A Cooperative Clustering Protocol for Energy Saving of Mobile Devices with WLAN and Bluetooth Interfaces

Jong-Woon Yoo, *Member*, *IEEE*, and Kyu Ho Park, *Member*, *IEEE*

Abstract—One of the most widely used wireless communication standards is a Wireless Local Area Network (WLAN) (IEEE 802.11). However, WLAN has a serious power consumption problem. In this paper, we propose a novel energy saving approach that exploits the multiradio feature of recent mobile devices equipped with WLAN and Bluetooth interfaces. Unlike previous approaches, our work is based on clustering. In our work, a cluster is a Bluetooth Personal Area Network (PAN), which consists of one cluster head and several regular nodes. The cluster head acts as a gateway between the PAN and the WLAN, enabling the regular nodes to access the WLAN infrastructure via low-power Bluetooth. We present a distributed clustering protocol, Cooperative Networking protocol (CONET), which dynamically reforms clusters according to each node's bandwidth requirement, energy use, and application type. CONET does not require modifications of existing wireless infrastructures because clustering is performed independently of WLAN access points. We implemented the CONET prototype with four wearable computing devices to evaluate the performance on real hardware. We also simulated CONET for large networks of more than 100 mobile nodes. Both results demonstrate that our approach is effective in reducing the power consumption of WLAN.

Index Terms—Wireless communication, protocol architecture, multiradio, energy efficiency, clustering.

1 Introduction

Wireless local area network (WLAN) [23], or IEEE 802.11, has created a wave of popular interest because of its sufficient bandwidth and well-constructed infrastructures. However, a serious problem of WLAN is its considerable energy consumption, energy consumed by WLAN interfaces accounts for more than 50 percent of the total energy consumption in hand-held devices and up to 10 percent in laptops [6], [14]. Because mobile devices are usually driven by limited battery power, it is essential to devise novel solutions to reduce the power consumption due to the WLAN interface without degrading its performance.

About 70 percent of smart phones in the market have a Bluetooth interface as a secondary radio for personal area networking [22]. The Bluetooth standard is primarily designed for low-power consumption, requiring only about a 10th of the WLAN power [6]. However, because of its limited power, Bluetooth supports a low bandwidth of only 2 Mbps (version 2.0 + EDR), with a short range of 10 meters (class 2). In this work, we explore the idea of using this coexistence of high-power/high-bandwidth WLAN and low-power/low-bandwidth Bluetooth in a single mobile platform to solve the power consumption problem in WLAN-based communication systems.

Several previous works have exploited Bluetooth as a secondary radio to reduce the overall power consumption [4], [6], [7]. Bluetooth is mainly used to provide always a connected channel between mobile devices and the WLAN access point (AP). In On Demand Paging [4] and Wake on Wireless [1], mobile devices and the AP exchange control messages, e.g., wake-up messages, via low-power channels.[1] This allows a mobile device to turn off the WLAN interface when it is not being used. CoolSpots [6] and SwitchR [7] use Bluetooth more actively to lengthen the power-off time of WLAN: Bluetooth is used not only for the wake-up channel, but also for data communication when applications demand low data rates. WLAN is powered up only when the data rate reaches the Bluetooth limit. However, these approaches usually assume that APs also have both WLAN and Bluetooth interfaces (and specialized software to control them). This assumption guides the hardware and software modifications to our wireless infrastructures.

Unlike these previous works, our approach is based on clustering. Clustering is commonly used in sensor networks for network scalability [28], [29], load balancing [13], [30], data aggregation [27], or energy efficiency [9], [11], [13]. In our work, clustering makes nodes (i.e., mobile devices) that share their WLAN interfaces with each other. Fig. 1 depicts the concept of our approach and compares it to the previous approaches. As shown in Fig. 1b, a cluster is a Bluetooth Personal Area Network (PAN) [31] that consists of one cluster head (CH) and several regular nodes (RNs). CHs are responsible for coordination among the nodes within their clusters and the forwarding of packets from the PANs (clusters) to the WLAN, and vice-versa. CHs keep their

• *The authors are with the Computer Engineering Research Laboratory, Department of Electrical Engineering, KAIST, 355 Gwahangno, Yuseong-gu, Daejeon 305-701, Republic of Korea.*
E-mail: jwyoo@core.kaist.ac.kr, kpark@ee.kaist.ac.kr.

Manuscript received 17 June 2009; revised 15 Jan. 2010; accepted 31 Mar. 2010; published online 27 Aug. 2010.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-06-0241.
Digital Object Identifier no. 10.1109/TMC.2010.161.

1. Wake on Wireless uses customized low-power/low-bandwidth RF modules instead of Bluetooth.

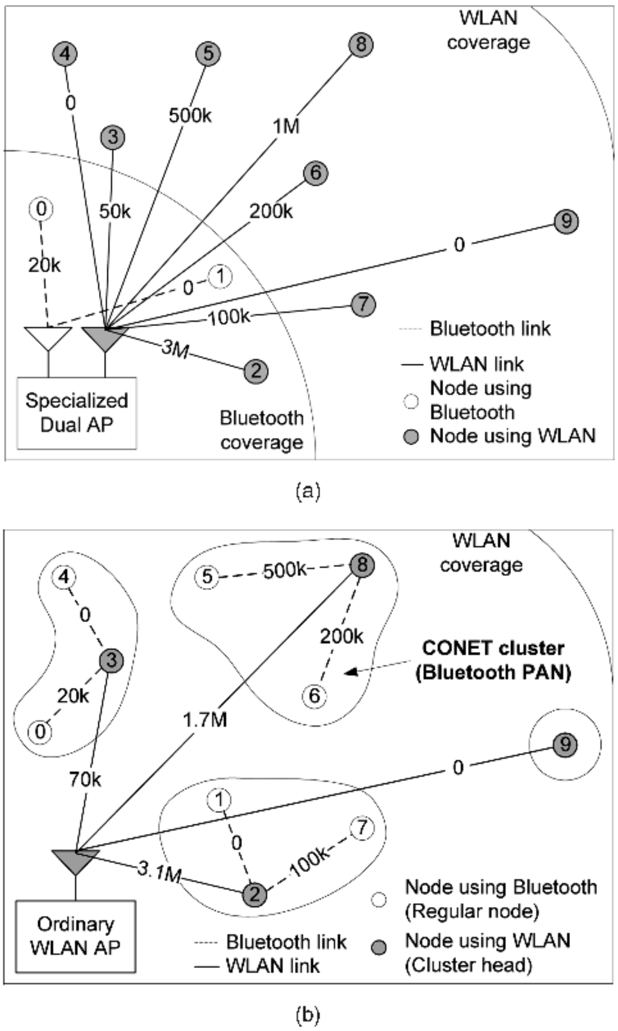

Fig. 1. A comparison of the previous approaches and our approach for the same network of 10 nodes. The numbers of the links (straight lines) represent data rates in bits per second. 0 indicates that the node is in the idle listening state. (a) Previous approaches based on the dual AP. Only 2 (node 0 and 1) of 10 nodes can communicate via Bluetooth. The others should keep their WLAN interfaces on due to the limited coverage (from nodes 3 to 9) and bandwidth (node 2) of Bluetooth. (b) Our approach based on clustering. Ten nodes are grouped into four clusters, and each cluster meets the bandwidth requirements of all nodes. Six regular nodes can save energy.

WLAN interfaces on to provide links to the WLAN AP, allowing RNs to use only Bluetooth and turn their WLAN interfaces off in order to save energy. Clustering is periodically performed in a distributed manner based on the energy uses and bandwidth requirements of the nodes.

In this work, clustering is performed independently of WLAN APs. Therefore, our approach does not require modifications to existing infrastructures (i.e., ordinary APs can be used), while the previous approaches require specialized APs with dual radios. Moreover, we solved the scalability problem of the previous works, as shown in Fig. 1a. Because of the large difference between the communication ranges of WLAN and Bluetooth, only a few devices close to the dual AP can use the low-power radio. In our case, on the other hand, since clusters can be created anywhere, most devices can obtain the benefit of energy saving, as shown in Fig. 1b.

One unique requirement which distinguishes our approach from the traditional clustering problem in sensor networks is that, unlike sensor nodes which are left unattended after deployment, mobile devices (e.g., PDAs) are arbitrarily controlled by their users. This necessitates the consideration of node mobility as well as a large variance of bandwidth requirements of various applications. Moreover, because all devices have equal significance, rotating the CH role among all devices is necessary to distribute energy consumption. Mobile devices also can be turned off at any time and powered again depending on the users' needs, which necessitates the consideration of unexpected link failures.

This paper presents a distributed clustering protocol, Cooperative Networking protocol (CONET). CONET has four main objectives:

1. improving the energy efficiency of wireless networks by exploiting a secondary radio,
2. dynamically configuring clusters to meet the bandwidth requirements of all nodes,
3. producing well-distributed cluster heads, and
4. minimizing control overhead.

CONET dynamically clusters the network according to each node's bandwidth, energy, and application type. We have implemented the CONET prototype using wearable computers [20] to evaluate its performance on real hardware systems. We also simulate CONET for large networks of more than 100 mobile nodes and evaluate the performance. Both results demonstrate that CONET is effective in reducing the power consumption of WLAN-based communication systems.

The remainder of this paper is organized as follows: Section 2 describes the problem. Section 3 presents the CONET protocol, and Section 4 discusses the issues in implementation. Section 5 shows the effectiveness of CONET via real hardware evaluation and simulations. Section 6 briefly surveys related works. Finally, Section 7 gives concluding remarks and directions for future research.

2 Problem Statement

The mobile devices that we consider in this paper are popular user terminals, such as smart phones or wearable computers. For the rest of this paper, we simply refer to a mobile device as a *node*. We assume the following properties about the nodes and wireless networks:

1. Each node has one WLAN interface (primary) and one Bluetooth interface (secondary).
2. There is at least one WLAN access point in the field. Each node can communicate with the access point using its WLAN interface, regardless of its location and time.
3. The WLAN access points do not have Bluetooth interfaces. This is typical for most existing wireless environments. Therefore, the previous approaches [6], [4], [1] are inapplicable.
4. Each node i knows the total bandwidth required, $NeedBW_i(t)$, and the free bandwidth of its Bluetooth link, $FreeBW_i(t)$.
5. Each node i can measure its residual energy $E_i(t)$.
6. All Bluetooth interfaces have the same communication range.

The final goal of our CONET is to reduce the power consumption in wireless networking applications. For this purpose, we first classify popular applications into two types: *group networking* and *individual networking*. Next, we propose a general clustering protocol that considers both application types.

- **Group networking.** In this case, a group of nodes have a common goal and need to prolong the *group lifetime* to achieve that goal. The group lifetime can be defined as the time elapsed until the first node in the group depletes its battery. For example, let us assume that some friends are playing network games together using their nodes. In this case, the maximum time during which they can play together will depend on the node with the lowest remaining battery. CONET can be applied here to prolong the group lifetime: the users can play for more time by rotating the CH role and letting nodes with lower energies be RNs and nodes with higher energies be CHs for most of the time.
- **Individual networking.** In this case, we consider unrelated individuals running their own applications (i.e., no common goal), even when they are geometrically close to each other. In a subway train, for example, many people may use their mobile nodes simultaneously, but each of them is likely to have different purposes: one may visit websites or one may just wait for incoming messages while keeping her wireless interface on. Cooperative clustering can also be applied to this case for energy conservation. An important requirement in this case is that the benefit of each node gained by cooperation should be equal for all individuals because, unlike group networking, no one will want to spend more energy just to help unrelated users. Therefore, CONET should distribute the advantages of clustering in a fair way for these types of applications.

Our goal is to design a general clustering protocol that satisfies the requirements of the above application types. To accomplish this, we separate cost functions from the clustering algorithm and provide two cost functions for each of application type. Users can select proper cost functions for their applications. Depending on the selected cost function, a different set of nodes is selected as cluster heads to meet the user requirements. Also, the following requirements must be met:

1. Clustering should be completely distributed. Each node independently makes its decisions based only on local information.
2. For each cluster c_j, the sum of bandwidth requirements of all regular nodes within the cluster must not exceed the maximum data rate of Bluetooth R^B, i.e., $\sum_{RN_k \in c_j} NeedBW_k(t) \leq AR^B$, where RN_k is the regular node of ID k and $NeedBW_k(t)$ is the required bandwidth for node k at time t.
3. At the end of the clustering process, each node should be either a cluster head or a regular node that belongs to exactly one cluster.
4. Clustering should be efficient in terms of processing complexity and message exchange.

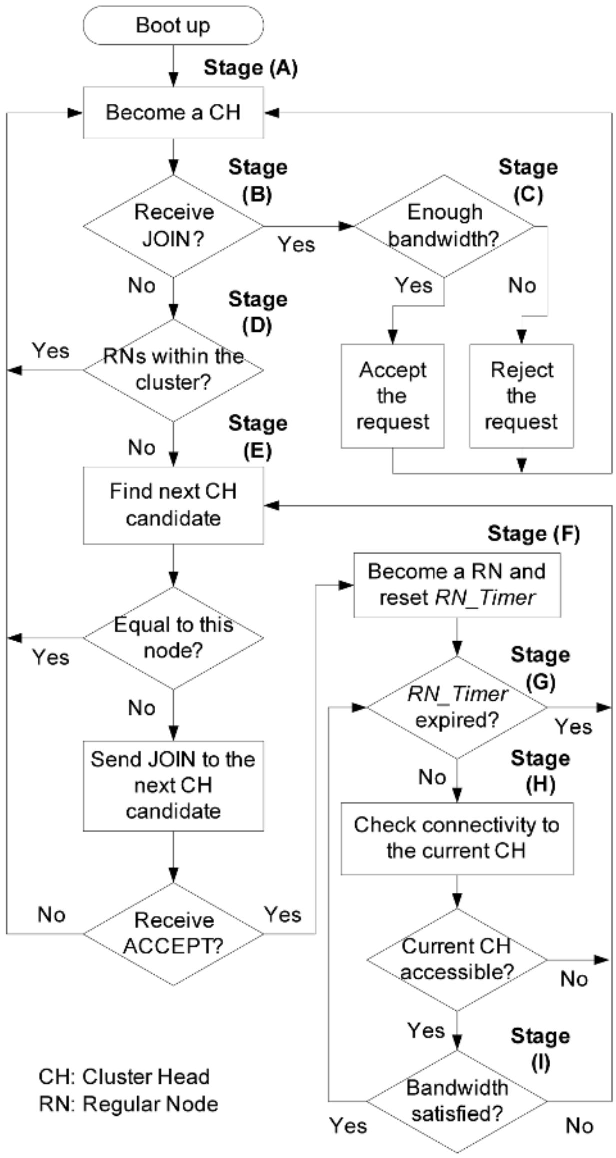

Fig. 2. Flow chart of CONET clustering protocol.

3 The CONET Protocol

This section describes CONET in detail. First, we present the protocol design. Next, we define the parameters and cost functions.

3.1 Protocol Operation

Fig. 2 shows the details of our protocol. Nodes exchange clustering messages via Bluetooth. For easy understanding, we describe our protocol based on the example shown in Fig. 3 with a group networking scenario: nodes 1, 2, and 3 have a common collaborative task and attempt to maximize the group lifetime.

3.1.1 *Cluster Head Advertisement*

When a node is newly booted up, it becomes a CH, as shown in the flow chart (Fig. 2). Assume that all three nodes of the example shown in Fig. 3 are booted up at the same time. Then, since all of them independently become CHs, three clusters are created (Fig. 3a). However, the only

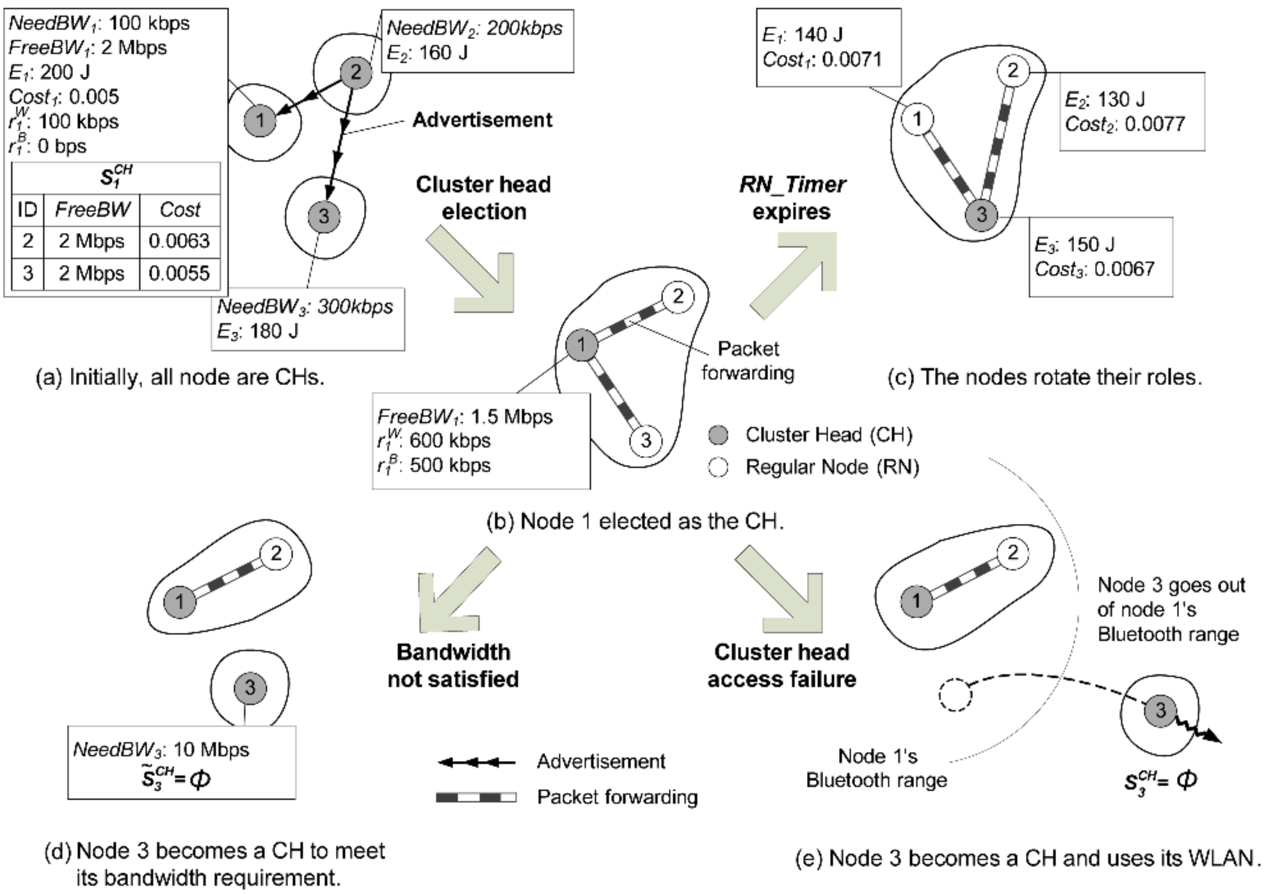

Fig. 3. An illustrative example of CONET protocol operation.

member of each cluster is the cluster head itself. Like these clusters, a cluster which has no RNs is called a *trivial cluster*, and the head of the trivial cluster is called a *trivial cluster head* (*tCH*). Therefore, tCHs do not need to use Bluetooth for packet forwarding, but only for advertising. Note that tCH is a subset of CH.

When a node becomes a CH, it starts to advertise its resource information periodically (say, every 200 msec) via Bluetooth. It repeats advertising as long as it is a CH. The advertisement message of node i contains the clustering cost C_i, the amount of bandwidth available for packet forwarding $FreeBW_i$, and some required information, such as the ID and the network address. Each node manages a set S_i^{CH}, which stores the information advertised by neighboring CHs.

Because CHs act as gateways that connect Bluetooth nodes (RNs) to the WLAN access point, $FreeBW_i$ of CH i should be the smaller value between $FreeBW_i^W$ and $FreeBW_i^B$, the amount of free bandwidth on Bluetooth and WLAN links, respectively. To estimate the free bandwidth on a wireless link, we can use well-studied bandwidth estimation techniques [17], [18], [37]. For example, we can estimate the free bandwidth using the idle channel time as proposed in [17]. A channel is considered to be idle if the node is not sending or receiving data through the channel and a carrier or interference signal is not sensed on that channel. By monitoring the idle channel times of WLAN and Bluetooth channels (T^{W_I} and T^{B_I}, respectively) during a period of time T, each node i can estimate its $FreeBW_i^W$ and $FreeBW_i^B$ using a moving average with weight $\alpha \in [0, 1]$ as follows:

$$FreeBW_i^W = \alpha FreeBW_i^W + (1-\alpha)\frac{T^{W_I}}{T}R^W - R^{mar}, \quad (1)$$

$$FreeBW_i^B = \alpha FreeBW_i^B + (1-\alpha)\frac{T^{B_I}}{T}R^B - R^{mar}, \quad (2)$$

where R^W and R^B are the transmission rates of WLAN and Bluetooth, respectively. R^{mar} is a predefined constant used to maintain the free bandwidths to be slightly lower than the bandwidth actually available. It is necessary to switch between radio interfaces dynamically based on the current data rate. Later, in this section, we explain the details of interface switching. Our current design assumes that R^W is a predefined constant, but the IEEE 802.11 standard [23] provides multiple transmission rates depending on Signal-to-Noise Ratio (SNR). We plan to improve CONET to support multiple rates in our future research.

Although only CHs advertise their resources, RNs also measure free bandwidth for cluster head election (discussed later). However, because RNs use only Bluetooth, their $FreeBW^W$ values are always zero, and thus, meaningless. Therefore, the free bandwidth of RNs should be set to $FreeBW^B$. In summary, the free bandwidth of node i, $FreeBW_i$, can be obtained as follows:

$$FreeBW_i = \begin{cases} MIN(FreeBW_i^W, FreeBW_i^B), \\ \qquad\qquad \text{if node } i \text{ is a CH,} \\ FreeBW_i^B, \quad \text{otherwise.} \end{cases} \quad (3)$$

Note that CONET does not limit the bandwidth estimation technique to [17]. It can also operate with other techniques [18], [37] with minor modifications.

3.1.2 Responding to JOIN Requests

In stage (B) of Fig. 2, each CH waits for JOIN requests from other nodes for a short time (say, 1 second). The JOIN message of node i includes the amount of required bandwidth $NeedBW_i$. Upon receiving a JOIN message, the CH goes to stage (C) and compares its *FreeBW* with the sender's *NeedBW*. If the CH has a sufficient amount of free bandwidth for the sender (i.e., $FreeBW \geq NeedBW$), it will accept the request, but, otherwise, reject it. After responding to the request, the CH returns to the initial stage. The sentence, "Become a CH," in stage (A) means "keep the CH role" for the nodes that are already CHs. At the initial moment shown in Fig. 3a, because no node has sent a JOIN message yet, all the nodes go down to stage (D).

3.1.3 Cluster Head Election

When there is no JOIN request, the CH counts the number of RNs within its clusters (stage (D) in Fig. 2). If there is at least one RN in the cluster, the CH returns to the first step and keeps its current role. This allows RNs to select their next CHs by themselves, which is necessary for network stability: If CHs stop their roles of packet forwarding regardless of the associated RNs, the RNs will occasionally lose their links to the WLAN access point. Furthermore, clusters will be reformed quite frequently if CHs ignore the status of each RN, such as the first association time (the time at which the RN has joined).

The chance for energy saving is given to trivial CHs (tCHs), which turned out to have no RNs within their clusters at the end of stage (D). A tCH selects its next CH by itself. In stage (E), each tCH calls the FIND_NEXT_CH procedure, which presents the CH election process of CONET. Assume that node i calls FIND_NEXT_CH. It then executes the following procedure:

FIND_NEXT_CH:

1) Prune the node which has insufficient bandwidth. Let S_i^{CH} and $\tilde{S}_i^{CH}$ be the original and pruned neighboring CH set of node i. For each node $k \in S_i^{CH}$, if $MIN(FreeBW_i, FreeBW_k) \geq NeedBW_i$, then copy k into $\tilde{S}_i^{CH}$.
2) Find i's next CH candidate which has the lowest cost among i and all nodes in $\tilde{S}_i^{CH}$ (node IDs for tie-breaking).
3) Return the selected node.

For example, let us assume that node 1, which is a trivial CH at the moment depicted in Fig. 3a, reaches stage (E) and calls the FIND_NEXT_CH procedure. First, node 1 prunes the CHs that cannot satisfy its bandwidth demand $NeedBW_1$ from election. If the node runs constant bit rate applications, such as VoIP, $NeedBW_1$ can be determined explicitly. However, in general TCP-based applications, it is not trivial to predict the amount of needed bandwidth in advance because TCP will match its sending rate to the available capacity. In this case, we set $NeedBW_i$ to the node's current data rate (r_i^W if the node is a CH, r_i^B otherwise). The node will become an RN (switch to Bluetooth) only if the free bandwidth on the link to the CH is higher than its current data rate.

Even though all nodes estimate $FreeBW$ using (3), the estimation results of two neighboring nodes could be different due to the limited radio range [17]. For example, let us assume that there is a hidden flow on the left side of node 1 in Fig. 3a, which is in node 1's radio range, but out of nodes 2's radio range. In this case, $FreeBW_1$ will be estimated to be smaller than $FreeBW_2$ because the flow only interferes the idle channel time of node 1. Therefore, the maximum bandwidth between nodes 1 and 2 is bounded by the smaller value $FreeBW_1$. This indicates that the free bandwidth on the link between nodes i and k should be the minimum value between $FreeBW_i$ and $FreeBW_k$, i.e., $MIN(FreeBW_i, FreeBW_k)$. In this example shown in Fig. 3a, because there is no hidden flow and all nodes have equal available bandwidths of 2 Mbps, neither node 2 nor 3 is pruned when node 1 calls the FIND_NEXT_CH procedure. Therefore, node 1's pruned CH set $\tilde{S}_1^{CH}$ is exactly the same as the original CH set S_1^{CH}.

Next, node 1 selects the lowest cost node among the nodes in $\tilde{S}_1^{CH}$ and itself. For simple explanation, let us assume that the cost of each node i, C_i, is simply the reciprocal of its residual energy E_i, i.e., $C_i = 1/E_i$.[2] The purpose of this cost function is to select the node with the highest residual energy as the CH for other low-energy nodes. Because node 1 has the lowest cost in the case of Fig. 3a, it returns to stage (A) and repeats the above processes. Similarly, nodes 2 and 3 elect node 1 as their CH and send JOIN messages to it. As their requests are accepted by node 1, they go to stage (F) and become the RNs of node 1. Finally, nodes 1, 2, and 3 are clustered together, as shown in Fig. 3b.

3.1.4 Role Switching

It is necessary to rotate the CH role regularly to balance the energy consumption among all nodes. To do so, each RN has a timer, *RN_Timer*, which expires every T_{RN} seconds. When the timer expires, the RN goes to stage (E) again and calls the FIND_NEXT_CH procedure to elect its next CH. Depending on the election result, the RN itself becomes a new CH or joins one of the existing clusters, including its current cluster.

The transition from Figs. 3b and 3c shows the cluster reformation due to the timer expiration. Let us assume that since the first cluster is created (Fig. 3b), the RNs, nodes 2 and 3, have consumed 30 Joules, while the CH, node 1, has consumed 60 Joules, regardless of their data rates.[3] As *RN_Timer* independently expires in the RNs, they move from stage (G) to stage (E) in Fig. 2. Then, node 3 finds out that it has the lowest cost among the nodes. Thus, it becomes a CH and node 2 joins the new cluster. At this

2. The specific design of the cost function for group networking is given in Section 3.2.

3. In our experiments, energy consumption of an RN is about 50 percent of that of a CH. Our results in Section 5.1.2 show that the effect of data rates on energy consumption is negligible.

moment, node 1 eventually becomes a trivial CH because no node is associated to it. Then, node 1 goes to stage (E) and finds out that node 3 is the lowest cost CH. Thus, it joins node 3, resulting the new cluster structure, as shown in Fig. 3c. By regularly switching roles in this manner, energy consumption can be distributed.

3.1.5 Recovering Cluster Head Failures

Due to the mobility of nodes, if the distance between an RN and its CH becomes longer than the Bluetooth range, the RN will not be able to access its CH anymore. The same situation happens when users intentionally turn off Bluetooth or WLAN (or both) interfaces or shut down their nodes. In these cases, the RN immediately goes to stage (E) in Fig. 2 to find its new CH. To do so, each RN continuously checks the connectivity to its current CH in stage (H).

For example, as shown in Fig. 3e, as node 3 goes out of the Bluetooth communication range of its current CH (node 1), it loses the connection to the CH. As soon as it detects the CH failure in stage (H), it goes to stage (E) and calls the FIND_NEXT_CH procedure. In the case of Fig. 3e, because node 3 has no neighboring CHs (i.e., $S_3^{CH} = \Phi$), it becomes a CH and continues its previous communication using its WLAN interface.

3.1.6 Satisfying Bandwidth Requirements

Unlike the tiny nodes in sensor networks, the nodes considered in this work have a large variety of applications, resulting in time-varying bandwidth requirements with huge variations. Therefore, nodes should selectively use either Bluetooth or WLAN depending on the requirements.

To do so, each RN i associated to CH k monitors the amount of free bandwidth on the link between nodes i and k, i.e., $MIN(FreeBW_i, FreeBW_k)$, and the current data rate r_i^B. When $NeedBW_i$ is known explicitly (i.e., CBR applications), the RN directly compares the current data rate with $NeedBW_i$ to check whether its bandwidth requirement is satisfied or not. If $r_i^B < NeedBW_i$, then it goes to stage (E) of Fig. 2 to find a new CH which can meet its bandwidth demand.

In general TCP applications, the amount of bandwidth needed is not determined explicitly. In this case, node i assumes that its $NeedBW_i$ is equal to r_i^B and controls radio interfaces to satisfy r_i^B. Similar to [19], the free bandwidth is estimated to be slightly lower than the actually available bandwidth ((1)-(3)) so that the value of $MIN(FreeBW_i, FreeBW_k)$ will cross zero and become negative as r_i^B reaches the capacity limit. In this case, i.e., $MIN(FreeBW_i, FreeBW_k) < 0$, the RN goes to stage (E) of Fig. 2 to find a new CH. Otherwise, it returns to stage (G) and repeats the above processes.

This situation is illustrated in Fig. 3d: as $NeedBW_3$ increases to 10 Mbps, node 3 finds out that the current CH (node 1) cannot meet this requirement (stage (I) of Fig. 2). Thus, node 3 goes to stage (E) to find a new CH, but no node can satisfy the requirement of 10 Mbps. Therefore, its pruned CH set $\tilde{S}_3^{CH}$ becomes Φ. Finally, node 3 becomes a new CH, as shown in Fig. 3d.

3.2 Application Types and Cost Functions

In this section, we present two cost functions designed for group networking and individual networking.

3.2.1 Group Networking

As described in Section 2, the main objective of group networking is to prolong the group lifetime. In sensor networks, one popular cost function used to maximize the network lifetime is primarily based on the residual energy of each node [11], [13], e.g., *(maximum energy)/(residual energy)*. This cost function distributes energy dissipation over the network particularly well when the power consumption rates are equal for all nodes. In CONET, however, a variety of nodes types made by different venders are clustered together, breaking the homogeneity of the power consumption rate. Therefore, our cost function for the group networking case is based on each node's *estimated lifetime*, the estimated time for a node to survive in the future. We define the cost of being a CH for node i at time t, $C_i(t)$, as follows:

$$C_i(t) = \frac{1}{\tilde{L}_i(t)}, \tag{4}$$

where $\tilde{L}_i(t)$ is node i's estimated lifetime. We assume that each node i knows its current power consumption $P_i(t)$ and residual energy $E_i(t)$. Then, the lifetime estimation is based on the moving average of the current and past power usage with weight $\beta \in [0, 1]$ [33], i.e.,

$$\tilde{P}_i(t) = (1 - \beta)P_i(t) + \beta P_i(t - T_s), \tag{5}$$

where $\tilde{P}_i(t)$ represents the *future* power consumption estimated at time t. Once $\tilde{P}_i(t)$ has been estimated, the node can calculate its $\tilde{L}_i(t)$ as follows:

$$\tilde{L}_i(t) = \frac{E_i(t)}{\tilde{P}_i(t)}. \tag{6}$$

From (4) and (6), the cost of node i for group networking can be calculated as $C_i(t) = \tilde{P}_i(t)/E_i(t)$.

3.2.2 Individual Networking

For individual networking, energy saving should be as equal as possible to all cooperating nodes. This motivates us to use the *energy saving ratio* (*ESR*) as the cost for individual networking. Our goal is to equalize ESR among all cooperating nodes. Consider node i, which was booted up at $t = 0$ and has cooperated with others using CONET for $[0, t]$, $t > 0$. The role (CH or RN) of the node and cluster organization may have changed with time, depending on its resource usage. Using the cumulative amount of energy consumption, the energy saving ratio of node i at time t, $ESR_i(t)$, can be defined as follows:

$$ESR_i(t) = \frac{\tilde{E}_i^{tr}(t) - E_i^{co}(t)}{\tilde{E}_i^{tr}(t)} = 1 - \frac{E_i^{co}(t)}{\tilde{E}_i^{tr}(t)}, \tag{7}$$

where $\tilde{E}_i^{tr}(t)$ represents the expected energy that would be consumed if node i had communicated in the *traditional* WLAN-only manner (i.e., without CONET) during $[0, t]$. $E_i^{co}(t)$ represents the energy actually consumed by the node when it has used CONET. Therefore, $E_i^{co}(t)$ mainly depends on the history of the node's current and past roles: when the node is an RN, $E_i^{co}(t)$ will increase more slowly than $\tilde{E}_i^{tr}(t)$ because RNs use only Bluetooth, resulting in an increase in $ESR_i(t)$. Otherwise, when the node is a CH, $E_i^{co}(t)$ will increase as fast as (or slightly faster than) $\tilde{E}_i^{tr}(t)$; thus, $ESR_i(t)$ will decrease.

According to (7), high ESR means that by cooperating with others, the node could save more energy than the others. Therefore, every time nodes rotate their roles, CONET selects high-ESR nodes as the next CHs, allowing low-ESR nodes to become RNs for energy saving. We define the cost of being a CH for node i at time t, $C_i(t)$, as follows:

$$C_i(t) = 1 - ESR_i(t) = \frac{E_i^{co}(t)}{\tilde{E}_i^{tr}(t)}. \quad (8)$$

With appropriate role switching periods, our protocol equalizes ESR among nodes. Section 5.2.2 evaluates this property by simulation.

We estimate $\tilde{E}_i^{tr}(t)$ and $E_i^{co}(t)$ based on nodes' roles and data rates. We assume that each node i knows the instantaneous data rates of its WLAN and Bluetooth links, $r_i^W(t)$ and $r_i^B(t)$, and the parameters listed in Table 3. Then, $\tilde{E}_i^{tr}(t)$ can be estimated as follows:

$$\tilde{E}_i^{tr}(t) = \int_0^t X_i(t)\left[P_i^W\left(r_i^{loc}(t)\right) + P_i^B(0)\right]dt, \quad (9)$$

$$X_i(t) = \begin{cases} 0, & \text{If node } i \text{ is a trivial cluster head,} \\ 1, & \text{otherwise,} \end{cases} \quad (10)$$

where $P_i^W(r)$ and $P_i^B(r)$ represent the WLAN and Bluetooth power of node i for a data rate r, respectively. Both power functions can be defined as follows:

$$P_i^W(r) = P_i^{W_A} \cdot \frac{r}{R^W} + P_i^{W_I} \cdot \left(1 - \frac{r}{R^W}\right), \quad (11)$$

$$P_i^B(r) = P_i^{B_A} \cdot \frac{r}{R^B} + P_i^{B_I} \cdot \left(1 - \frac{r}{R^B}\right). \quad (12)$$

If the node had not used CONET, it would not forward other nodes' packets. Thus, the energy consumption for forwarding other nodes' packets should be eliminated from $\tilde{E}_i^{tr}(t)$ estimation. In (9), $r_i^{loc}(t)$ denotes the local data rate due to only node i's own packets, which can be approximated as follows:

$$r_i^{loc}(t) = Y_i(t)\left[r_i^W(t) - r_i^B(t)\right] + [1 - Y_i(t)]r_i^B(t), \quad (13)$$

$$Y_i(t) = \begin{cases} 0, & \text{If node } i \text{ is an RN,} \\ 1, & \text{otherwise.} \end{cases} \quad (14)$$

From (13) and (14), $r_i^{loc}(t)$ is equal to $r_i^W(t) - r_i^B(t)$ when the node is a CH. Since the CH's Bluetooth is only used to forward others' packets, we can assume that $r_i^W(t) - r_i^B(t)$ approximates to the data rate caused by CH i's local applications. When the node becomes an RN, $r_i^W(t) = 0$ and $r_i^{loc}(t) = r_i^B(t)$ because RNs use only Bluetooth. In the traditional method, local data packets always go through WLAN and Bluetooth stays idle all the time. Therefore, in (9), $r_i^{loc}(t)$ and zero are passed to P_i^W and P_i^B, respectively, for $\tilde{E}_i^{tr}(t)$ estimation.

Similarly, the actual energy consumption $E_i^{co}(t)$ can be obtained as follows:

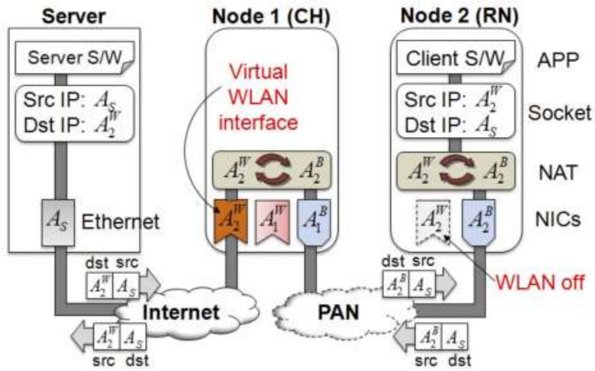

Fig. 4. A seamless handoff technique using NAT.

$$E_i^{co}(t) = \int_0^t X_i(t)\left[P_i^W\left(r_i^W(t)\right)Y_i(t) + P_i^B\left(r_i^B(t)\right)\right]dt. \quad (15)$$

$Y_i(t)$ eliminates the WLAN power consumption for RNs. Therefore, $E_i^{co}(t)$ heavily depends on the node's current and past roles. Note that due to the presence of $X_i(t)$ in (9) and (15), $ESR_i(t)$ does not change when node i is a trivial cluster head (tCH), which means that it does not cooperate with others.

4 Implementation Issues

The major concern in CONET implementation is how to handle connection handoffs seamlessly. Vertical handoffs occur as nodes changes radio interfaces from WLAN to Bluetooth, and vice versa. Horizontal handoffs happen when RN changes their cluster because handoffs generally lead to packet loss and TCP performance degradation, and efficient handoff mechanisms which have been proposed in the literature should be considered when we implement CONET.

Network Address Translation (*NAT*) [25] can be used for handoff management. NAT is the process of modifying network address fields in packet headers. In TCP, because a connection is defined by a pair of IP addresses (source/destination) and port numbers, each node's IP address and the port number must be preserved regardless of clustering operations, while the routing is ongoing. Because the address/port translation function of NAT is applicable to this purpose, various techniques using NAT for handoffs have been proposed [15]. The NAT-based handoff technique provides a simple way of implementing CONET as our protocol can be implemented only using a few network commands like *iptables*, *route*, or *arping*, which are already in the standard distribution of Linux. The NAT, however, has been reported to be unsafe for frequent changes in IP addresses [15]. Thus, secure mechanisms for the NAT [32] should be considered together in this implementation.

Fig. 4 is a two-node cluster example that shows how NAT can be used in CONET for seamless vertical handoff. For a simple explanation, we assume that each node i has two different IP addresses A_i^W and A_i^B, which are statically assigned to the WLAN and Bluetooth interfaces of node i, respectively. As node 2 joins to node 1, the physical media of node 2 are changed from the WLAN to Bluetooth. For the continuation of the previous connection of node 2 to the

Fig. 5. Appearance of UFC. A power supply is directly connected for experiments.

TABLE 2
Power Breakdown of UFC in Idle Mode

Component	Power Consumption (mW)	%
WLAN interface	880	49.4
Bluetooth interface	120	6.7
CPU	280	15.7
SDRAM	80	4.5
Other	420	23.7
Total	1780	100

Other includes flash memories, LEDs, voltage regulators, and so on.

server, the source address of its outgoing packets should be modified to A_2^B, because RNs use only Bluetooth. The NAT component in node 2 takes charge of modifying IP headers so that its outgoing packets can reach to node 1 via Bluetooth and incoming packets can be delivered to the unmodified socket. Then, these packets of node 2 go through the NAT component of node 1 again, which translates node 2's Bluetooth address to node 2's (not node 1's) WLAN address. This is necessary because the clustering operations should be transparent to hosts in the Internet. The *Virtual Interface* (also known as *IP aliasing*) technique [26] is applied so that the WLAN interface of node 1 can have multiple IP addresses A_1^W and A_2^W. With virtual interface and two-level address translation techniques, the packets of node 2 can maintain the same source/destination pair as after clustering and can arrive at the server correctly.

5 Performance Evaluation

In this section, we evaluate the performance of CONET. First, we present experimental results from the prototype that we have implemented using customized wearable computing platforms. The main purpose of the prototype evaluation is to study the energy efficiency and overheads of CONET in real hardware systems. Next, because it is too complex to test with a number of real devices (e.g., more than 100 nodes with movements), we evaluate the performance of CONET for a large-scale mobile network via extensive simulations.

5.1 Evaluation with Prototype

Ubiquitous Fashionable Computers (UFCs) [20] are used as mobile nodes. UFC is a wearable computing platform (Fig. 5) that has three kinds of wireless interfaces: WLAN, Bluetooth, and ZigBee (IEEE 802.15.4). Table 1 summarizes the

TABLE 1
Specification of the Three Wireless Interfaces of UFC

Type	ZigBee	Bluetooth	WLAN
H/W	Crossbow MICAz (CC2420)	CSR BlueCore03	3COM 3CRUSB 10075
Standard	Not used	Bluetooth 1.0	802.11b
Maximum data rate	Not used	1 Mbps (440* kbps)	11 Mbps (5.1* Mbps)

Values marked with ∗ are measured. We measured the effective maximum rate of Bluetooth and WLAN using an FTP download workload.

specifications of these interfaces. In the prototype evaluation, ZigBee is removed from the UFCs. The main processing module of UFC is based on the Intel XScale processor, PXA270, and runs Linux 2.6. We measured the power consumption of major components. The power breakdown for a UFC in idle mode is presented in Table 2. Observe that the WLAN interface consumes about 880 mW which is almost half of the total power consumption (1,780 mW).

To make each UFC aware of its residual energy, we removed its battery and directly connected the Agilent E3648A power supply to the UFC, as shown in Fig. 5. This equipment is capable of measuring the power consumption in real time and feeding back the measurement result via an RS-232 serial interface. We connected the power supply and the UFC with an RS-232 cable and wrote a simple logging program that updates the residual energy based on the reported power consumption. With this hardware/software setup, we can set the initial energy of each node to any value and measure the fine-grained power consumption and residual energy. Currently, we set β in (5) to 0.2 to estimate the future power consumption. The choice of this value is based on empirical analysis of sensitivity.

In the prototype evaluation, we used the NAT-based switching technique to manage handoffs, which is described in Section 4. All nodes were stationary and close to each other (i.e., within the Bluetooth range) during the experiments. Unless otherwise specified, we set the default role switching period T_{RN} to 120 seconds for all experiments. The effect of these time values will be discussed in Section 5.2.2.

5.1.1 Node Role and Power Consumption

To understand the effect of node roles on power consumption, we organized a two-node cluster using two UFCs. The initial energies of both nodes were set to be equal. Neither of them generated network traffic, i.e., all nodes were idle, during the experiment. We assumed a group networking scenario in this experiment: thus, the cost function of (4) is used. Since the UFCs were set to have exactly the same conditions, they just rotated their roles every 120 (T_{RN}) seconds. We measured the time-varying behavior of the power consumption of the two UFCs (UFC1 and UFC2) according to their roles.

The measurement result is shown in Fig. 6. When UFC1 was the CH, it consumed about 1.78 watts. As it became the RN, the average power consumption of UFC1 was reduced to only 0.90 watts, which is almost half of the CH power consumption. This is because the RN used only Bluetooth and turned off its WLAN interface. By switching the CH

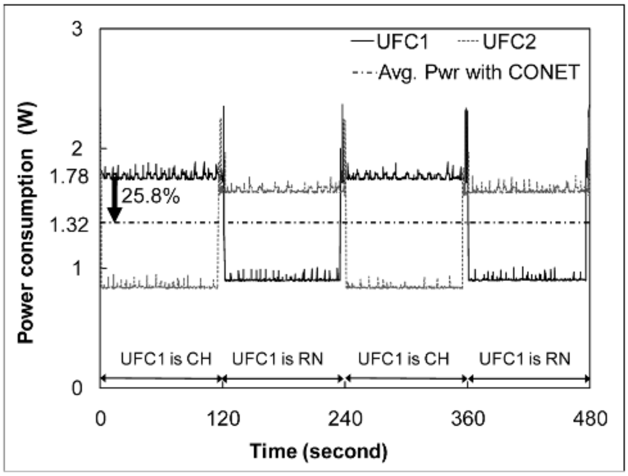

Fig. 6. Time-varying power consumption according to the node roles.

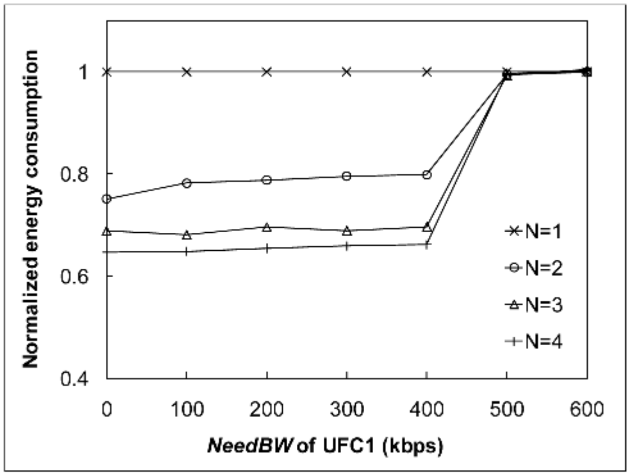

Fig. 7. Effects of the number of nodes (N) and the bandwidth requirement (*NeedBW*) on energy efficiency.

role with UFC2 every 120 seconds, UFC1 consumed about 1.32 watts on average. Compared to the traditional WLAN-only communication, both UFC1 and UFC2 can reduce energy consumption by 25 percent.

5.1.2 Number of Nodes and Data Rates

To evaluate the effects of the number of nodes (N) on the performance, we varied N from 1 to 4. All nodes are sufficiently close (i.e., within the Bluetooth range) to each other: thus, they could be grouped into the same cluster. We also investigated the effects of bandwidth requirements (*NeedBW*). A traffic generator, D-ITG [21], was ported on UFC1 and generated Poisson distributed TCP traffic of various data rates to the test server. The test server is directly connected to the AP with an Ethernet cable; thus, the data rate is not limited by any external networks. We varied the data rate of UFC1 from 0 to 600 kbps. Other UFCs stayed in the idle state during the experiments. We performed experiments on group networking scenarios with the same initial energies for all nodes. Therefore, clustered nodes rotated their roles every 120 (T_{RN}) seconds in a round-robin manner. We measured the energy consumption of UFC1 for 16 minutes for each case. The experiment results are shown in Fig. 7. Each curve was normalized to energy consumed by UFC1 when there was no other nodes with which to cooperate (i.e., N is 1).

When the number of nodes (N) is 1, UFC1 should use its WLAN interface from that point on, as it does in traditional networking. This case is a baseline for comparison. As N increases, the energy consumption of UFC1 decreases if its *NeedBW* does not exceed the bandwidth limit of Bluetooth. For example, when its *NeedBW* is 200 kbps, UFC1 consumes only 78 percent ($N = 2$), 70 percent ($N = 3$), and 65 percent ($N = 4$) of the baseline. This is because, as the number of cooperating nodes increases, UFC1 can spend more time as an RN. However, when its *NeedBW* exceeds the Bluetooth limit, UFC1 cannot be the RN because its requirement cannot be satisfied by other nodes. As presented in Table 1, the effective maximum data rate of Bluetooth is 440 kbps. Therefore, when UFC1's *NeedBW* is 500 or 600 kbps, it should always use its WLAN regardless of N, consuming as much energy as the baseline.

When UFC1's *NeedBW* is less than the effective maximum data rate, 440 kbps, energy consumption slightly increases with *NeedBW*. For example, when N is 2, the normalized energies of UFC1 with *NeedBW* of 0 and 400 kbps are 0.75 and 0.80, respectively. This is because Bluetooth is less energy-efficient than WLAN from a pure energy/bit point of view: WLAN and Bluetooth consume 100 and 210 nJ/bit (calculated with the values in Table 3), respectively. Therefore, the difference of energy consumption between CONET and the traditional WLAN-only method decreases as the data rate increases. However, this small degradation can be ignored in light of the considerable energy saving incurred by removing the idle power consumption of WLAN.

The coexistence of Bluetooth and WLAN also affects the data rates in wireless environments [38]. We plan to investigate the effect of the coexistence in future work.

5.1.3 Group Lifetime

As previously mentioned in Section 3.2.1, CONET prolongs the group lifetime for group networking. To evaluate the lifetime extension performance, we set multiple UFCs to have different initial batteries and ran CONET with the cost function of (4). We measured the group lifetime, i.e., the time when any of UFCs stops due to the battery exhaustion.

First, we performed an experiment with two UFCs, UFC1 and UFC2, whose normalized initial batteries are set to 1 and 1.5, respectively. The experimental results are shown in Fig. 8. The time values (x-axis) are normalized to the

TABLE 3
Simulation Parameters and Values

Parameters	Values	Comments
P^{W_A}	1100 mW	Active Tx/Rx power of WLAN
P^{W_I}	880 mW	Idle listening power of WLAN
P^{B_A}	220 mW	Active Tx/Rx power of Bluetooth
P^{B_I}	120 mW	Idle listening power of Bluetooth
R^W	54 Mbps	Maximum bit rate of WLAN
R^B	2 Mbps	Maximum bit rate of Bluetooth

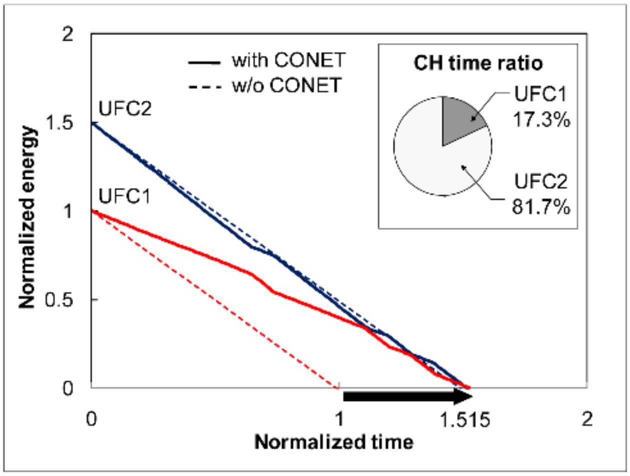

Fig. 8. Extension of the group lifetime of a two-node cluster.

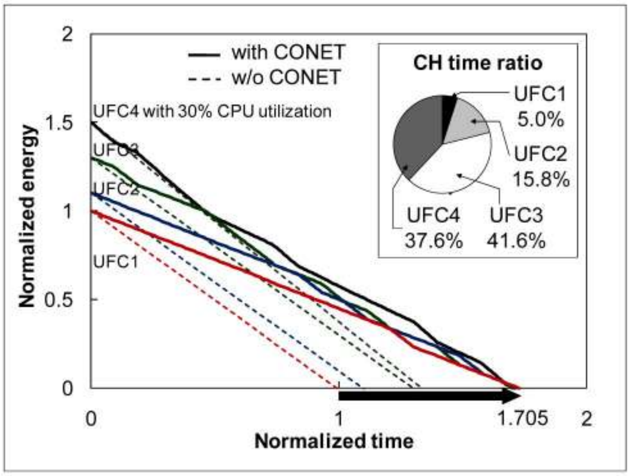

Fig. 9. Extension of the group lifetime of a four-node cluster. UFC4 has an extra CPU load.

traditional lifetime of UFC1. The result shows that CONET extends the group lifetime by about 52 percent. During the experiment, UFC2 became the CH more frequently than UFC1 because of UFC2's higher initial energy. The pie diagram shows that UFC2 was the CH for 82 percent of the total lifetime, while UFC1 for only 18 percent. It is important to note that although UFC2 spent most of the time as the CH, its lifetime was also slightly lengthened by about 1.5 percent. This is because the energy saved by turning the WLAN off for a few periods is much larger than the overhead of clustering operations, such as role switching. Section 5.1.4 discusses the overhead in detail.

Next, we increased the number of UFCs to four and performed the same experiment. An additional CPU-intensive load that requires a 30 percent CPU usage, on average, was given to UFC4, whose initial battery is the largest among the UFCs; therefore, UFC4's battery will be consumed more rapidly than others. The CPU load reflects a situation where one of the UFC users in the same group performs another task, such as watching a movie or listening to music with her UFC, while playing an online game simultaneously. Fig. 9 shows the results. Because the future lifetime of each UFC is estimated by using both the current and past power usage, even when UFC4 has the highest energy, UFC3 was frequently selected as the next CH. This experiment proves that the implemented CH election mechanism works correctly according to the algorithm described in Section 3.2.1. As a result, the group lifetime of this cluster is extended by about 70 percent.

5.1.4 Switching Overhead

When an RN becomes a CH, it should turn on its WLAN interface, load the appropriate software, such as a device driver, and modify its network settings. This sequence of jobs incurs overhead in terms of time and energy. Fig. 10 shows the overhead in role switching from RN to CH. In the ideal role switching from CH to RN, the WLAN interface is enabled immediately without any preparation processes, and the power curve rises vertically (like a step pulse). In the real case, however, several seconds of time are required to activate the WLAN interface and load the corresponding software modules. These processes also consume some energy. Fig. 10 shows that activating the WLAN interface of a UFC takes about 3.8 seconds and consumes about 3,620 mJ of additional energy. Fig. 11 shows the power consumption when a CH becomes an RN. Turning off the WLAN interface also includes time and energy overhead for disabling the WLAN interface, unloading the WLAN device driver, and modifying NAT and route configuration. For a UFC, the time to completely switch from CH to RN is about 1.4 seconds, and the amount of additional energy is about 830 mJ.

5.2 Evaluation with Simulation

In this section, we evaluate the performance of CONET via simulations. Unless otherwise specified, we assume that 100 nodes are uniformly dispersed into a 70×70 meter field. Because it is unrealistic to assume that all 100 nodes have the same purpose (like sensor networks) of prolonging the group lifetime, we only consider individual networking in this simulation. An ordinary WLAN access point is located at (0, 0). We assume that the WLAN and Bluetooth

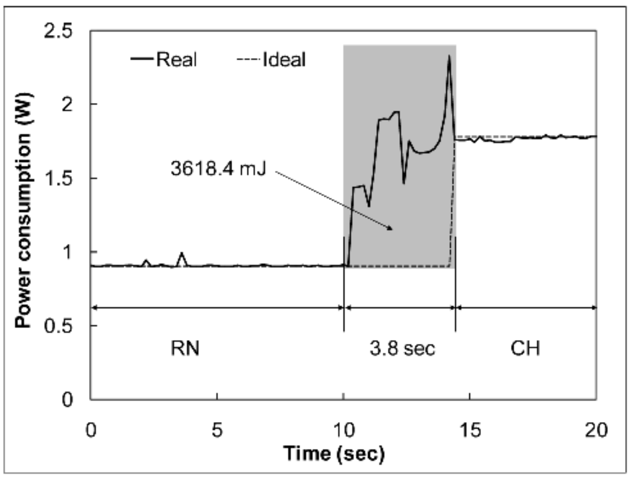

Fig. 10. Overhead of switching from RN to CH.

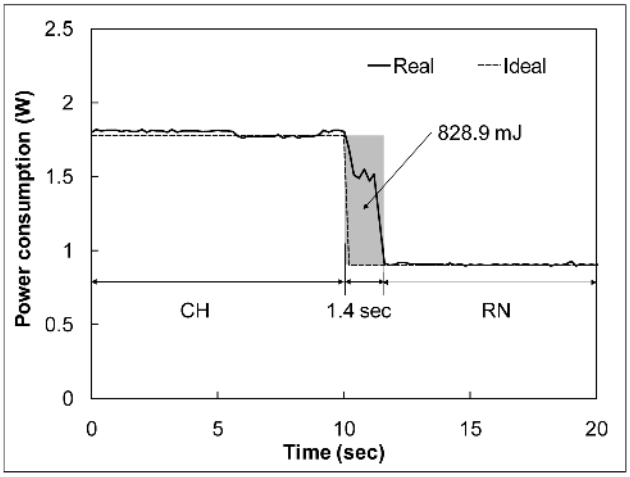

Fig. 11. Overhead of switching from CH to RN.

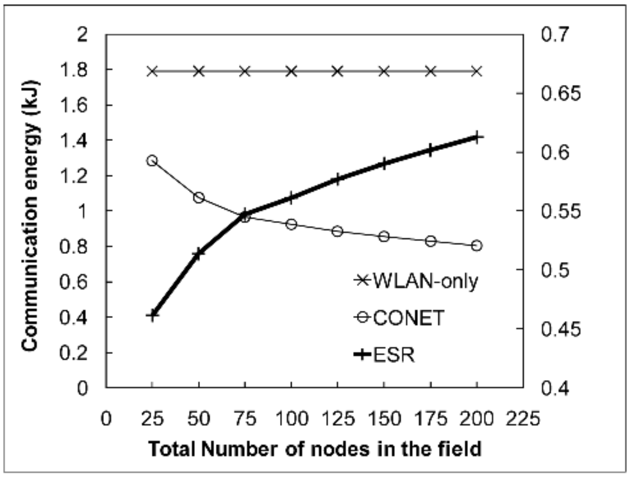

Fig. 12. Effects of the node density on the average communication energy and energy saving ratio (ESR).

communication ranges are 100 and 10 meters, respectively. Therefore, all nodes can communicate with the access point anywhere in the field using their WLAN interfaces. We do not consider the multirate support of WLAN, i.e., the maximum bit rate of WLAN is constant (54 Mbps).

The motion of the nodes follows the *Random Waypoint Movement with Pause* model [36]. In the beginning, nodes are uniformly distributed over the entire field. Then, each node randomly chooses a location as a next destination (way point). The distribution of the way points is uniform. At the same time, the node randomly picks a velocity between 0.5 and 2.0 m/s. Then, it moves to the destination with the chosen speed. After arriving at the destination, the node pauses for a random period between 30 and 600 seconds. Every node repeats the above procedure until the simulation ends.

At the beginning of each pause period, a node triggers the CBR traffic of a random data rate between 0 and 1,000 kbps. This CBR traffic lasts for a random period between 0 and 120 seconds. The node then has a random think time between 0 to 60 seconds. If the node is still in the pause period after the think time, it initiates more random CBR traffic. Note that a node can have traffic while it is moving if its last traffic is not terminated before the pause period ends. This procedure is repeated until the simulation ends. The parameter values used in the simulation are summarized in Table 3. The simulation time is 1,800 seconds for each experiment.

5.2.1 Effects of Node Density

We vary the number of nodes in the field from 25 to 200 to study how CONET works with low to high node density. Fig. 12 shows the effects of the node density on the communication energy consumption and the energy saving ratio (ESR). In traditional WLAN-only networking, each node consumes about 1.79 kJ on average, regardless of the number of nodes. In CONET, on the other hand, energy consumption decreases as the node density increases because more nodes can be grouped into a cluster and share WLAN interfaces. A node belonging to a larger cluster can spend more time as an RN.

For example, when there are 100 nodes using CONET, the average energy consumption of each node is about 0.93 kJ. This is only 52 percent of the energy consumed in WLAN-only networking, showing an energy savings of approximately 48 percent. Observe that the ESR value, which is 56 percent (0.56) for the 100 nodes case, is slightly higher than 48 percent. This is because, as discussed in Section 3.2.2, the ESR is calculated using only energy consumed during cooperation with at least one node, i.e., energy consumed when the node is a trivial CH is removed from the ESR calculation. As the number of nodes increases from 100 to 200, the ESR also increases from 0.56 to 0.61 because each node can spend more time as an RN.

Fig. 13 illustrates the relation between the node density and energy saving. It shows the cluster organizations and average cluster sizes (the average number of nodes in a cluster) according to the total number of nodes. When the number of nodes is 25, the portion of trivial CHs is about 31 percent, which means that only 69 percent of the nodes cooperate with other nodes on average. This is because when the nodes are sparsely distributed, each node is likely to find no neighboring nodes for cooperation. As a result, the

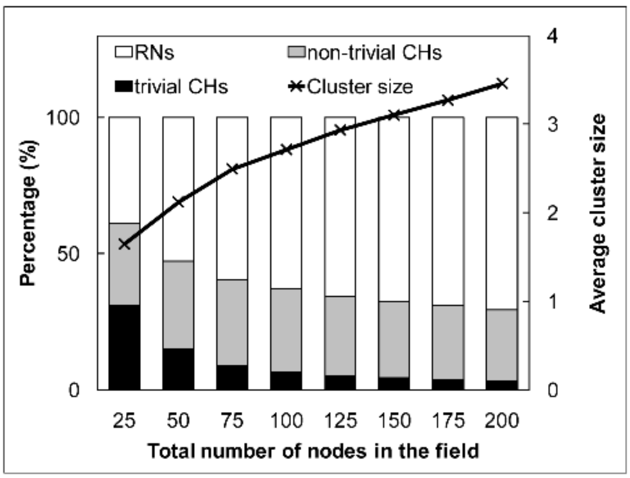

Fig. 13. Effects of the node density on the cluster organizations and the average cluster sizes.

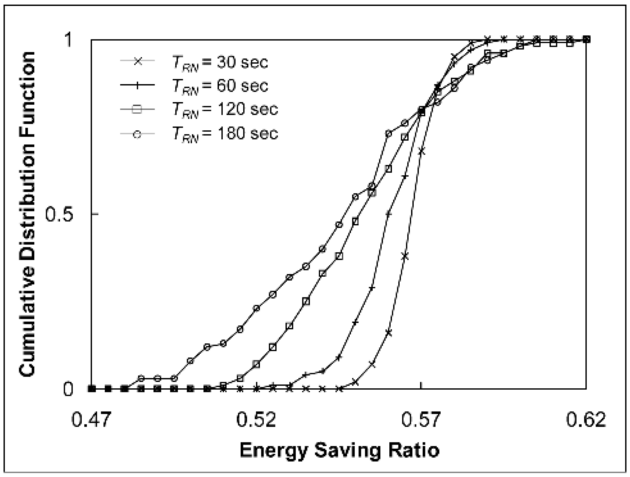

Fig. 14. Effects of the role switching period (T_{RN}) on the fairness of energy saving ratio (ESR).

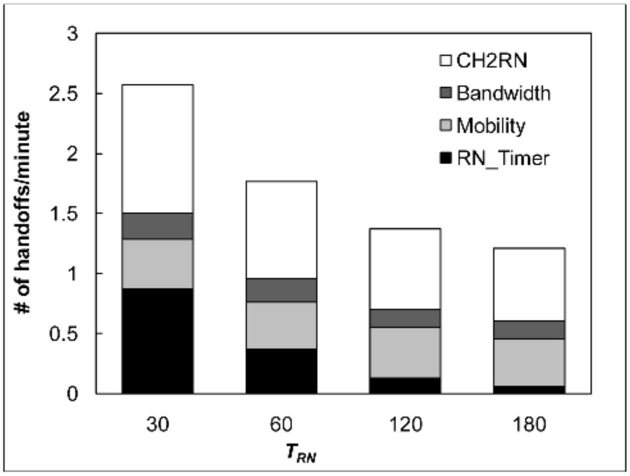

Fig. 15. Average handoff rates and the causes of handoffs with various role switching periods (T_{RN}).

average cluster size for the 25-node case is less than 2 (1.64). On the other hand, when the number of nodes is 150, the average cluster size is larger than 3 (3.10), which means that each CH has more than two RNs within its cluster, on average. Note that the average cluster size is not proportional to the total number of nodes. This is because CONET adjusts the cluster size so that the sum of the bandwidth requirements of RNs does not exceed the Bluetooth capacity.

5.2.2 Effects of Switching Period T_{RN}

As described in Section 3.2.2, providing as fair of an energy saving ratio (ESR) as possible among all nodes is important for individual networking. The fairness of ESR primarily depends on the switching period T_{RN}. We vary T_{RN} from 30 to 180 seconds to study the effects of the switching period on the fairness of ESR. The total number of nodes is fixed at 100 for all cases.

Fig. 14 shows the cumulative distribution functions of ESR with various T_{RN} values. It shows that higher frequency role switching (e.g., $T_{RN} = 30$ seconds) distributes ESR in fairer way. However, it incurs too many handoffs, which can break network stability and cause high switching overhead. On the other hand, if T_{RN} is too long (e.g., 180 seconds), the variance of ESR values increases, disrupting the ESR fairness. Therefore, T_{RN} should be decided carefully, considering both fairness and overhead. In most of our experiments, T_{RN} is set to 120 seconds by default because it results in less than a 2 percent energy overhead with relatively good fairness.

Fig. 15 shows the effects of T_{RN}, on average, handoff rates in detail. *CH2RN* means the handoffs due to the role switching from CH to RN. *Bandwidth* and *Mobility* represent the handoffs caused by bandwidth change (stage (I) in Fig. 2) and the failure on link to the current CH (stage (H)), respectively. *RN_Timer* means the handoffs due to the timer expiration. Among these reasons, *CH2RN* and *RN_Timer* are directly related to T_{RN}.

Fig. 15 shows that when T_{RN} is 30 seconds long, more than 2.5 handoffs occur per minute, and the main reasons are *CH2RN* and *RN_Timer*. As T_{RN} increases, both the overall handoff rate and the portions of *CH2RN* and *RN_Timer* also decrease, while the portions of *Bandwidth* and *Mobility* remain unchanged.

6 Related Work

Many previous studies have investigated techniques that reduce the energy consumption due to WLAN interfaces in single radio mobile devices. They optimize the power consumption at various layers, such as the application layer [33], transport/network layer [34], and MAC layer [35]. The IEEE 802.11 standards also define several low-power modes, such as PSM in the legacy 802.11 [23] and Automatic Power Save Delivery (APSD) in 802.11e [24]. They allow nodes to keep their WLAN cards in the sleep state when they do not have to communicate and switch to active state periodically (PSM) or at application-specific instants of time APSD to retrieve data buffered in the access point. Although the majority of the WLAN interface's circuitry is turned off in the sleep state, the base power consumption for the minimal host card interaction and state transition is not negligible, which is typically 200-400 mW [4], [6]. On the other hand, CONET allows RNs to completely turn off their WLAN interfaces and use only Bluetooth. Moreover, since Bluetooth also supports low-power modes, such as *sniff* mode [31], which operate in similar manner to PSM but consume an order of magnitude less power than PSM (e.g., 25 mW [6]), RNs can save more energy using Bluetooth low-power modes. Of course, CHs can operate using PSM or APSD to communicate with the access point, resulting in lower average power consumption than PSM or APSD.

Some advanced WLAN chipsets [39] dramatically reduce the idle power consumption, but require cost and time for hardware upgrade or worldwide deployment. As a result, the majority of today's hand-held products still have power consumption problem due to the WLAN interface [40], [41]. In contrast, CONET needs only a simple software patch at OS level, resulting fast deployment to existing mobile devices and infrastructures.

As mobile devices increasingly feature multiple radios, the idea of using a secondary low-power radio to reduce the power consumption of the WLAN interface has been proposed [2]. In [1], [3], a VoIP device exploits a secondary radio as a wake-up channel, but this incurs long latencies for activating the sleeping device. For general applications, several paging schemes have been proposed [4], [5], but they also contain the latency problem to activate the WLAN channel. CoolSpots [6] and SwitchR [7] use the secondary radio not only for control signaling but also for data communication. They alleviate the latency problem and save more energy by lengthening the power-off time of WLAN interfaces. However, as mentioned in Section 1, they require hardware/software modifications of existing WLAN environment for deployment. Conceptually, CoolSpots and SwitchR can be special cases of CONET: if a dual AP exists, it can be regarded as a stationary node whose cost is always zero (lowest) thus always acts as a cluster head.

7 Conclusion and Future Work

In this paper, we have presented CONET, a bandwidth-aware and energy-efficient clustering protocol for multi-radio mobile networks. CONET uses Bluetooth to reduce the power consumption of WLAN in mobile devices. It dynamically reconfigures the clusters based on the bandwidth requirements of applications to avoid the performance degradation. We have classified the applications into two cases: group networking and individual networking. CONET runs the same election algorithm for both cases, but uses different cost functions. CONET maximizes the group lifetime for the group networking case and fairly distributes the energy gain among all nodes for the individual networking case. One key feature of our approach is that it does not require modifications to existing wireless environments, paving the way to easy deployment. Although this paper describes CONET based on WLAN/Bluetooth, we believe that it can be easily extended to other interface combinations, such as WiMAX/Bluetooth.

CONET can be applied to advanced types of sensor networks in which nodes have multiple radio interfaces [10]. Although we have only provided algorithms for one-hop clustering, we can extend our protocol to support multihop clustering. This can be achieved by applying general multihop clustering algorithms, such as Max-Min D-Cluster formation [16].

References

[1] E. Shih, P. Bahl, and M.J. Sinclair, "Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices," *Proc. ACM MobiCom*, pp. 160-171, 2002.

[2] P. Bahl, A. Adya, J. Padhye, and A. Wolman, "Reconsidering Wireless Systems with Multiple Radios," *ACM SIGCOMM Computer Comm. Rev.*, vol. 34, no. 5, pp. 39-46, 2004.

[3] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta, "Wireless Wakeups Revisited: Energy Management for VoIP over Wi-Fi Smartphones," *Proc. ACM MobiSys*, pp. 179-191, 2007.

[4] Y. Agarwal, C. Schurgers, and R. Gupta, "Dynamic Power Management Using on Demand Paging for Networked Embedded Systems," *Proc. Asia South Pacific Design Automation Conf.*, vol. 2, pp. 755-759, 2005.

[5] S.M. Kim, J.W. Chong, B.H. Jung, M.S. Kang, and D.K. Sung, "Energy-Aware Communication Module Selection through ZigBee Paging for Ubiquitous Wearable Computers with Multiple Radio Interfaces," *Proc. Int'l Symp. Wireless Pervasive Computing*, pp. 37-41, 2007.

[6] T. Pering, Y. Agarwal, R. Gupta, and R. Want, "CoolSpots: Reducing the Power Consumption of Wireless Mobile Devices with Multiple Radio Interfaces," *Proc. ACM MobiSys*, pp. 220-232, 2006.

[7] Y. Agrawal, T. Pering, R. Want, and R. Gupta, "SwitchR: Reducing System Power Consumption in a Multi-Client, Multi-Radio Environment," *Proc. IEEE Int'l Symp. Wearable Computers*, pp. 99-102, 2008.

[8] M. Anand and J. Flinn, "PAN-on-Demand: Building Self-Organizing WPANs for Better Power Management," Technical Report CSE-TR-524-06, Dept. of Electrical Engineering and Computer Science, Univ. of Michigan, 2006.

[9] A. Chaman and S. Pierre, "On the Planning of Wireless Sensor Networks: Energy-Efficient Clustering under the Joint Routing and Coverage Constraint," *IEEE Trans. Mobile Computing*, vol. 8, no. 8, pp. 1077-1086, Aug. 2009.

[10] D. Lymberopoulos, N.B. Priyantha, M. Goraczko, and F. Zhao, "Towards Energy Efficient Design of Multi-Radio Platforms for Wireless Sensor Networks," *Proc. ACM Int'l Conf. Information Processing in Sensor Networks*, pp. 257-268, 2008.

[11] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 3, no. 4, pp. 366-379, Oct.-Dec. 2004.

[12] V. Kawadia and P.R. Kumar, "Power Control and Clustering in Ad Hoc Networks," *Proc. IEEE INFOCOM*, pp. 459-469, 2003.

[13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Trans. Wireless Comm.*, vol. 1, no. 4, pp. 660-670, Oct. 2002.

[14] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Proc. ACM MobiCom*, pp. 85-96, 2001.

[15] H. Nguyen, H. Morikawa, and T. Aoyama, "Personal Mesh: A Design of Flexible and Seamless Internet Access for Personal Area Network," *IEICE Trans. Comm.*, vol. E89-B, no. 4, pp. 1080-1090, 2006.

[16] A.D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Hyunh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM*, pp. 32-41, 2000.

[17] Y. Yang and R. Kravets, "Contention-Aware Admission Control for Ad Hoc Networks," *IEEE Trans. Mobile Computing*, vol. 4, no. 4, pp. 363-377, July/Aug. 2005.

[18] L. Chen and W.B. Heinzelman, "QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks," *IEEE J. Selected Areas in Comm.*, vol. 23, no. 3, pp. 561-572, Mar. 2005.

[19] S. Kim and I. Yeom, "TCP-Aware Uplink Scheduling for IEEE 802.16," *Proc. Comm. Systems Software and Middleware and Workshops*, pp. 349-355, 2008.

[20] J. Lee, S.-H. Lim, J.-W. Yoo, K.-W. Park, H.-J. Choi, and K.H. Park, "A Ubiquitous Fashionable Computer with an i-Throw Device on a Location-Based Service Environment," *Proc. IEEE Advanced Information Networking and Applications Workshops*, vol. 2, pp. 59-65, 2007.

[21] S. Avallone, S. Guadagno, D. Emma, A. Pescape, and G. Ventre, "D-ITG: Distributed Internet Traffic Generator," *Proc. IEEE Int'l Conf. Quantatative Evaluation of Systems*, pp. 316-317, 2004.

[22] GSMArea, http://www.gsmarea.com/search.php3, Dec. 2009.

[23] *IEEE 802.11 (Legacy), Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, 1997.

[24] *IEEE 802.11e, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, 2005.

[25] K. Egevang and P. Francis, "The IP Network Address Translator (NAT)," *RFC 1631*, May 1994.

[26] H. Pillay, "Setting up IP Alliasing on a Linux Machine Mini-Howto," http://www.tldp.org/HOWTO/mini/IP-Alias, 2001.

[27] Y. Wu, X.-Y. Li, Y. Liu, and W. Lou, "Energy-Efficient Wake-Up Scheduling for Data Collection and Aggregation," *IEEE Trans. Parallel and Distributed Systems*, vol. 21, no. 2, pp. 275-287, Feb. 2010.

[28] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. ACM MobiCom*, pp. 263-270, 1999.

[29] S. Zhao and D. Raychaudhuri, "Scalability and Performance Evaluation of Hierarchical Hybrid Wireless Networks," *IEEE/ACM Trans. Networking*, vol. 17, no. 5, pp. 1536-1549, Oct. 2009.

[30] M. Cheng, X. Gong, and L. Cai, "Joint Routing and Link Rate Allocation under Bandwidth and Energy Constraints in Sensor Networks," *IEEE Trans. Wireless Comm.*, vol. 8, no. 7, pp. 3770-3779, July 2009.
[31] Bluetooth Special Interest Group, "Bluetooth Personal Area Networking Profile," *Specification of the Bluetooth System,* Ver. 1.1, 2001.
[32] R. Stanto, "Securing VPNs: Comparing SSL and IPsec," *Elsevier Computer Fraud and Security,* vol. 2005, no. 9, pp. 17-19, 2005.
[33] J. Flinn and M. Satyanarayanan, "Energy-Aware Adaptation for Mobile Applications," *Proc. ACM Symp. Operating Systems Principles (SOSP '99),* pp. 48-63, Dec. 1999.
[34] X.-Y. Li, Y. Wang, H. Chen, X. Chu, Y. Wu, and Y. Qi, "Reliable and Energy-Efficient Routing for Static Wireless Ad Hoc Networks with Unreliable Links," *IEEE Trans. Parallel and Distributed Systems,* vol. 20, no. 10, pp. 1408-1421, Oct. 2009.
[35] Y. He and R. Yuan, "A Novel Scheduled Power Saving Mechanism for 802.11 Wireless LANs," *IEEE Trans. Mobile Computing,* vol. 8, no. 10, pp. 1368-1383, Oct. 2009.
[36] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. ACM MobiCom,* pp. 85-97, 1998.
[37] S.H. Shah, K. Chen, and K. Nahrstedt, "Available Bandwidth Estimation in IEEE 802.11-Based Wireless Networks," *Proc. Workshop Bandwidth Estimation (BEst '03),* 2003.
[38] N. Golmie, N. Chevrollier, and O. Rebala, "Bluetooth and WLAN Coexistence: Challenges and Solutions," *IEEE Trans. Wireless Comm.,* vol. 10, no. 6, pp. 22-29, Dec. 2003.
[39] Atheros, http://atheros.com/news/AR6002.htm, Oct. 2007.
[40] iPhone Technical Specification, http://www.apple.com/iphone/specs.html, Mar. 2010.
[41] N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. Corner, "Virtual Compass: Relative Positioning to Sense Mobile Social Interactions," Microsoft Technical Report MSR-TR-2010-5, Jan. 2010.

Jong-Woon Yoo received the BS and MS degrees in electronic engineering in 2005 and 2007, respectively, from the Korea Advanced Institute of Science and Technology (KAIST), Korea, where he is currently working toward the PhD degree in electrical engineering, focusing on energy-efficient communication protocols for wireless mobile networks. His research interests include sensor networks, ubiquitous computing, and human-computer interaction (HCI). He is a member of the IEEE.

Kyu Ho Park received the BS degree in electronics engineering from Seoul National University, Korea, in 1973, the MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 1975, and the DrIng degree in electrical engineering from the Universite de Paris XI, France, in 1983. He has been a professor in the Department of Electrical Engineering and Computer Science, KAIST, since 1983. He was the president of the Korea Institute of Next Generation Computing in 2005-2006. His research interests include computer architectures, file systems, storage systems, ubiquitous computing, and parallel processing. He is a member of KISS, KITE, the Korea Institute of Next Generation Computing, the IEEE, and the ACM.