environment with no packet loss. The achievable data throughput is

TABLE III
EXPERIMENTAL PARAMETERS

Parameter	Value
Number of replied packets	2 ~ 3
Wireless data rate	1Mbps
Payload length	27 bytes
Tx power level of base station	0 dBm
Default Tx power level of Eco node	-10 dBm
Number of broadcast packets one time	10
Maximum ReTx by software	3
Default Handoff Threshold	0.85
Default Inc_RF_Pwr Threshold	0.95



Fig. 6. A substitute gateway node implemented with a Linux laptop equipped with two transceivers for experimental setup.

- for two reply packets per pull, 38.33 kbits/s (23-byte payload) and 45 kbits/s (27-byte payload)
- for three reply packets per pull, 43.12 kbits/s (23-byte payload) and 50.625 kbits/s (27-byte payload)

The data throughput of EcoMAC is higher than that of a CSMA-style MAC (29.65 kbits/s) and slightly lower than that of a TDMA-style MAC (53.456 kbits/s) [18]. We can further improve the data throughput to 52.571 kbits/s (23-byte payload) and 61.71 kbits/s (27-byte payload) by increasing the number of reply packets per pull to 10. However, the round-trip time of pulling all nodes also increases with the number of reply packets. The round-trip time of pulling all nodes increases linearly with the number of nodes. In addition, the round-trip time also increases linearly with the number of replied packets as shown in Fig. 8. Excessively long round-trip time may decrease the sensitivity of handoff detection. If the mobility is high, then the mobile may miss the best timing for handoff while waiting during the round trip time. Currently, we choose 2 or 3 replies per pull for high sensitivity of handoff detection. If the command packet is lost, the base station will perform retransmission by software. The extra delay for each retransmission is about 12 ms, and the long delay will decrease the data throughput significantly as shown in Fig. 9.

2) *Mobile Nodes*: Fig. 10 shows the data throughput for mobile nodes while several nodes in the same PAN need handoffs at the same time. When only one node needs to perform handoff, the data throughput decreases slightly due to the overhead of handoff handling on the base station. In case multiple nodes need handoff, the aggregate data throughput can still be approximated with that for the case of one-

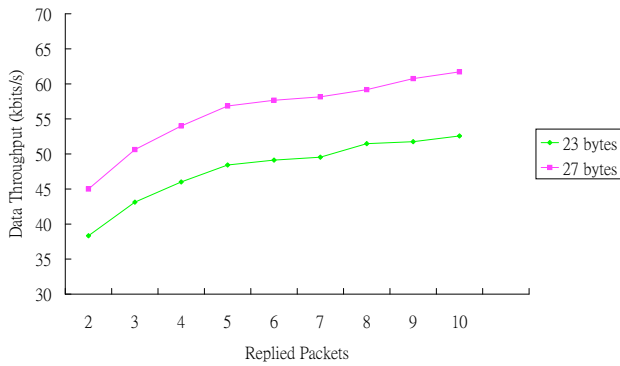


Fig. 7. Data throughput for stationary nodes with multiple replies

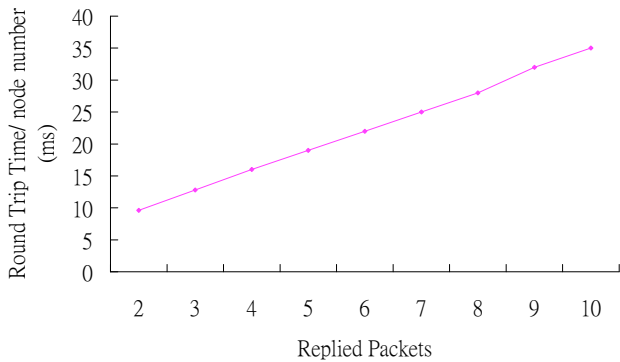


Fig. 8. Average time one node takes for stationary nodes with multiple replies

node handoff. Because the Lookup Server schedules only one handoff at a time, other nodes will need to wait until the previous handoff finishes. The base station still schedules the next data transmission time for those nodes that are waiting for *Handoff-Response*. Therefore, the total data throughput decreases by only 3.3% ~ 5.7% as the number of concurrent handoff requests increases.

C. Handoff Duration

Although the number of pending handoffs does not adversely affect throughput, the waiting time increases linearly with the number of handoff requests. The waiting time for

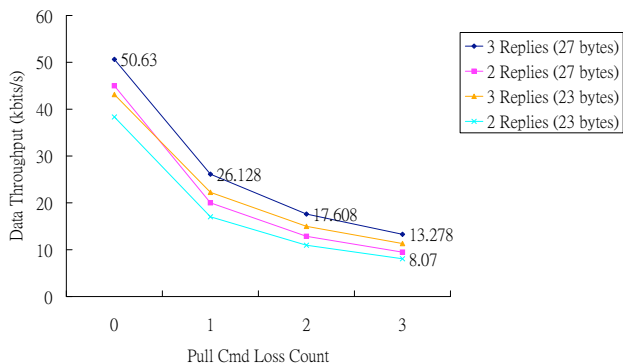


Fig. 9. Data throughput for stationary nodes when pull command loss

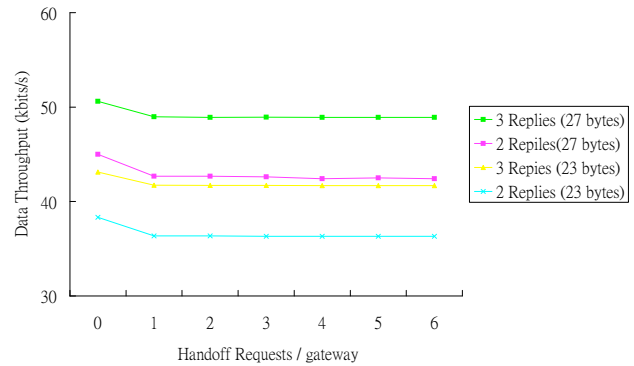


Fig. 10. Data throughput for mobile nodes while several nodes in the same PAN need to perform handoff

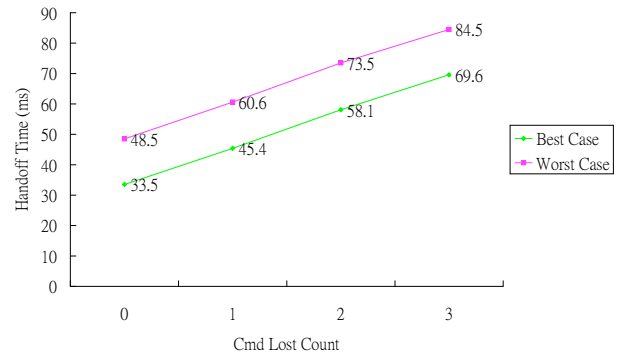


Fig. 11. Handoff time vs. command loss

a node to receive a *Handoff-Response* message from the Lookup Server is relatively short, about 15 ms. However, if there are five other nodes in a given PAN waiting for handoff, then the waiting time for the 6th node is about 345 ms. The Lookup Server permits the nodes of different PANs to broadcast and initiate handoff at the same time, so that each node does not need to wait for nodes in other PANs to begin handoff.

1) *Total Handoff Time*: The total handoff time is the duration from the time a node broadcasts 10 packets on the public channel (for link quality test) until the time it successfully switches to the new gateway. The channel reconfiguration time consumes about 200 μ s. Fig. 11 shows that EcoPlex exhibits fast handoff. The shortest total handoff time is about 33.5 ms in the best case and 48.5 ms in the worst case, since the handling time for 10 broadcast packets varies from 15 ms to 30 ms. The handoff time of EcoMAC is close to that of Bluetooth, which is around 15 ms, and this is rather competitive. However, for each control message lost during handoff, it incurs 12 ms of extra delay for each retransmission in purely software implementation. Hardware support for acknowledgment and retransmission can cut the delay to 250 μ s (48 \times speedup) using the Enhanced Shockburst MAC, which is used in the next generation prototype of the Eco node.

2) *Timing Breakdown*: Fig. 12 shows a breakdown of the time consumed by each stage of the handoff process. The time

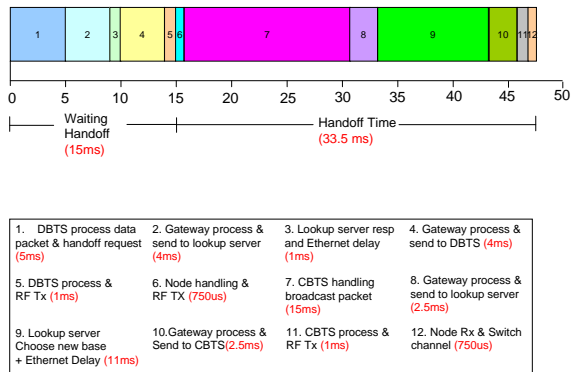


Fig. 12. Consuming Time of each segment during handoff process

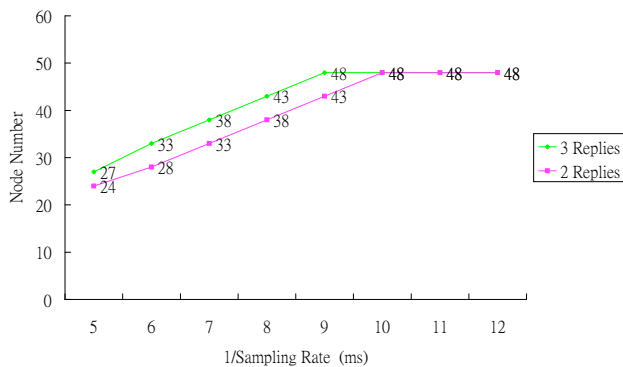


Fig. 13. Number of nodes a gateway can handle over different data rates.

can be divide into two main parts: *handoff-waiting time* and *handoff time*. The handoff-waiting time is measured with the assumption that no other nodes are undergoing handoff at that time. Therefore, the Lookup Server can immediately respond to the handoff request. Otherwise, the handoff-waiting time will become the bottleneck.

3) *Scalability*: Fig. 13 shows the maximum number of nodes that an Eco base station can handle at different data rates. If each node samples data every 5 ms, then a base station can handle up to 24 nodes if each node replies with two packets per pull; and up to 27 nodes for three reply packets per pull. As the sampling rate decreases, the number of nodes each base station can handle is up to 48.

D. Code Size

The total firmware size on our Eco node in EcoPlex is 2504 bytes, where 1084 bytes are taken up by the drivers for RF send/receive, SPI read/write, ADC read/write, etc. EcoMAC occupies 1420 bytes, and this shows that we can indeed keep the node complexity very low. This code size is very small comparing with other MAC protocol such as B-MAC [20], S-MAC, X-MAC, TDMA, etc [21], as shown in Fig. 14. Some of them are implemented on top of TinyOS, and the whole code size with TinyOS is about 17 ~ 19 Kbytes, which is

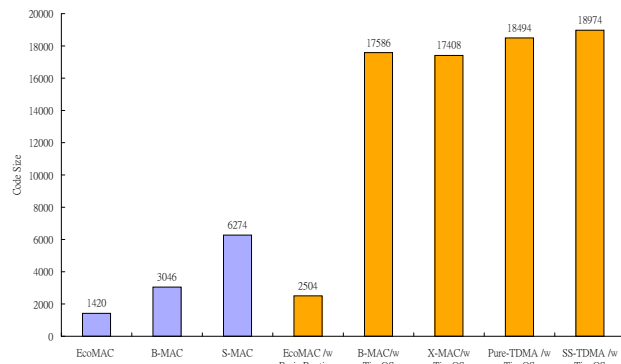


Fig. 14. Code size comparison of MAC protocols for low-complexity PAN.

larger than our code size and exceeds the size of Eco's total EEPROM capacity (4096 bytes).

E. Discussion

From above experimental results, we can see two major bottlenecks: handoff waiting time and packets retransmission. The average waiting time for handoff response increases as several nodes in the same PAN move together. Due to the fact that nodes in the same PAN interfere with each other if they broadcast simultaneously, the Lookup Server forces those nodes to serialize handoff. Serialization may be inefficient, and one possible solution is *cluster handoff*, where one representative node interacts with the gateways on behalf of a cluster of nodes that experience similar RF conditions.

The most time-consuming operation during the handoff process is the broadcasting of N packets on the control channel for testing link quality. To reduce the total handoff overhead, if several nodes need to perform handoff concurrently, then the cluster-handoff scheme will allow only one node per cluster to broadcast at a time. After the Lookup Server determines the new gateway of that node, it will notify all nodes in that cluster to switch to the private channel of the new gateway. It also notifies the original gateway to remove those nodes in that cluster from its PAN and notifies the new gateway to add those nodes into its PAN. By the cluster-handoff scheme, those nodes in the same cluster can switch to a new gateway concurrently in a short time without taking much time to broadcast serially. However, this scheme has to absorb the risk of unsuitable assignments of gateway to nodes in these cases, since other nodes have not broadcast packets for measuring link quality.

Figs. 9 and 11 show that throughput and handoff time are very sensitive to packets loss. The reason is that the current Eco hardware does not support auto retransmission and auto acknowledgment. Software implementation of retransmission incurs extra 12 ms delay each time. With hardware-supported auto retransmission (in the next generation Eco nodes, being prototyped right now), retransmission time can be cut down to 250 μ s, or 48 times faster than our current software implementation.

VI. CONCLUSIONS AND FUTURE WORK

We present EcoPlex for enabling ultra-compact Eco nodes to participate in full-featured ZigBee networks. The infrastructure support for virtual identity and roaming empowers these resource-constrained but highly wearable nodes to contribute to a variety of sensing applications and leverage rich collections of tools without burdening them with complexity due to generality. We believe that *horizontal interoperability* between different wireless PAN technologies (ZigBee, Bluetooth, Z-Wave) is becoming increasingly important, as no one standard clearly dominates another in all areas. This effort is orthogonal to *vertical interoperability* with TCP/IP already done for most tiered network structures. What we have demonstrated with ZigBee and Eco is that not only is horizontal interoperability feasible but it can also work efficiently.

Future work includes localization, interoperability with additional wireless PAN protocols, and cluster handoff. First, we do not currently know the actual location of the nodes or even the gateways themselves. By knowing their locations or physical layout, it may be possible to more accurately estimate the best choice of gateway to handle a given node. We have demonstrated the ideas for very simple Eco nodes and mid-complexity ZigBee networks. We expect other network standards to present similar issues. Cluster handoff seems to be one feature that will overcome a major bottleneck that cannot be addressed by hardware upgrade alone, but its effectiveness has not been demonstrated. Finally, we are applying this system to several real-world applications involving mobile subjects.

ACKNOWLEDGMENTS

The authors thank members of the EPL team for development of Eco and EZ-Gate. Chulsung Park designed the initial version of Eco and Chien-Ying Chen further improved the design. Tsung-Hsuan Li ported the Linux kernel of EZ-Gate and Pang-Neng Li helped with hardware manufacturing. An-Ping Wang, Wei-Han Chen, Yu-Ting Chen, Yi-Hsuan Tu, Shun-Yao Yang, Yi-Lung Tsai, and Yen-Chiu Li assisted in the software development of EZ-Gate.

REFERENCES

- [1] "Telosb datasheet," http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Data%20sheet.pdf.
- [2] M. Beigl, A. Krohn, T. Riedel, T. Zimmer, C. Decker, and M. Isomura, "The uPart experience: Building a wireless sensor network," in *Proceedings of the ACM Conference on Information Processing in Sensor Networks (IPSN)*, 2006.
- [3] C. Park and P. H. Chou, "Eco: Ultra-wearable and expandable wireless sensor platform," in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*, Apr. 2006.
- [4] G. Mulligan, "The 6LoWPAN architecture," in *EmNets '07: Proceedings of the 4th workshop on Embedded networked sensors*. New York, NY, USA: ACM, 2007, pp. 78–82.
- [5] G. H. Cooper, "TinyTCP," <http://www.pdos.lcs.mit.edu/roofnet>, Oct. 2002.
- [6] S. Shon, "Protocol implementations for web based control systems," *International Journal of Control, Automation, and Systems*, vol. 3, pp. 122–129, Mar. 2005.
- [7] A. Dunkels, "Full TCP/IP for 8-Bit architectures," in *Proceedings of the 1st international conference on Mobile systems, applications and services*, 2003, pp. 85 – 98.
- [8] A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kullathumani, H. Zhang, H. Cao, M. Sridharan, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, C. Zhang, M. Nesterenko, R. Shah, S. Kulkarni, M. Aramugam, L. Wan, M. Gouda, Y.-R. Choi, D. Culler, P. Dutta, C. Sharp, G. Tolle, M. Grimmer, B. Ferriera, and K. Parker, "ExScal: Elements of an extreme scale wireless sensor network," in *Proceedings of the IEEE International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA)*, Aug. 2005.
- [9] B. G. Richard Guy, J. Hicks, R. Kapur, N. Ramanathan, T. Schoellhammer, T. Stathopoulos, K. Weeks, K. Chang, L. Girod, and D. Estrin, "Experiences with the extensible sensing system ESS," Tech. Rep., Mar. 2006.
- [10] R. Szweczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2004.
- [11] S. Rhee, D. Seetharam, and S. Liu, "Techniques for minimizing power consumption in low data-rate wireless sensor networks," in *Proceedings of the IEEE Wireless Communication and Networking Conference(WCNC)*, Mar. 2004.
- [12] V. A. Kottapalli, A. S. Kiremidjian, J. P. Lynch, E. Carryer, T. W. Kenny, K. H. Law, and Y. Lei, "Two-tiered wireless sensor network architecture for structural health monitoring," in *Proceedings on SPIE - Smart Structures and Materials: Smart Systems and Nondestructive Evaluation for Civil Infrastructures*, Aug. 2003.
- [13] O. Gnawali, B. Greenstein, K.-Y. Jang, A. Joki, J. Paek, M. Vieira, D. Estrin, R. Govindan, and E. Kohler, "The TENET architecture for tiered sensor networks," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2006.
- [14] A. Dunkels, F. Osterlind, and Z. He, "An adaptive communication architecture for wireless sensor networks," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2007.
- [15] M. George, L. Kallidukil, and J.-M. Chung, "Bluetooth handover control for roaming system applications," in *The 2002 45th Midwest Symposium Circuits and Systems, 2002(MWSCAS-2002)*, vol. 1, 2002, pp. 404–407.
- [16] A. Kansal and U. Desai, "Mobility support for Bluetooth public access," in *Proc. IEEE International Symposium on Circuits and Systems, 2002 (ISCAS 2002)*, vol. 5, 2002, pp. V-275–V-728.
- [17] T.-S. Lee, C.-Y. Ke, N.-Y. Ko, C.-H. Hsueh, W.-H. Chen, C.-H. Lee, and P. H. Chou, "EZ-Gate: Design and implementation of a high-performance, Linux gateway for ZigBee-compliant sensor networks," National Tsing Hua University, Tech. Rep., 2007.
- [18] C. Chen and P. H. Chou, "EcoDAQ: A case study of a densely distributed real-time system for high data rate wireless data acquisition," in *Proc. 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2008.
- [19] Tektronix, http://www2.tek.com/cmsreplive/psrep/13295/3GW_19558_0_2008.06.30.11.07%42_13295_EN.pdf.
- [20] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (Sensys)*, 2004.
- [21] K. Klues, G. Hackmann, O. Chipara, and C. Lu, "A component-based architecture for power-efficient media access control in wireless sensor networks," in *Proceedings of the 5th international conference on Embedded networked sensor systems (Sensys)*, 2007.