# HartFi: An Energy-Efficient Localization System

Song Han† Tianji Li†§ Chen Qian† Douglas Leith§ Aloysius K. Mok† Simon S. Lam†
†University of Texas at Austin §National Univ. of Ireland Maynooth
{shan, cqian, mok, lam}@cs.utexas.edu {tianji.li, doug.leith}@nuim.ie

## ABSTRACT

Location-based applications (LBAs) are emerging to be the killer applications on mobile devices. To know the whereabouts of devices, various interfaces (*i.e.*, GPS, Wi-Fi or cellular) can be used to sense their locations. Ideally, localization should be done all the time. However, keeping any of these interfaces running continuously would drain a device's battery rapidly. In this paper, we present a radical design of a collaborative localization system called HartFi, which enhances existing devices with a low-power 802.15.4-based WirelessHART interface. A salient feature of this added interface is that its energy consumption is up to two orders of magnitude less than that of a standard Wi-Fi interface; yet it provides a comparable range of coverage. In the HartFi system, therefore, WirelessHART interfaces are used whenever and wherever it is feasible to share location information that has been obtained using GPS/Wi-Fi/cellular interfaces. We have designed a mechanism to avoid location error accumulation in HartFi, which raises its localization accuracy to a level comparable to that of Wi-Fi localization. We are implementing a HartFi system at the moment and current results are promising.

## 1. INTRODUCTION

Location-based applications (LBAs), *e.g.*, Foursquare [1], Facebook Places [2], Gowalla [3], MicroBlog [12], TrafficSense [14], Pothole Patrol [11], PeopleNet [15], are becoming increasingly popular. A common and natural requirement of these applications is that they need to know the physical location of a device, and many applications would greatly benefit from having this information available at all times rather than only intermittently. However, existing localization technology is energy intensive and drains the battery of a mobile device too quickly to be usable for sustained periods.

The Global Positioning System (GPS [10]) currently provides the most accurate technique for localization of mobile devices. GPS can normally locate a device with errors of about 10 meters [12, 19]. However, GPS has two major disadvantages: 1) it is energy hungry, for example, Vtrack [19] and MicroBlog [12] report that a phone that was continuously using GPS can only last for about 10 hours before its battery is drained, and 2) if there is no line of sight between a GPS device and the satellites then location information cannot be determined, with an immediate consequence of this being that GPS often does not work at all in indoor environments and in city center areas with many high buildings [16].

To reduce energy consumption and enhance coverage within buildings and built-up areas, other interfaces, such as, cellular, Wi-Fi and Bluetooth interfaces can be used for locating devices. However, the tradeoff between location accuracy, coverage and energy consumption is complex:

- Cellular networks cover large areas and provide coverage both indoors and outside. Cellular interfaces consume less energy than GPS (a device using cellular localization can last for 60 hours). However, the location estimates via cellular base stations are much less accurate than GPS, with errors of around 400 meters [18]; see Table 1.
- Wi-Fi interfaces are nowadays ubiquitous on smartphones and tablets. Their estimation accuracy and energy efficiency lie between those of GPS and cellular: current techniques for Wi-Fi location estimation normally have errors of around 40-meters, and a device using Wi-Fi localization can last for about 40 hours [4, 8, 9].
- Modern mobile devices are also often equipped with Bluetooth interfaces. However, the coverage of Bluetooth only extends to about 10 meters and its energy consumption is comparable to Wi-Fi. (Note that the per-bit energy consumption of Bluetooth is much less than that of Wi-Fi. However, Bluetooth expends much greater time/energy in discovery and other tasks.) If this small coverage area is sufficient, location information using Bluetooth can be of high accuracy [16].

IEEE 802.15.4 technology (e.g., WirelessHART [17], ZigBee [6]) is becoming popular in sensor networks, smart home environments, and industry control. They can be expected to become available on mobile devices in the near future (e.g., SIM cards with a ZigBee chip embedded already exist). A salient feature of 802.15.4 is that radios consume up to two orders of magnitude less energy than standard Wi-Fi, while at the same time it covers an area comparable to Wi-Fi. The data rate supported by 802.15.4 (250 kbps) is one order of magnitude less than Wi-Fi's, but it is already sufficient for location-based applications.

In this paper, we introduce a radical design called HartFi. HartFi combines the use of Wi-Fi and WirelessHART for energy-efficient localization. In HartFi, only a few devices in the network need to directly communicate with Wi-Fi APs to determine their locations. Once the location information is available, it will be broadcasted using the low-power WirelessHART interface. After hearing these broadcast packets and at the same time measuring RSS (Received Signal Strength),

| | GPS | Wi-Fi | Cellular | Bluetooth | 802.15.4 |
|---|---|---|---|---|---|
| Lifetime(h) | 10 | 40 | 60 | 60 | TBD, but longer than cellular |
| Coverage(m) | Outdoor | 50 | Everywhere | 10 | 35-75 |
| Error(m) | 10 | 40 | 400 | 10 | TBD |
| Radio Emitting Power(mW) | Vary | 32–200 | 100–2000 | 10 | 1 |

**Table 1: Comparison among popular wireless technologies. Note that the actual energy consumption is greater than the radio transmission power due to the energy consumption of the analogue front-end (amplifiers etc), digital signal processing and, importantly, protocol overheads.**

other devices can derive their own locations and once again broadcast their locations. HartFi thus has the potential of achieving remarkable energy saving compared to existing Wi-Fi-based localization systems as most communication is now through low-power WirelessHART interfaces.

The HartFi system also features larger coverage than Wi-Fi. This is due to the fact that the WirelessHART interfaces form a multi-hop mesh network. Areas that are not covered by Wi-Fi APs can now be covered by this mesh network.

To ease deployment, we design an incremental algorithm with which the HartFi system works regardless of the number of WirelessHART interfaces. That is, if there is no WirelessHART, the HartFi system becomes the original Wi-Fi based system. As the number of WirelessHART increases, the gain of energy saving also improves. See Section 2 for details.

We are in the progress of implementing a hardware testbed and a large scale simulation. Preliminary but encouraging results can be found in Section 3.

Note that in this paper we use Wi-Fi as the reference interface. The proposed HartFi system works when one replaces the Wi-Fi with GPS, and indeed with more saving expected.

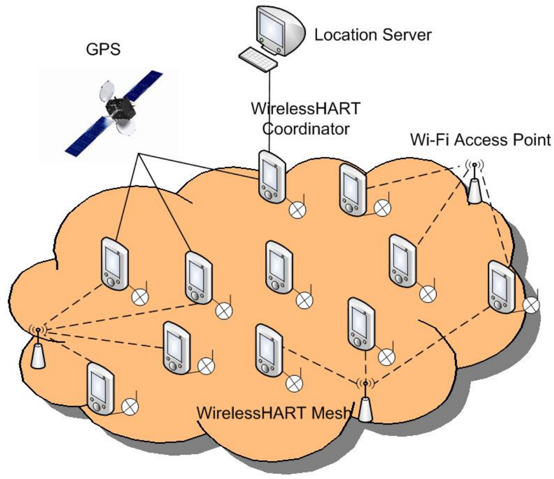


**Figure 1: The infrastructure of the HartFi system.**

## 2. THE HartFi SYSTEM

### 2.1 The Infrastructure

The infrastructure of the HartFi localization system is depicted in Fig. 1. A device in such system can have multiple interfaces, including GPS, Wi-Fi, cellular, Bluetooth and WirelessHART. All other interfaces work as usual, but the WirelessHART interfaces form a multiple-hop mesh network. In this mesh network, a device can either be a coordinator or a normal node. There is one and only one coordinator in this mesh, which schedule all the transmissions. That is, this mesh is TDMA in nature. All normal nodes simply join and listen to the coordinator's schedule.

In particular, formation of the WirelessHART mesh network consists of three main phases: network initialization, device joining and normal operation.

When a device powers up, its WirelessHART interface first scans all 16 channels to detect the advertisement messages broadcasted from existing WirelessHART nodes for a period of time. If it cannot hear any advertisement messages, it will take the role of coordinator. It will then broadcast advertisement messages periodically to notify others of its existence. Each advertisement contains the network id, the communication links that are available to new devices to join in and the absolute slot number for new devices to synchronize with the coordinator.

If the device can detect one or more advertisement messages around it, it will go through a join process to fully integrate into the network. The general progression that must be followed for the joining device to become operational includes the following steps:

- It listens for an advertisement message to synchronize to the network clock and identify potential parents.
- It then presents its credentials to the coordinator.
- The Network Manager at the coordinator provides the first session key and network key to the device. The device is then in the quarantined state.
- The Network Manager then proceeds to integrate the device into the network by provisioning the device with normal superframes and links.
- The device becomes operational and begins acquiring bandwidth and communication resources required for various communication purposes.

After the new devices have fully joined into the network, they will also broadcast advertisement messages. This will significantly increase the scale of the formed WirelessHART mesh network and covers many areas that cannot be covered by Wi-Fi access points.

In the normal operation phase, the WirelessHART devices will communicate with each other according to the communication schedule configured by the coordinator. If there is no data message exchanged between two devices, Keep-alive messages will be exchanged between them periodically to keep them synchronized.

### 2.2 Localization

To determine its own position, a device can use one of its available interfaces (called client NIC in the following) to communicate with a few network devices (called master or anchor devices) which know their own locations. For example, for Wi-Fi localization, the client interface is a Wi-Fi card and the master devices are APs. For cellular localization, the cellular interface is the client, the master devices are BSs.

While communicating with the master devices, a device can measure certain metrics to determine its distance to the master devices. Once three or more master devices' locations are known, this node's own location can be calculated using trilateration.

To compute the distance, a mobile device might measure up to four metrics regarding its position relative to a master device: angle of arrival, time of arrival, time difference of arrival, and RSS. However, the first three options require line of sight, which is usually not feasible in the indoor environments that we are targeting [8] [13]. Hence, we focus on the use of RSS measurements to infer distance, e.g. following [8]. The following simple example illustrates the potential for power saving in HartFi.

### 2.2.1 An Example

Suppose there are 3 APs and 3 mobile devices. If these devices have Wi-Fi only, then each of them has to power on its Wi-Fi interface in order to hear packet transmissions from each of the APs and thereby infer distance (via RSS measurements) and gather information on the AP locations (contained within the packets). Since the APs are assumed to know their own locations, each device can then estimate its location using trilateration. This requires each mobile device to keep its Wi-Fi interface powered on to receive at least 3 packet transmissions (one from each AP). The aggregate device power consumption then corresponds to that required to receive at least $3 \times 3 = 9$ packets.

If these devices have both Wi-Fi and WirelessHART interfaces, we proceed using the following bootstrapping approach. The first device powers on its Wi-Fi interface in order to receive packets from each of the 3 APs, and can then determine its location at the energy cost of 3 Wi-Fi packets. The second device can then use this HartFi device as a master device, and estimate its location by communicating with only 2 APs as well as the first HartFi device. This carries an energy cost of 2 Wi-Fi and 1 WirelessHART packet. The third device can determine its location from listening to 1 Wi-Fi AP and the first two HartFi devices, at the energy cost of 1 Wi-Fi and 2 WirelessHART packets. In total, the energy cost is 6 Wi-Fi packets and 3 WirelessHART packets. Since WirelessHART consumes about 1/100 radio energy of Wi-Fi, we can expect to have around a 33% energy saving in this case.

The energy saving will become even more significant as more devices have joined into the formed WirelessHART mesh. A new device can listen on the broadcast links and capture the <Loc, RSS> pairs from all its neighbors and then localize itself. It has to listen to the Wi-Fi APs only if not enough WirelessHART neighbors can be detected. By propagating its location through broadcasting on its WirelessHART interface, each new device further expands the HartFi localization system.

### 2.2.2 Localization Algorithm

Pseudocode for this iterative location estimation process is given in Algorithm 1. The basic idea of this algorithm can be summarized as follows. If a device in HartFi system has no WirelessHART interface, it just use Wi-Fi for localization. If the device has WirelessHART interface, it listens to the neighbor broadcasts. When it hears more than three HartFi devices (in 2D) that provides location information, it computes its own location by referring these HartFi devices. Otherwise, the device running this algorithm needs to communicate with APs to obtain enough master devices.

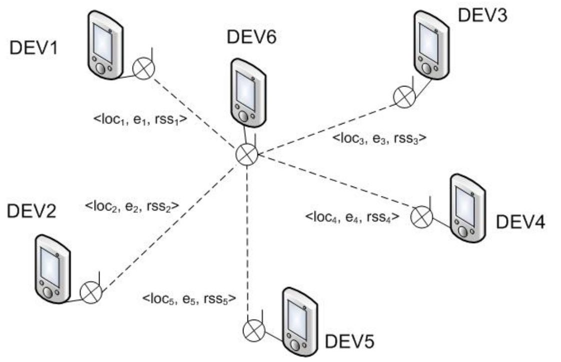


**Figure 2: Illustrating collaborative estimation with 5 HartFi devices**

The information provided by each master device $i$ is presented as a triple $<loc_i, e_i, rss_i>$, where $loc_i$ is $i$'s location, $e_i$ is $i$'s local error factor (explained in the next subsection), and $rss_i$ is the RSS measurement that infers distance.

**Algorithm 1** The HartFi Localization Algorithm

*hasWH* = TRUE;
$S_{wf} = \varnothing$; // The set of $<loc_i, e_{wf}, rss_i>$ triples from Wi-Fi APs
$S_{wh} = \varnothing$; // The set of $<loc_i, e_i, rss_i>$ triples from HartFi neighbors

*hasWH* = CheckWH();
**if** (*hasWH* == FALSE) **then**
    ScanAvailableWiFiAPList();
    Associate with the first 3 APs with best RSS and get $S_{wf}$;
    $<loc, e>$ = EstLocation($S_{wf}$);
**else**
    $S_{wh}$ = ListenNeighborLocBcast();
    **if** ($|S_{wh}| \geq 3$) **then**
        $<loc, e>$ = EstLocation($S_{wh}$);
    **else**
        ScanAvailableWiFiAPList();
        Associate with the first $3 - |S_{wh}|$ APs with best RSS and get $S_{wf}$;
        $<loc, e>$ = EstLocation($S_{wh} \bigcup S_{wf}$);
    **end if**
**end if**
Broadcast the $<loc, e>$ pair through WirelessHART interface;

### 2.2.3 Using Collaborative Localization for Error Control

In the foregoing example, it is clear that the accuracy of the location estimate derived by the third device is subject not only to errors in the distance measurements to the AP but is also affected by errors in the other two HartFi devices. We therefore expect that the estimation error for the third device will be higher than for the first two devices. When others use the third device as a master, the error will continue to propagate and accumulate.

There are two ways to mitigate this issue. First, we can turn on GPS and/or Wi-Fi interfaces when the errors are too large. Doing so however has to disadvantage of wasting energy and so we will consider this as a backup plan when the second technique is not efficient.

Second, errors can be mitigated by collaborative location estimation. In particular, each HartFi device considers all its neighboring master HartFi devices, rather than three of them, to localize itself. An example of using 5 master devices is shown in Fig. 2. Every HartFi device estimates its location by the algorithm EstLocation($S$) (pseudocode shown in Algorithm 2), where $S$ is the set of triples from master devices. The algorithm assigns the device a location $l$ that minimize the sum of the squared error between the distance from $l$ and the measured distance to each master device:

$$l = \underset{l}{\operatorname{argmin}} \sum_{i \in S} f_i(\| l - loc_i \| - d_i)^2$$

where for each master device $i$, $\| l - loc_i \|$ is the distance from $l$ to $loc_i$, $f_i$ is the confidence for referring this master device. $f_i$ is computed as:

$$f_i = e_{wf} / (e_{wf} + e_i)$$

where $e_{wf}$ is an empirical constant value of the average error in Wi-Fi localization, and $e_i$ is the error factor for the master device $i$. It is clear that when $e_i$ is large, the confidence value is small. We use $e_{wf}/(e_{wf}+e_i)$ rather than $1/e_i$ to avoid extreme values.

**Algorithm 2** EstLocation($S$) Algorithm

tb **Input**: A set $S$ of <$loc_i$, $e_i$, $rss_i$> triples from neighbors, and the current estimated location $l'$ and the error factor $e'$
**Output**: Device's updated estimated location $l$ and error factor $e$.

```
for each <loc_i, e_i, rss_i> triple in S do
    Derive d_i, the estimated distance to i, from rss_i according to the RSS model;
    f_i = e_wf / (e_wf + e_i); // confidence for each master device
end for
l = argmin_l Σ_{i∈S} f_i(|| l − loc_i || − d_i)^2
if l' = null then
    e = β · Σe_i/|S|; // β: tuning parameter larger than 1
else
    e_new = sqrt(Σe_i/|S| · || l − l' ||);
    e = α · e' + (1 − α) · e_new; // α ∈ (0,1): parameter for moving average
end if
return <l, e>;
```

The error factor for each AP is $e_{wf}$. When it is the first time for a HartFi device to run localization, it initializes its error factor to be $e = \beta \cdot \frac{\sum e_i}{|S|}$, where $\beta$ is a tuning parameter larger than 1. Such initialization means that the error factor is always larger than the average value of the error factors from its masters. After that, when finish location estimation once, the device updates its error factor by a moving average:

$$e = \alpha \cdot e' + (1 - \alpha) \cdot e_{new}$$

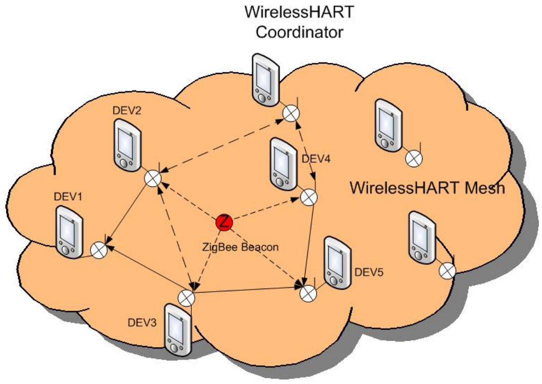


**Figure 3: Leveraging ZigBee devices in localization**

$$e_{new} = \sqrt{\frac{\sum e_i}{|S|} \cdot \| l - l' \|}$$

$l'$ and $e'$ are the previous estimated location and error factor. $e_{new}$ is the geometric mean of the average error factors of masters and the difference between the updated prior locations. When most masters have low error factors, or the updated location does not have much change, $e_{new}$ will be very small. It helps the system to converge.

After finishing updating $l$ and $e$, the HartFi device broadcasts these two values. Every HartFi device runs the localization process iteratively.

We are currently running large-scale simulations to find the proper value of $\alpha$ and $\beta$ for accurate and fast convergence.

## 2.3 Leveraging ZigBee devices in HartFi system

ZigBee interface is currently more popular in the market than WirelessHART. We choose WirelessHART instead of the ZigBee interface to collaborate with Wi-Fi in the HartFi system because WirelessHART has several dominating advantages over ZigBee. The most important is that it supports packet level channel hopping, which also plays a key role in other low-power wireless standards suited to coexistence with Wi-Fi in the 2.4 GHz frequency band.

We note that ZigBee and WirelessHART both use the same 802.15.4 PHY layer. The ZigBee MAC layer message is strictly compliant to the 802.15.4 MAC format while WirelessHART MAC message is the data type of the 802.15.4 MAC message with its first byte of the frame control field set as 0x41. For this reason, a WirelessHART interface can hear any ZigBee packets in its neighborhood. Since the PHY and MAC headers of ZigBee packets are not encrypted, WirelessHART interface can simply derive the transmitting ZigBee device's PAN id, nickname and the RSS of the messages. If multiple WirelessHART interfaces (for example, DEV2, DEV3, DEV4 in Fig. 3) can hear from the same ZigBee device, they will report the ZigBee device's RSS information to the coordinator. Combined with these reporters' own locations, the coordinator can estimate the location of the ZigBee device even it is not integrated into the HartFi system. The learnt ZigBee location can then be broadcasted in the

**Figure 4: One of our WirelessHART devices.**

whole mesh. This can further help other WirelessHART interfaces to reduce the number of possible requests to Wi-Fi access points. For example, DEV5 has the <loc, rss> pairs from neighbor DEV3 and DEV4, and can overhear the ZigBee device. With the addition of the ZigBee device's location which is broadcasted in the network, DEV5 can estimate its own location without talking to any Wi-Fi AP.

# 3. EXPERIMENTAL RESULTS

## 3.1 The Hardware

We set up a set of experiments to compare the power consumption between the Wi-Fi and WirelessHART interfaces. Four ASUS netbooks are used in the experiments (see Fig. 4). Each netbook is equipped with an internal 802.11b/g card and an external WirelessHART development board. Our WirelessHART board is built with the Freescale DEMOJM128 toolkit and runs the WirelessHART stack developed by ourselves [17]. The toolkit includes a 50.33 MHz ColdFire V1 core, 128 KB of flash memory and 16K RAM with security circuit. It also features an on-board logic analyzer, a virtual serial port and a Mini-AB USB connector.

Note that we picked netbooks for this work rather than phones since our current WirelessHART development board has only a serial and a USB interface, none of which can be connected to a phone. However, we note that the HartFi system should in future have the WirelessHART interface integrated with either the SIM card or the Wi-Fi card. The former integration is the same as the Telecom service's suggestion [7] for ZigBee and SIM card, and the latter has been done for Bluetooth and Wi-Fi by major Wi-Fi vendors such as Broadcom and Atheros. In private communication with Marvell, who also produces Bluetooth and Wi-Fi chipsets, we have been informed that integrating WirelessHART and Wi-Fi is straightforward given the current technology.

Note also that our WirelessHART interface is actually a development board, which means that results presented in this work can be greatly improved once it is really integrated in a phone.

## 3.2 Energy Consumption Measurements

| | Both Off ($T_{off}$) | WH ($T_{wh}$) | Wi-Fi ($T_{wf}$) | Saving |
|---|---|---|---|---|
| Netbook 1 | 92 | 82 | 58 | 70.588% |
| Netbook 2 | 115 | 107 | 84 | 87% |
| Netbook 3 | 123 | 117 | 88 | 80% |
| Netbook 4 | 91 | 81 | 61 | 63.33% |

**Table 2: Power consumption by the Wi-Fi and WirelessHART interfaces.**

We begin by presenting experimental measurements which provide an indication of the potential energy consumption reduction possible via use of the HartFi system. To collect data, we connect our testbed netbooks with an AP using their Wi-Fi interfaces, and they ping the AP at 1-second intervals. We then use the HartFi interface to connect the netbooks to a WirelessHART coordinator and they send 'hello' messages with the same size as the ping packets. In this way we are emulating the energy usage of a localization algorithm running at 1-second intervals. We measure the time (in minutes) taken to consume half of the full battery capacity (i.e. for the power meter to drop from 100% to 50%). These measurements are shown in Table 2. From these results, we can see that using the WirelessHART interface ($T_{wh}$) leads to 20 – 30 minutes longer battery life than when using the Wi-Fi interface ($T_{wf}$). This translates into a 75.23% power saving, calculated as $\frac{T_{wh}-T_{wf}}{T_{off}-T_{wf}} \times 100\%$.

## 3.3 Inferring Distance From RSS

We use the model proposed in [8] to map from RSS to distance, with parameters tuned to our test environment, i.e., the 5th floor of the ACES building at UT Austin. The floor map is not included due to limited space; please refer to [5] if interested. The RSS model for Wi-Fi is as follows.

$$P_d = \begin{cases} P_{d_0} - 10 \cdot n \cdot \log(\frac{d}{d_0}) - nW \cdot WAF & \text{if } nW < C \\ P_{d_0} - 10 \cdot n \cdot \log(\frac{d}{d_0}) - C \cdot WAF & \text{if } nW \geq C \end{cases}$$

where $d_0$ is the reference distance used in the model and $d$ is the transmitter-receiver (T-R) separation distance. $P_{d_0}$ is the received power at the reference distance and $P_d$ is the received power when the T-R separation distance is $d$. $n$ is the signal delay factor, $C$ is the limit for the number of walls, $WAF$ is the factor for the wall effect and $nW$ is the actual number of walls between a test point and the measured AP. We select 11 test points and measure the signal strength from the 3 APs in four directions. The measurement results are summarized in Fig. 5. It can be seen that the RSS model with parameter values of $d_0 = 1$ m, $P_{d_0} = -27$dBm, $n = 1.5, C = 4$, $WAF = 3.1$ is in good agreement with our measurements.

We also conduct a similar measurement for WirelessHART interface and the result is presented in Fig. 6. We put a WirelessHART device at the location of Wi-Fi AP3 and measure its signal strength. Since 3 of the 11 test points in the last experiment cannot be covered by the WirelessHART interface, we add 8 more test points (16 in total). The RSS model is kept unchanged except that $P_{d_0}$ is set as $-45$dBm. In Fig. 6, we observe that the RSS model matches the measurements very well. Although the one hop distance that a WirelessHART interface can cover (32m) is relatively shorter than that of a Wi-Fi interface (41 m), the WirelessHART interface has a

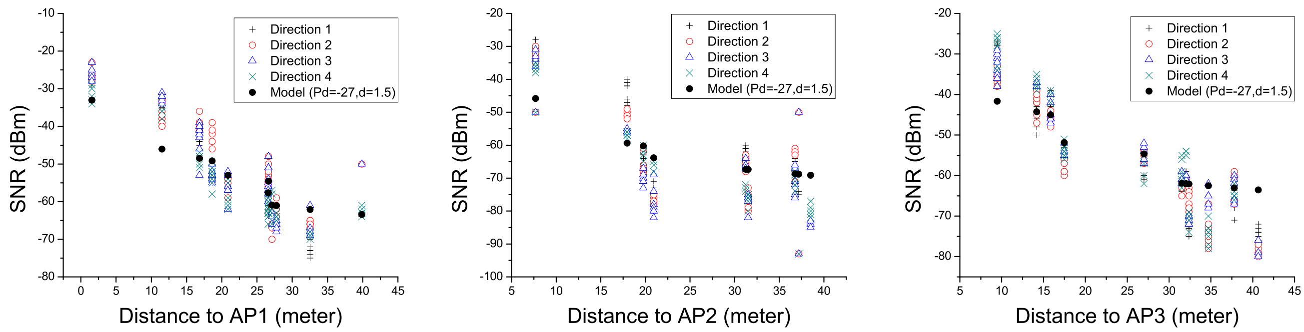


**Figure 5: Measurements vs. the RSS model of three Wi-Fi APs.**

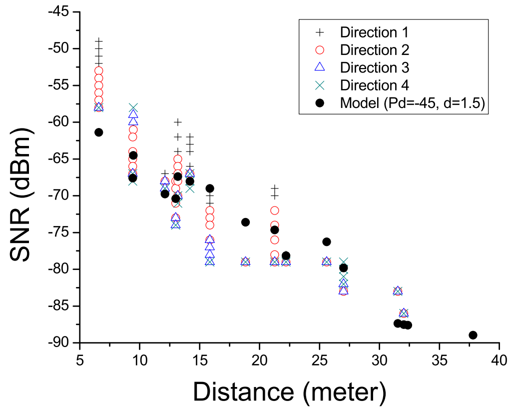


**Figure 6: Measurements vs. the RSS model of WirelessHART device.**

much lower power consumption. Notice that since the WirelessHART interfaces in HartFi system will form a multi-hop mesh, the HartFi localization system is expected to cover a much larger area than Wi-Fi system.

Currently we are measuring the localization accuracy in both HartFi and Wi-Fi localization systems. We are going to evaluate the performance of the error control mechanism (proposed in Section 2.2.3) in reducing the error propagation and accumulation in HartFi systems.

## 4. CONCLUSION AND FUTURE WORKS

Power saving is crucial for mobile devices especially when continuous location-based applications are running. Typical localization techniques based on GPS, Wi-Fi and Bluetooth are energy hungry while cellular localization suffers from unacceptable accuracy. In this paper, we present our radical design of a collaborative localization system called HartFi. HartFi takes the advantage of the lower-power 802.15.4 based WirelessHART interface to form a mesh among the mobile devices. Location information is broadcasted among the WirelessHART interfaces to reduce the usage of Wi-Fi interfaces as much as possible thus greatly prolongs the mobile device's lifetime. We also propose a novel technique to reduce the error propagation in HartFi system. HartFi localization system is expected to achieve the same level accuracy as Wi-Fi localization system while significantly reduce the energy consumption on mobile devices.

We are building a HartFi testbed in UT Austin. Initial results are promising and show that our RSS model in the testbed environment is accurate and our error propagation reduction technique is effective. We are doing more experiments to quantitatively compare the HartFi system with pure Wi-Fi system on energy consumption and localization accuracy.

## 5. REFERENCES


[1] Foursquare. http://foursquare.com/.
[2] Fplaces. http://www.facebook.com/places/.
[3] Gowalla. http://gowalla.com/.
[4] Skyhook. http://www.skyhookwireless.com/.
[5] Testbed floorplan. www.cs.utexas.edu/~shan/floor.pdf.
[6] ZigBee Alliance. http://www.zigbee.org.
[7] Expanding communication services zigbee telecom services enable value-added services for telecom. *Zigbee Alliance Whitepaper*, 2010.
[8] P. Bahl and V. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *Infocom*, 2000.
[9] I. Constandache, R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *Infocom*, 2010.
[10] P. Enge and P. Misra. Special issue on gps: The global positioning systems. *Proc. of the IEEE*, Jan. 1999.
[11] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *MobiSys*, 2008.
[12] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: Sharing and querying content through mobile phones and social participation. In *MobiSys*, 2008.
[13] H. Liu, H. Darabi, P. Banerjee, and H. Liu. Survey of wireless indoor positioning techniques and systems. In *IEEE Transactions on Systems, Man, and Cybernetics*, 2007.
[14] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys*, 2008.
[15] M. Motani, V. Srinivasan, and P. S. Nuggehalli. Peoplenet: engineering a wireless virtual social network. In *MobiCom*, 2005.
[16] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *MobiSys*, 2010.
[17] J. Song, S. Han, A. K. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt. WirelessHART: Applying wireless technology in real-time industrial process control. In *RTAS*, 2008.
[18] S. Tekinay. Wireless geolocation systems and services. *Communications Magazine, IEEE*, 36(4), Apr. 1998.
[19] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *SenSys*, 2009.