PinPoint: An Asynchronous Time-Based Location Determination System

Moustafa Youssef*
Depart. of Computer Science
University of Maryland
College Park, MD 20740, USA
moustafa@cs.umd.edu

Adel Youssef
Depart. of Computer Science
University of Maryland
College Park, MD 20740, USA
adel@cs.umd.edu

Chuck Rieger
AlphaTrek, Inc.
Ashton, MD 20861, USA
ChuckRieger@comcast.net

Udaya Shankar
Depart. of Computer Science
University of Maryland
College Park, MD 20740, USA
shankar@cs.umd.edu

Ashok Agrawala
Depart. of Computer Science
University of Maryland
College Park, MD 20740, USA
agrawala@cs.umd.edu

ABSTRACT

This paper presents the design, implementation and evaluation of the PinPoint location determination system. PinPoint is a distributed algorithm that enables a set of n nodes to determine the RF propagation delays between every pair of nodes, from which the inter-node distances and hence the spatial topology can be readily determined. PinPoint does not require any calibration of the area of interest and thus is rapidly deployable. Unlike existing time-of-arrival techniques, PinPoint does not require an infrastructure of accurate clocks (e.g., GPS) nor does it incur the $o(n^2)$ message exchanges of "echoing" techniques. PinPoint can work with nodes having inexpensive crystal oscillator clocks, and incurs a ***constant*** number of message exchanges per node to determine the location of n nodes. Each node's clock is assumed to run reliably but asynchronously with respect to the other nodes, i.e., they can run at slightly different rates because of hardware (oscillator) inaccuracies. PinPoint provides a mathematical way to compensate for these clock differences in order to arrive at a very precise timestamp recovery that in turn leads to a precise distance determination. Moreover, each node is able to determine the clock characteristics of other nodes in its neighborhood allowing network synchronization. We present a prototype implementation for PinPoint and discusses the practical issues in implementing the mathematical framework and how PinPoint handles the different sources of error affecting its accuracy. Evaluation of the prototype in typical indoor and outdoor environments shows that PinPoint gives an average accuracy of four to six feet, in different environments, allowing PinPoint to support accurate rapidly deployable localization scenarios.

*Also affiliated with Alexandria University, Egypt.

MobiSys'06, June 19–22, 2006, Uppsala, Sweden.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems

General Terms

Algorithms, Measurement, Performance, Theory.

Keywords

Location determination, rapidly deployable location determination, time-of-arrival location determination, ranging techniques

1. INTRODUCTION

Location determination technologies have been an active research area. Many systems have been developed over the years based on different technologies including GPS [10], wide-area cellular-based systems [28], infrared-based systems [2, 30], various computer vision systems [16], physical contact systems [21], and radio frequency (RF) based systems [3, 15, 17, 23, 34]. These systems provide the technology needed for a wide range of applications. Examples [6, 18] include ubiquitous computing environments, location-based protocol enhancement, and location support for disaster management systems.

In this paper, we present the design, implementation and evaluation of the PinPoint[1] location determination system for ad-hoc networks. PinPoint is a distributed protocol that runs on all the nodes in the network. It does not require any pre-planing for deployment and thus is suitable for scenarios that require rapid deployment.

[1]The PinPoint technology presented here is different from the PinPoint technology described in [32]. The differences between the two technologies are discussed in Section 6 on related work.

PinPoint is a time-of-arrival (ToA) based system which depends on measuring the time it takes a signal to reach the receiver to estimate the distance between the sender and the receiver. Traditional ToA based systems either require synchronized clocks, e.g. the GPS system [10], or uses the "echoing" method, e.g. [32, 22], where a node measures the roundtrip time of a signal transmitted to a remote node to estimate the distance to this node. Systems that require synchronized clocks are expensive to implement while systems based on the "echoing" method require $o(n^2)$ message exchanges to estimate the location of n nodes ($o(n)$ per node) and suffer from more variance in the time measurement due to the echoing requirement. PinPoint does not require synchronized clocks and requires $o(n)$ message exchanges to locate n nodes ($o(1)$ messages per node) while using one-way messages only.

Central to the PinPoint idea is the notion of sending and receiving a "timestamp" between two mobile units. Each unit is assumed to have its own clock, which runs reliably but ***asynchronously*** with respect to the other units, i.e., the clocks' current "time of day" might be considerably different, and they might run at slightly different rates because of hardware (oscillator) inaccuracies. PinPoint provides a mathematical way to compensate for these clock differences in order to arrive at a very precise timestamp recovery that in turn leads to a precise distance determination. Moreover, each node is able to determine the clock characteristics of other nodes in its neighborhood allowing network synchronization.

PinPoint works in two steps. In the first step (*ranging*), each node estimates the distances to all the other nodes in its neighborhood. In the second step (*range combining*), the nodes use the distance estimates to estimate the network topology. In this paper ***we focus on the ranging problem*** as it is the more challenging one. We briefly discuss how PinPoint performs range combining in Section 3.

PinPoint has the following features:

- No time synchronization is required between nodes in PinPoint. Nodes measure the transmission and reception time of a signal based on their local clock.
- PinPoint does not require any infrastructure support or pre-planning for the area of deployment. Therefore, it is well suited for scenarios that require rapid deployment.
- PinPoint has low computation cost. The algorithm used is based on simple algebraic operations on scalar values. Moreover, it requires $o(1)$ number of messages per node to locate n nodes.
- PinPoint is independent of the communication technology used between the nodes. For example, it can be used with the *narrow-band RF*, *spread spectrum* [29], or *ultra wideband* technologies [31]. In this paper, we present a prototype implementation of PinPoint using the spread spectrum technology (compatible with the 802.11 standard [29]).

The rest of the paper is organized as follows: In Section 2 we present the basic idea behind PinPoint, the mathematical framework it uses to obtain the distance estimate, and the different sources of error affecting PinPoint accuracy. Section 3 describes the PinPoint node architecture and the PinPoint protocol. We present our prototype implementation of PinPoint in Section 4. Section 5 presents PinPoint evaluation based on the developed prototype. In Section 6 we discuss the related work. Finally, sections 7 and 8 discuss different aspects of the system and conclude the paper.

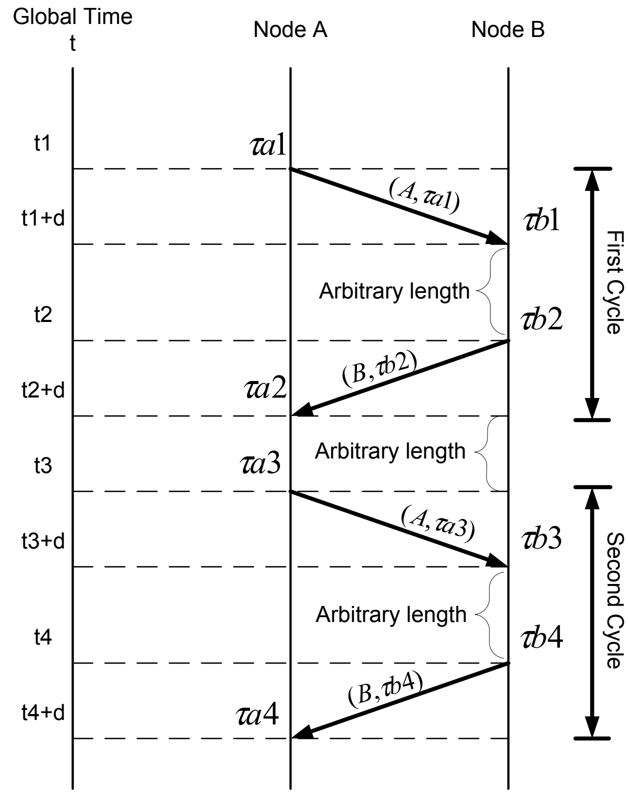

Figure 1: The PinPoint measurement phase. The phase contains two cycles with n (number of nodes) turns per cycle. Note that all messages are one-way messages and all times are measured by local clocks.

2. PINPOINT FUNDAMENTALS

2.1 Introduction

The goal of PinPoint is to be able to determine the relative positions of nodes, possibly mobile, in 3D. This section describes how two units, say A and B, in a set of units participating in the PinPoint protocol obtain the distance between them. The general protocol between a set of n nodes is described in Section 3.4.

Basic PinPoint operation (Figure 1) calls for one unit (unit A) to broadcast a signal to all other nodes (here unit B). As A sends the signal, it captures the transmit timestamp. B receives this signal, and captures the receive timestamp. The units then swap roles, wherein unit B broadcasts a timestamp signal with B capturing its transmit timestamp and A capturing its receive timestamp. These exchanges are repeated twice. Finally, the units swap information about all sent and received timestamps with each other so that both sides know the eight timestamp values.

After this exchange each unit has enough information, as explained in the rest of the section, to eliminate the basic difference in the two units' clock readings, and hence to determine the time of flight of the signal between the two units. Once the time of flight is known, the physical distance between the two units is known, because of the constant nature of the speed of light (and radio waves). In addition, each node also determines the clock characteristics (drift and offset relative to itself) of the nodes in its neighborhood. In the rest of this section, we give the mathematical formulation of PinPoint.

2.2 Notation

As a notational convention, we will always use τ to denote local clock times, and t to denote global clock times. Also, when discussing local clock times, the letter contained in the subscript on the τ will always indicate the clock which records the time, so for example τ_{a1} is a time recorded by the local clock at node A. We refer to the clock offset and drift rate of a node as α and β respectively.

Throughout, we use time equivalent of distance, i.e., we represent a distance d by the time it will take light to travel that distance. In the environments we expect this technology to be used, the speed of light does not vary significantly, thus making this measure of distance stable.

2.3 Mathematical Formulation

Without loss of generality, we fix on any two nodes, A and B, in the network, and assume that they are neighbors. We describe the messages that they expect to receive from each other and the processing involved in estimating the location. The same discussion holds for the case of n nodes as discussed in Section 3.4. Both nodes A and B have local clocks, which may have some drift and offset, and for global time t, their clock readings are respectively

$$\begin{aligned} \tau_a &= \beta_a(\alpha_a + t) \\ \tau_b &= \beta_b(\alpha_b + t) \end{aligned} \tag{1}$$

Figure 1 shows the messages exchanged and the timing in a measurement cycle. Now, at time t_1 node A broadcasts a tuple giving its identity and a timestamp, the latter denoted by τ_{a1}. That is,

$$\tau_{a1} \equiv \tau_a(t_1) = \beta_a(\alpha_a + t_1) \tag{2}$$

and the tuple broadcast is (A, τ_{a1}). Node B receives it and records the time of receipt as τ_{b1}. If we denote the time distance between A and B as d, then the (global) time at which B should receive the broadcast from A is $t_1 + d$, so that

$$\tau_{b1} = \beta_b(\alpha_b + t_1 + d) \tag{3}$$

Since each node is running the same decentralized protocol, B also sends a two-tuple, say at global time t_2. In steps similar to above, B broadcasts the tuple (B, τ_{b2}), where

$$\tau_{b2} \equiv \tau_b(t_2) = \beta_b(\alpha_b + t_2) \tag{4}$$

and A receives this broadcast at global time $t_2 + d$, which A believes is time

$$\tau_{a2} = \beta_a(\alpha_a + t_2 + d) \tag{5}$$

Once this first round of messages has completed, they both send a second round, say with A sending its second message at time t_3, and B sending its message at time t_4. Using notation similar to before, we thus have $\tau_{a3} = \beta_a(\alpha_a + t_3), \tau_{b3} = \beta_b(\alpha_b + t_3 + d), \tau_{b4} = \beta_b(\alpha_b + t_4)$, and $\tau_{a4} = \beta_a(\alpha_a + t_4 + d)$.

Collecting all eight of these together in one place for reference (we refer to these as "the reference equations"), we have:

$$\begin{array}{l|l} \tau_{a1} = \beta_a(\alpha_a + t_1) & \tau_{b1} = \beta_b(\alpha_b + t_1 + d) \\ \tau_{a2} = \beta_a(\alpha_a + t_2 + d) & \tau_{b2} = \beta_b(\alpha_b + t_2) \\ \tau_{a3} = \beta_a(\alpha_a + t_3) & \tau_{b3} = \beta_b(\alpha_b + t_3 + d) \\ \tau_{a4} = \beta_a(\alpha_a + t_4 + d) & \tau_{b4} = \beta_b(\alpha_b + t_4) \end{array} \tag{6}$$

Now that both rounds of messages are complete, the nodes enter the information exchange portion of the algorithm, with A sending the values τ_{a2} and τ_{a4} to B, and B sending the values τ_{b1} and τ_{b3} to A. At this point, both of the nodes know all eight values τ_{a1}, τ_{a2}, τ_{a3}, τ_{a4}, τ_{b1}, τ_{b2}, τ_{b3}, and τ_{b4}.

The nodes use these values to independently compute the ratio β_a/β_b. To see how this is done, note that

$$\frac{\tau_{a3} - \tau_{a1}}{\tau_{b3} - \tau_{b1}} = \frac{\beta_a(\alpha_a + t_3) - \beta_a(\alpha_a + t_1)}{\beta_b(\alpha_b + t_3 + d) - \beta_b(\alpha_b + t_1 + d)} = \frac{\beta_a}{\beta_b} \tag{7}$$

The next series of steps is designed to allow for the calculation of the values $\beta_a d$ and $\beta_b d$. The reasons for this are twofold. First, both β_a and β_b are typically close to one, so the quantities $\beta_a d$ and $\beta_b d$ are both good estimates of d. Second, both quantities are valuable for helping synchronize the clocks in the network.

We show here how to determine $\beta_b d$. The development for $\beta_a d$ is similar. First, consider the quantities Δ_1 and Δ_2 given by

$$\Delta_1 \equiv \tau_{b1} - \tau_{a1} = \beta_b(\alpha_b + t_1 + d) - \beta_a(\alpha_a + t_1) \tag{8}$$

and

$$\Delta_2 \equiv \tau_{a2} - \tau_{b2} = \beta_a(\alpha_a + t_2 + d) - \beta_b(\alpha_b + t_2) \tag{9}$$

Averaging these two quantities gives

$$\begin{aligned} \frac{\Delta_1 + \Delta_2}{2} &= \frac{1}{2}[\beta_b(\alpha_b + t_1 + d) - \beta_a(\alpha_a + t_1) \\ &\quad + \beta_a(\alpha_a + t_2 + d) - \beta_b(\alpha_b + t_2)] \\ &= \frac{\beta_a + \beta_b}{2} d + \frac{t_2 - t_1}{2}(\beta_a - \beta_b) \end{aligned} \tag{10}$$

Now, from the first and second lines of the reference equations (6), we have

$$\tau_{a2} - \tau_{a1} = \beta_a(t_2 - t_1) + \beta_a d \tag{11}$$

or equivalently that

$$t_2 - t_1 = \frac{\tau_{a2} - \tau_{a1}}{\beta_a} - d \tag{12}$$

Substituting Equation 12 into Equation 10 gives

$$\begin{aligned} \frac{\Delta_1 + \Delta_2}{2} &= \frac{\beta_a + \beta_b}{2} d + \frac{\beta_a - \beta_b}{2}\left(\frac{\tau_{a2} - \tau_{a1}}{\beta_a} - d\right) \\ &= \beta_b d + \frac{1}{2}\left(1 - \frac{\beta_b}{\beta_a}\right)(\tau_{a2} - \tau_{a1}) \end{aligned} \tag{13}$$

or equivalently

$$\beta_b d = \frac{\Delta_1 + \Delta_2}{2} + \frac{1}{2}\left(\frac{\beta_b}{\beta_a} - 1\right)(\tau_{a2} - \tau_{a1}) \tag{14}$$

Note that every quantity on the right side of Equation 14 is computable by every node, as long as they have the eight quantities on the left sides of the reference equations. Note further that determining $\beta_b d$ gives us $\beta_a d$, since $\beta_a d = \left(\frac{\beta_a}{\beta_b}\right)(\beta_b d)$, and the nodes know both right hand side quantities.

2.4 Discussion

All the message exchanges in the PinPoint protocol are one-way and timestamped by the local clocks. PinPoint makes use of the broadcast medium to achieve its $o(n)$ one-way message exchanges (Section 3.4). Note that although "echoing" based systems, e.g. [32, 22], may have a broadcast medium, these systems cannot benefit from it as a node must unicast each message to one of the nodes and wait for the echo message from this node to estimate the round trip time, given a total of $o(n^2)$ message exchanges.

2.5 Sources of Error

There are three main sources of errors in PinPoint: the built-in hardware delay, multipath effect, and the non-line-of-sight (NLOS) transmission.

The built-in hardware delay refers to the delay of the different components of PinPoint hardware before the timestamping process. Usually, this delay is constant or exhibits small variations. The architecture of the PinPoint unit (as discussed in Section 3) permits it to be self-calibrating: A PinPoint node could periodically send and timestamp a calibration sequence to itself to adjust its own zero reference (i.e. the node will act as both the sender and the receiver of the PinPoint messages).

Multipath refers to the fact that a signal transmitted from the sender may reach the receiver through different paths. Since each path can have a different length, different signals may arrive at different times at the receiver antenna. There are two important characteristics for multipath propagation [24]. The direct (LOS) signal will always reach the receivers before the other multipath components. Second, multipath components will normally be weaker than the LOS component. PinPoint mitigates the effect of multipath by basing its calculation on the *first* and *longest* chain of received baseband signals (as discussed in Section 4). This corresponds most of the time to the LOS component based on the above two characteristics.

NLOS transmission exists when the LOS component is blocked due to the presence of RF-opaque objects. In this case, the receiver timestamp is based on a NLOS signal and hence overestimates the true distance. PinPoint mitigates the NLOS effect in its range-combining phase where the estimated distances are used to estimate the locations of the nodes. In this phase, the redundant distance information and estimated distances from other nodes (where LOS propagation is dominant) can help detect the outliers (those based on NLOS measurements) and reduce the overall error. Moreover, since PinPoint is independent of the communication technology, a communication technology (and/or a transmission frequency) can be used to reduce the probability of NLOS propagation[2].

In Section 5 we present the performance evaluation of a PinPoint prototype in typical indoor and outdoor environments that shows the stable PinPoint performance under these sources of error.

[2]UWB [31, 9] is a technology that can be used for this purpose.

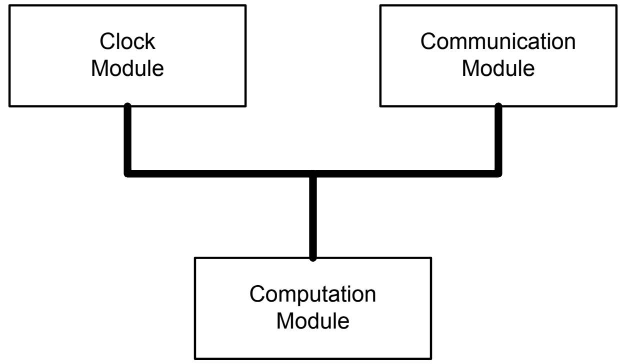

Figure 2: PinPoint node architecture. Each node consists of three modules: a clock module, a communication module, and a computation module, all interconnected by a bus.

3. PINPOINT NODE ARCHITECTURE AND PROTOCOL

Each PinPoint node consists of three modules: a clock module, a communication module, and a computation module, all interconnected by a bus (Figure 2). The modules and the bus are realizable from off-the-shelf hardware components as described in Section 4.

3.1 Clock Module

The clock module has a time-of-day clock whose offset and drift are assumed to be essentially constant over a few tens of milliseconds. High clock drifts, of the order of 100 parts per million (ppm), are perfectly acceptable. Thus PinPoint nodes can utilize inexpensive crystal oscillators to implement their clocks. The clock module also has a countdown timer, driven by the time-of-day clock, and registers for storing clock values at certain communication events.

3.2 Communication Module

The communication module has transmit and receive message buffers, the modulation and demodulation units, and the antenna. PinPoint technology requires only half-duplex operation.

The communication module and clock module are linked together so that the receive and transmit times of messages can be recorded. The transmit time of a message is the value of the transmitting node's time-of-day clock when a specified bit is transmitted. The receive timestamp of a message at a node is the value of the receiving node's time-of-day clock at the arrival of the specified bit. The required interconnection between the communication and clock modules to achieve this is implementable with nominal hardware.

3.3 Computation Module

The computation module consists of a general purpose computation and storage engine. It manages the clock and communication modules and carries out the operations of the PinPoint protocol described in the next section.

3.4 PinPoint Protocol

The PinPoint protocol is the protocol governing the exchange of messages, the measurement of send and receive times, the dissemination of this measurements to all nodes, and the calculation of the spatial locations and clock attributes of all nodes. We assume that all nodes are within the listening range of one another. We discuss the general case of large networks in Section 7.

The protocol operation consists of three phases: measurement phase, information exchange phase, and computation phase.

3.4.1 Measurement Phase

A measurement phase consists of two measurement cycles. In each cycle, each node broadcasts (in its turn) a message containing its ID and the transmit timestamp of the message, and records the receive timestamp of the messages broadcast by other nodes (as described in Section 2.3). For a network of n nodes, each node takes a turn for a total of n turns per cycle ($2n$ turns in the entire measurement phase) [3]. For example, in Figure 1, there are two turns per cycle corresponding to the two nodes. We assume that units will remain stationary during the *PinPoint* message exchange phase [4].

3.4.2 Information Exchange Phase

In this phase, each node broadcasts a message containing its receive timestamp for messages transmitted by other nodes in a measurement phase. For a network of n nodes, we have n turns in the information exchange phase.

3.4.3 Range Combining Phase

After the Information Exchange phase, each node has enough information to compute an estimate of the distance (range) to all nodes within its listening range, as described in the Section 2.3. The goal of the range-combining phase is to obtain the topology of the network using the estimated distances. Each PinPoint node uses a technique similar to the self-positioning algorithm (SPA) described in [5]. We assume that this algorithm will run only once. That is, once a local topology is determined, nodes entering or leaving are added or removed incrementally. No communication takes place during this phase. Therefore the total number of messages exchanged in the entire network is $3n$ which is a ***constant*** number of message exchanges per node.

4. PINPOINT IMPLEMENTATION

This section describes the current implementation of PinPoint[5]. We refer to this prototype as **PP2**. Figure 3 shows the current prototype.

4.1 PinPoint Hardware

The current PinPoint implementation is based on the following main hardware components:

- Altera Cyclone 1C20 FPGA development kit
- Maxim 2820 radio with Maxim 2242 RF power amp

[3]We discuss the contention resolution between nodes in Section 7.5.

[4]This is a reasonable assumption considering the length of the message exchange phase.

[5]We chose to build our own hardware out of off-the-shelf components to have full control over the different components of the system. This helped in reducing the variability in the system.

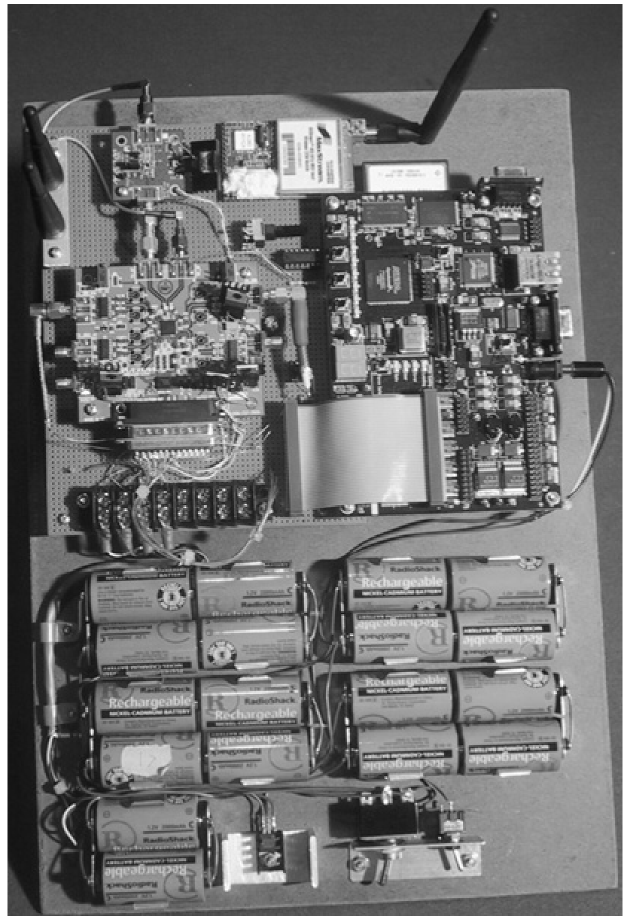

Figure 3: PinPoint prototype (PP2). The prototype is based on an Altera Cyclone 1C20 FPGA development kit, a Maxim 2820 radio, and MaxStream 9xStream radio modem.

- MaxStream 9xStream radio modem

The Altera Cyclone 1C20 FPGA provides the computation processor and the timestamping clock while the Maxim 2820 and Maxstream radio provide the communication modules needed for transmitting the signal used for timestamping and transferring the PinPoint protocol messages respectively. The rest of the section gives an overview about each of these components.

4.1.1 Altera Cyclone 1C20 FPGA development kit

The Cyclone 1C20 FPGA (field programmable gate array) from Altera has a phase lock loop (PLL) capable of running at clock rates up to 300 MHz, which we use as the PP2's main timestamping clock (giving around 3 ns accuracy).

4.1.2 Maxim 2820 radio with Maxim 2242 RF power amp

The MAX2820 is a 2.4 GHz radio chip intended for 802.11b and similar applications. The chip can cover distances of up to several hundred feet - compatible with PinPoint's operating range goals. The chip is needed for transmitting the signal used for timestamping.

Note that PinPoint design can be integrated with the current 802.11 based cards alleviating the need for the MAX2820 module.

4.1.3 MaxStream 9xStream radio modem

PinPoint's conceptual design requires that the PP2 units be able to exchange high-level information about timestamps after generating and acquiring those timestamps via a lower-level protocol (described in the next section). Although the MAX2820 radio chip would serve for both the high- and low-level protocols, the PP2 design implements these two protocols via two separate radio technologies: the MAX2820 radio chip for the low-level protocol, and a radio modem unit, the MaxStream 9xStream for the high-level protocol. The rationale behind this separation was simply to limit PP2 development time and cost, since the MAX2820 evaluation kit does not include the required hardware or software to implement a high-level data exchange protocol.

The MaxStream 9xStream unit offers a simple, UART-like data transport protocol using 900 MHz FM radio technology, which is capable of sending and receiving ASCII data streams over distances well in excess of PinPoint's requirements. The 9xStream performs error checking and correction, and is thus considered to be a completely reliable wireless data transport channel.

We want to emphasize two points here:

- PinPoint is independent of the technology used for data transmission. In the current prototype, data transfer occurs over 900 MHz FM radio technology.
- If PinPoint technology is integrated with an 802.11 based card, this module will not be needed as the timestamp information can be piggy-backed on 802.11 frames.

4.1.4 Timestamping

Due to noise, signal reflections, and other factors, we use a repetitive pattern of baseband pulses rather than a single signaling event to obtain a robust signaling technique. Having a repetitive stream of events allows the receiver to miss some or even most of them, yet still be able to identify moments in time that were also known to the transmit side.

After a variety of experiments, the following timestamp signaling protocol was selected (Figure 4):

- Send 20, 128 μs, baseband cycles.
- Send one baseband cycle with its 64 μs low half filled with 18 "dense pulses", each of cycle time approximately 3.2 μs
- Send an additional 20 baseband cycles.

By convention, the receiver identifies the "official" timestamp as the last rising baseband edge prior to the dense pulses. Since each baseband cycle takes 128 μs, this 41-cycle broadcast lasts just over 5 milliseconds (ms), fast enough in principle to be repeated many times per second.

4.1.5 Power Requirements

Currently, each PinPoint unit uses 18 rechargeable NiCad batteries (each 2000 ma-hour, 1.2 volts). Ten of them form the positive voltage source of nominally +12 volt. A +5 volt regulator shifts some of this unregulated supply down

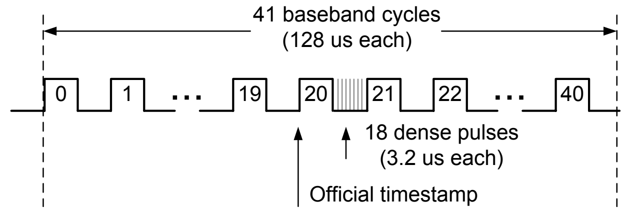

Figure 4: Timestamp timing sequence. The receiver identifies the "official" timestamp as the last rising baseband edge prior to the dense pulses.

to drive several components needing regulated +5 volts, and the rest goes to the Altera and radio boards, which have their own regulators onboard. The other eight batteries form the negative supply branch, nominally -9.6 volts, which is only used by several line drivers on the radio board.

Note that for future versions of PinPoint, integrated with wireless cards, PinPoint can be powered from the card's supply without requiring external power source.

4.2 PinPoint Software

4.2.1 PinPoint Operating System

The Altera FPGA development system includes a full C software development environment that is capable of compiling, then downloading and running ANSI standard C code on the Altera CPU (via the UART contained in the Altera reference design). This is the software development environment employed to develop the PinPoint operating system, PP2.C.

4.2.2 PinPoint Protocol Implementation

The current implementation of the PinPoint protocol is based on polling. We chose this protocol for its simplicity and we discuss other alternatives in Section 7. For a set of n nodes, one of them is a master node that polls the other $n-1$ nodes in turn. When the master polls a node, this node performs its turn of the protocol as explained in Section 3. Changing any of the nodes parameters is performed at any time through connecting the unit to the PC through the UART interface.

5. SYSTEM EVALUATION

We conducted several experiments to investigate the performance of PinPoint. In the first experiment, we examine how well PinPoint captures the relation between distance and time and how consistent the measurements are. The second set of experiments aims at studying the performance of PinPoint in indoor and outdoor environments for static units. The third set of experiments examines the performance of PinPoint when the unit is mobile.

In all experiments, we used two PP2 units. We call them the "log unit" and the "test unit". We connected the log unit to the PC in all experiments and logged all events and data related to the PinPoint protocol operation. The test unit was freely moving and we changed its location to evaluate different aspects of the PinPoint prototype. We start by presenting the system parameters.

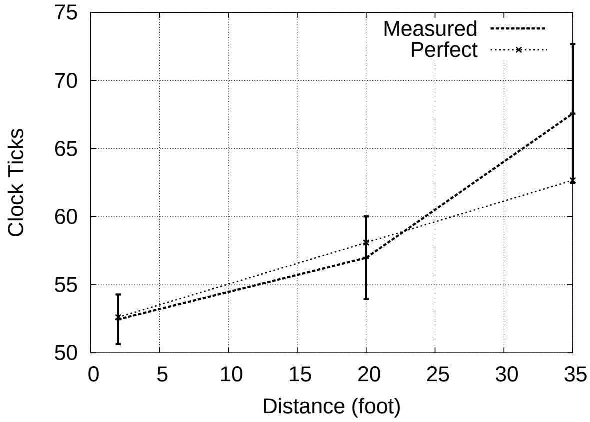

Figure 5: Relation between distance and the reported clock ticks. The error bars represent one standard deviation.

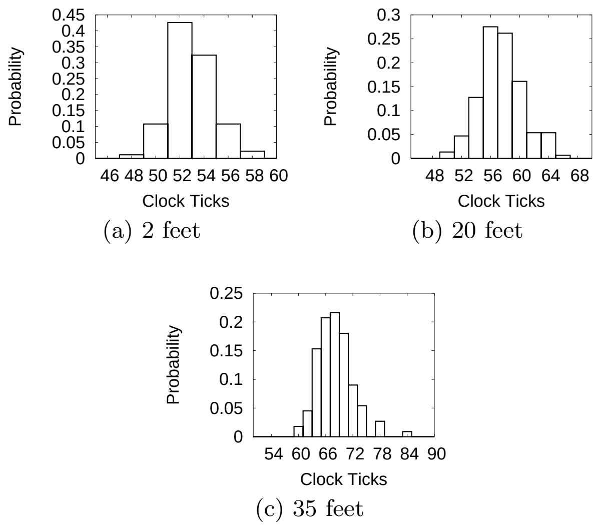

Figure 6: Clock ticks histogram for different distances.

5.1 System Parameters

As we pointed out in Section 4, the PP2 prototype uses a 300 MHz clock for timestamping. This means that each tick is 3.33 nanoseconds, and since light travels approximately 11.8 inches in one nanosecond, then each tick of the timestamp clock equates to approximately 3.28 feet of distance.

We calibrated the units by putting the test unit adjacent to the log unit. We measured a clock reading of 52 for this zero-distance configuration. This reading corresponds to the system overhead (built-in hardware delay). We subtract this offset from all of our calculations to get the time equivalent of distance. Multiplying this time by 3.28 gives the estimated distance in feet.

5.2 Basic Protocol Performance

Figure 5 shows the clock ticks reported by PinPoint when the test unit moves away from the log unit. We can see from the figure that as expected, as the distance increases, the number of clock ticks the signal takes to travel between the two units increases. The relation, however, is not a perfect straight line due to the error in measurements. We will discuss the error in PinPoint measurements in the next section.

Figure 6 shows the reported clock ticks histogram corresponding to the three distances in Figure 5. The variance of clock ticks histogram increases as the distance increases. The reason is that for larger distances, the transmitted signal becomes weaker and the signal to noise ratio decreases so that some erroneous dense edges may be detected incorrectly affecting the accuracy of the measured time.

5.3 Static Performance

We used two testbeds to evaluate the performance of PinPoint: an indoor testbed and an outdoor testbed.

5.3.1 Indoor testbed

We performed two indoor experiments. The first experiment was conducted in a basement of an area of approximately 1450 square feet (Figure 7). The basement is divided into three areas: the workshop, the pool table area, and the recreation area. The basement environment presents a challenging indoor environment as it has cabinets, book shelves, metallic objects, electronic devices, etc. For this testbed, the mobile unit was placed at eight different locations (both LOS and NLOS) in the basement and we recorded the clock readings. Figure 8 shows the actual and measured distances for the different test locations. We can see from Figure 8 and the derived summary in Table 1 that PinPoint has an average accuracy of 4.18 feet (4.4 standard deviation) and worst case error of 14.2 feet (Location 7) in this complex indoor environment. We believe that the performance at Location 7 is due to the severe multipath environment at that location.

Our second indoor testbed covered a multi-floor testbed and there were no direct LOS between the mobile unit and the log unit at all the 17 locations. Figure 9 shows the results. We can see from the figure that PinPoint maintains its performance even with no LOS between units. This confirms our discussion in Section 2.5. We also note from the table that the maximum error in the first experiment (at Location 7) is more than that of the second experiment (at Location 14). We believe that this is because the basement provides more opportunity for multipath effects than any of the 17 NLOS test locations.

5.3.2 Outdoor testbed

For the outdoor testbed (Figure 10), we placed the log unit outside a house (in the driveway) while the PP2 test unit was placed at seven test locations within a test radius of approximately 150 feet.

All outdoor test locations had a clear line of sight between the two units and we believe that reflections were present from neighboring houses, trees, several cars, and similar objects.

Figure 11 shows the PinPoint accuracy for the outdoor testbed. The average accuracy was 6.85 feet (4.35 standard deviation) with a maximum error of 13.64 feet (Location 2). These results are comparable to the indoor testbed results which indicates the consistency of the PinPoint performance. Table 1 summarizes the results of the indoor and outdoor experiments.

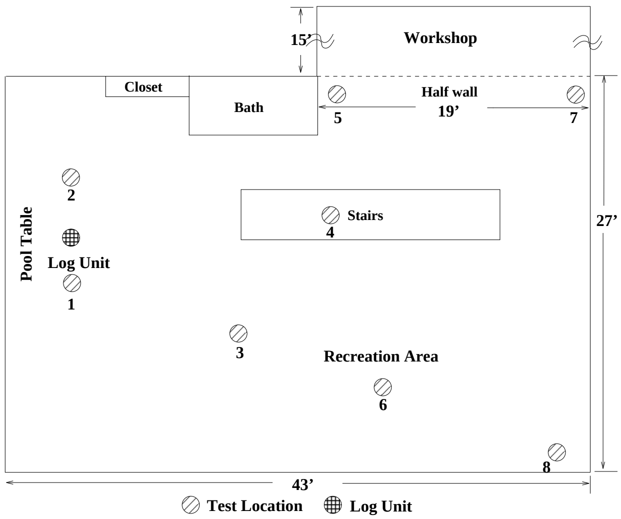

Figure 7: The layout of the indoor testbed (not to scale).

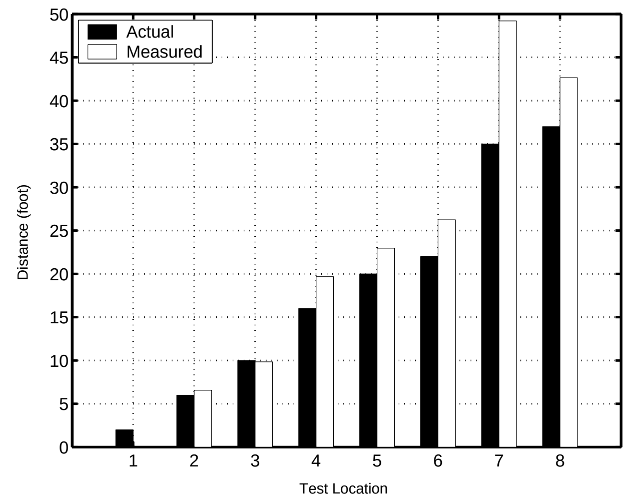

Figure 8: Performance of PinPoint in the first indoor testbed. The mobile unit had LOS and NLOS path to the log point at different locations.

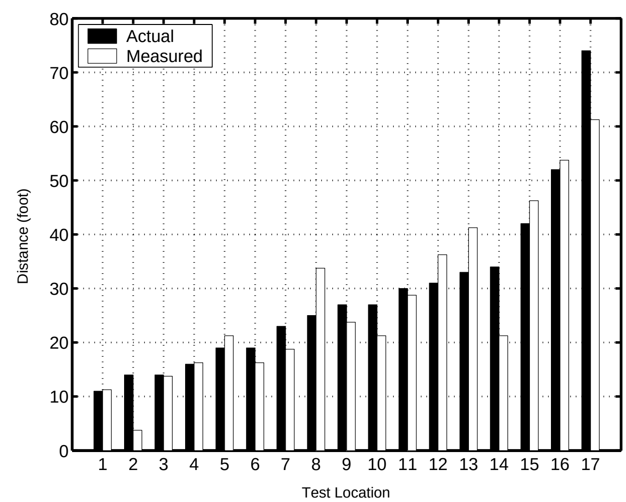

Figure 9: Performance of PinPoint in the second indoor testbed. The mobile unit did not have LOS to the log point at any location.

5.4 Mobile Performance

For this experiment, we moved the test unit with a constant speed emulating the typical user speed (about 2.5 feet per second). The user started seven feet away from the log unit and moved away till the distance became 160 feet. The user then remained stationary for 30 seconds (to see how fast the system stabilizes) and finally returned to the initial position with the same speed. Figure 12 shows the results. We can see from the figure that PinPoint can track the user in realtime. The reason is that the PinPoint exchange cycle is much faster than the user mobility rate.

6. RELATED WORK

Node localization has been the topic of active research and many systems have made their appearance in the past few years. The majority of existing localization systems consists of two basic phases: (1) ranging and (2) range combining. Ranging is the process of estimating node-to-node distances or angles. Range combining is the process of estimating node position using the estimated ranges.

In the rest of the section, we describe the different techniques for range estimation and range combining and how PinPoint relates to them. We do not explicitly differentiate between infrastructure based systems and ad hoc based systems as the focus of the paper is on the ranging part of PinPoint. We compare PinPoint to other location determination systems in Table 2.

6.1 Ranging Methods

The most popular ranging methods are: time-based methods, angle-of-arrival based methods, received signal strength based methods, and network connectivity based methods.

Time-based methods, like Time-of-Arrival (ToA) or Time-Difference-of-Arrival (TDoA), record the signal transmission time and the signal arrival time or the difference of arrival times. The propagation time can be directly translated into distance, based on the known signal propagation speed. These methods can be applied to different types of signals, such as RF and ultrasound. Traditional ToA based systems either require synchronized clocks, e.g. the GPS system [10], or uses the "echoing" method, e.g. [32, 22], where a node measures the roundtrip time of a signal transmitted to a remote node to estimate the distance to this node. Systems that require synchronized clocks are expensive to implement while systems based on the "echoing" method require $o(n^2)$ message exchanges to estimate the location of n nodes and suffer from more variance in the time measurement due to the echoing requirement.

The GPS system [10] is a widely used outdoor ToA-based system. The system is based on measuring the ToA of the signals transmitted from synchronized satellites. However, GPS receivers require a LoS to the satellite (and therefore

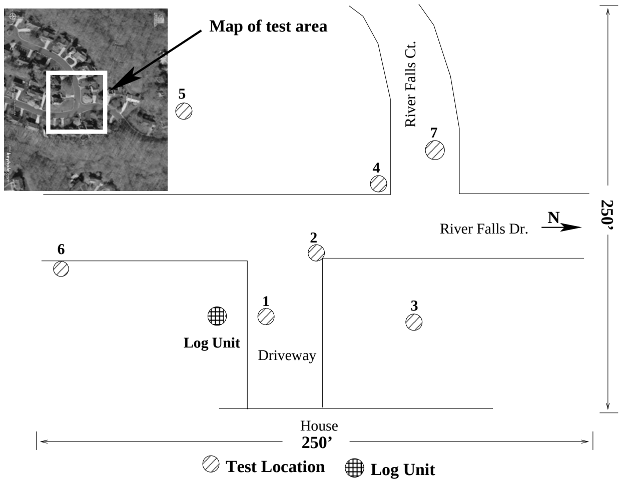

Figure 10: Sketch for the outdoor testbed (not to scale).

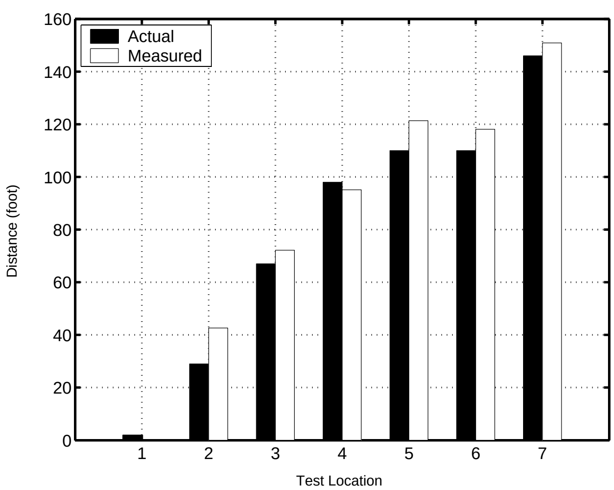

Figure 11: Performance of PinPoint in an outdoor environment.

their indoor usage is limited). The TPS system [7] is an example for TDoA-based systems. It computes ranges based on the TDoA of signals transmitted by three base stations with know locations to each node.

Ultrasound-based ToA systems, e.g. [23, 25], can achieve sub-feet accuracy but require dense deployment due to the limited propagation range of ultrasound signals. On the other hand, RF-based methods, e.g. [7] have a wider coverage range. However, their accuracy is lower than those of ultrasound based systems.

Recently, Ultra wideband (UWB) technology has been the focus of much research [31, 9]. The Precision Asset Location (PAL) system [11, 12] is a commercial UWB-based ranging system. The system consists of a set of four passive UWB receivers with known positions, and a collection of UWB active tags. The receivers are physically connected together in a ring topology using standard CAT-5 cables. Each active tag periodically transmits a packet. When the UWB

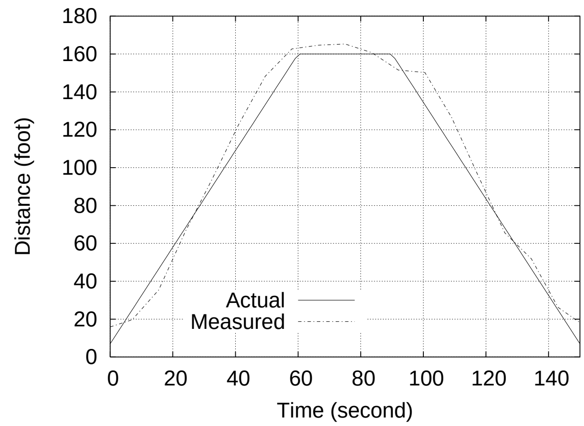

Figure 12: Performance of PinPoint for a mobile unit. The measured value is a moving average of five estimates. The user started seven feet away from the log unit and moved away till the distance became 160 feet. The user then remained stationary for 30 seconds and finally returned to the initial position with the same speed.

receivers receive the packet, differential time-of-arrival data from the four receivers is relayed to a computer which runs a nonlinear optimization algorithm to determine active tag coordinates. The PAL system requires an existing infrastructure and the receivers positions must be selected carefully to provide favorable siting of the active tags. With complex map geometry, determining the position of the UWB receivers is not an easy task. Moreover, the authors report that in a network of 100 nodes, the computation cost of one position update per second is very high and recommended a very low update frequency of one update per hour.

Angle-of-arrival (AoA) based methods [20] estimate the angle at which signals are received and use simple geometric relationships to calculate bearings to neighboring nodes with respect to node's own axis. Besides positioning, AoA methods can also provide orientation capabilities. However, AoA techniques require the nodes to be equipped with directional antenna or antenna array in order to measure the angle at which a signal arrives. These antenna arrays are expensive to implement and maintain.

Received-Signal-Strength-Indicator (RSSI) based methods [3, 34, 15] use the RSSI to estimate the device location. Theoretical and empirical models are used to translate the received RSSI into a distance estimate. For example in the *Horus* WLAN location determination system [34], the system works in two phases: offline phase and online phase. During the offline phase, the system constructs a radio map of the area of interest by storing the signal strength distributions from different access points at selected locations. During the online phase, the received signal strength vector is compared with the information stored in the radio map to determine the most probable user location. Such systems are attractive because they do not require specialized hardware (beside the wireless cards), however, they require calibration of the area of interest for building the radio map and thus are less suitable for scenarios that require rapid deployment.

Table 1: Testbeds' performance summary (all units in foot)

	Indoor	Indoor NLOS	Outdoor
Range Tested	37	74	146
Average Error	4.18	4.95	6.85
Standard deviation	4.44	4.19	4.35
95% Error	8.42	11	13.03
Maximum Error	14.2	12.75	13.64

Network connectivity based methods, e.g. [13] can be used for range estimation if the cost of range estimation hardware is expensive or if a sensor cannot receive signals from enough base stations. For example, the number of hops between two nodes can be used as an estimate of the range between these two nodes as in [19].

6.2 Range/Angle Combining Methods

Range combining is the process of estimating node position using the estimated node-to-node ranges. Trilateration, triangulation, multilateration, and landmark-proximity are the most known techniques for combining ranges.

Trilateration, e.g. as in [7, 19], locates a node by calculating the intersection of three circles. If the ranges contain error, the intersection of the three circles may not be a single point. Triangulation, e.g. as in [20], is used when the angle of the node instead of the distance is estimated, as in AoA methods. The node positions are calculated in this case by using trigonometric relationships. In this case, at least two angles are required. In multilateration, e.g. [26, 8, 14], the position is estimated from distances to three or more known nodes by minimizing the error between estimated position and actual position.

Proximity-based techniques are usually used when no range information is available. For example, the GPS-less system [4] employs a grid of beacon nodes with known locations; each unknown node sets its position to the centroid of the beacon locations it is connected to.

The PinPoint localization system belongs to the class of time-based methods. Compared to the other systems, e.g. the GPS system, it does not require any time synchronization between the nodes. The PinPoint technology explicitly handles the clock characteristics of different nodes and hence is able to achieve better accuracy. Moreover, it is a fully ad-hoc system with distributed localization algorithms running at every node. It does not require the nodes to be aware of their locations. From a power awareness perspective, all nodes transmit the same number of messages during the localization phase resulting in an even distribution of power consumption among the nodes. In summary, the PinPoint localization system provides fine-grained localization with an accuracy of few feet without requiring infrastructure support and with a wider coverage area. Table 2 compares PinPoint to other location determination technologies.

7. DISCUSSION

The current PP2 prototype proved that PinPoint is a promising technology that provides good accuracy without requiring any pre-planning for the area of interest. In this section, we discuss various aspects of the PinPoint design and implementation along with ongoing work to enhance the prototype.

7.1 Accuracy with Clock Rate

The current prototype achieves an average accuracy of four to six feet with a clock rate of 300MHz. Since the distance error is linear with respect to the accuracy of the clock, we expect that as the clock rate increases the average accuracy will increase.

7.2 Scalability

In the current prototype, we assume that all nodes are within the listening range of one another. For large networks, this may not be true. To solve this problem, clustering algorithms [33] can be used to split the network into subnetworks. The nodes within each subnetwork run the PinPoint protocol to calculate the local topology. Finally the subnetworks are merged, as needed, to obtain the global network topology.

7.3 Integration with the Current Wireless Standards

The hardware requirements of PinPoint are minimal and already exist in many of the current wireless standards. For example, PinPoint can be implemented on the current 802.11 [29] based cards. These cards already contain communication and timestamping modules. Computations can be carried out as part of the card's firmware or as an extension to the device driver. Note that since PinPoint uses one-way messages (i.e. does not require echoing of the transmitted messages) it is not affected by the contention-based mechanism of the 802.11 protocol as long as the timestamp is inserted when the station is clear to access the medium.

However, the current clock rate for the 802.11-based cards is still too coarse-grained to achieve the desired few feet target of PinPoint. For example, the Atheros [1] chipset uses a 40 MHz clock. This gives an accuracy of 25 ns. It is expected that future high-speed versions of the standards are going to include clocks with higher rates.

The PinPoint messages can also be piggy-backed with the normal frames exchanged.

7.4 Power Efficiency

The PinPoint prototype was not optimized for power efficiency. Another advantage of integrating PinPoint with the current wireless protocols is that it can benefit from the power optimization of the design of these cards.

The computational requirements of PinPoint are minimal as it is based on simple algebraic formulas. PinPoint requires $o(n)$ message exchanges to locate n nodes, compared to $o(n^2)$ messages exchanges for systems based on the "echoing" technique, e.g. [22, 32]. Note that during the information exchange phase of PinPoint, each message size is $o(n)$, compared to a constant message size of the "echoing" based systems. However, PinPoint approach has an advantage by reducing the overall system communication overhead by decreasing the number of messages sent. Moreover, larger messages size for the same amount of data allows for improved compression ratios. In addition, the distributed PinPoint design distributes the power consumption uniformally over all the nodes in the network.

7.5 Turn Scheduling Protocol

The current implementation of the PinPoint protocol is based on polling. We chose this algorithm for its simplicity. Among the other alternatives are:

Table 2: Comparison between PinPoint and other location determination technologies

System	Ranging Technology	Coverage Range	Accuracy	Rapid Deployment	Indoor/Outdoor
PinPoint	ToA	Hundreds of feet	Feet	Yes	Both
GPS [10]	ToA	Worldwide	Hundreds of feet	Yes (clients only)	Outdoor
Cricket [23]	ToA	Tens of feet	Sub-feet	No	Indoor
PAL (UWB) [12]	TDoA	Hundreds of feet	Feet	No	Indoor
TPS [7]	TDoA	Hundreds of feet	Feet	No	Outdoor
RADAR [3]	RSSI	Hundreds of feet	Feet	No	Indoor
Horus [34]	RSSI	Hundreds of feet	Feet	No	Both

- a TDMA-based protocol where each node is assigned a slot (especially with the time synchronization provided by PinPoint).
- a contention-based algorithm, like the one in the 802.11 standard, where nodes contend for the medium. When a node gets access to the medium, it starts its message exchange turn. Such protocols have mechanisms to allow fair sharing of the medium between the nodes. This protocol may be useful for relatively static scenarios, e.g. sensor networks, where delays in accessing the medium does not affect location estimation. Note also that PinPoint does not require message exchanges to be equally spaced in time.

Another aspect of the polling-based protocol is that since we assume that all nodes are within the listening range of each other, a node can start its turn immediately after the exchange cycle of the node that precedes it in the node-ID order. This eliminates the need for the master node. The nodes may come pre-configured with the transmit order, or a randomized algorithm can be used where each node broadcasts its ID so that each node knows the ID of the node that precedes it in the order.

7.6 Time synchronization

Although not discussed in the paper, a byproduct of the PinPoint protocol is that each node can determine the clock characteristics (drift and offset relative to itself) of the nodes in its neighborhood with an accuracy on the order of its clock tick. This allows any subset of the nodes to carry out a synchronized action within the network.

7.7 Applications

In addition to the traditional location-based applications (e.g. location-based routing [18], ubiqutious computing [6], providing node-ID [27], and sensor networks [7]) PinPoint provides the enabling technology for a set of new applications that are ad-hoc in nature and require rapid deployment. Examples of these applications are emergency-personnel tracking systems and military-personnel tracking in the battlefield.

8. CONCLUSIONS

The contribution of this paper is threefold: (1) we present the design of a distributed time-of-arrival based location technology that does not require clock synchronization and incur $o(1)$ message exchanges per node to locate all other nodes; (2) we present a prototype implementation for the proposed design and discuss its practical aspects; (3) we evaluate the performance of PinPoint using the developed prototype.

The PinPoint protocol is a distributed time-based protocol that runs on all the nodes in a PinPoint-enabled network. For a network of n nodes, PinPoint requires a ***constant*** number of message exchanges per node. Each node measures the transmission and reception timestamps for certain events with its local clock which may have its own offset and drift rate. We showed how PinPoint mathematically compensates for these clock differences in order to arrive at a very precise timestamp recovery that in turn leads to a precise distance determination. Moreover, each node is able to determine the clock characteristics of other nodes in its neighborhood and thus network synchronization can be achieved.

We also presented a prototype implementation for PinPoint that uses off-the-shelf clock modules and discussed the practical issues in implementing the mathematical framework and how PinPoint handles the different sources of error affecting its accuracy. We evaluated the prototype in typical indoor and outdoor testbeds. Our results show that PinPoint can achieve accuracy up to four to six feet on the average for indoor and outdoor environments. Moreover, PinPoint does not require any pre-planning for the area of interest and thus is suitable for scenarios that require accurate rapidly deployable localization systems.

9. ACKNOWLEDGMENTS

This work has been supported in part by a grant from the Maryland Technology Development Corporation (TEDCO) and the University of Maryland.

10. REFERENCES

[1] http://www.atheros.com.

[2] R. Azuma. Tracking Requirements for Augmented Reality. *Communications of the ACM*, 36(7):50–51, July 1993.

[3] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In *IEEE Infocom 2000*, volume 2, pages 775–784, March 2000.

[4] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5):28–34, October 2000.

[5] S. Capkun, M. Hamdi, and J. P. Hubaux. GPS-free positioning in mobile ad-hoc networks. *Cluster Computing Journal*, 5(2):118–124, April 2002.

[6] G. Chen and D. Kotz. A Survey of Context-Aware Mobile Computing Research. Technical Report Dartmouth Computer Science Technical Report TR2000-381, 2000.

[7] X. Cheng, A. Thaeler, G. Xue, and D. Chen. TPS: A Time-Based Positioning Scheme For Outdoor Sensor Networks. In *Proceedings of IEEE Infocom*, volume 4, pages 2685–2696, 2004.

[8] L. Doherty, K. Pister, and L. E. Ghaoui. Convex Position Estimation in Wireless Sensor Networks. In *Proceedings of IEEE Infocom*, volume 3, pages 1655–1663, 2001.

[9] G. Durisi and G. Romano. Simulation analysis and performance evaluation of an UWB system in indoor multipath channel. In *IEEE Conference on Ultra Wideband Systems and Technologies*, pages 255–258, 2002.

[10] P. Enge and P. Misra. Special issue on GPS: The Global Positioning System. *Proceedings of the IEEE*, 87(1):3–172, January 1999.

[11] R. J. Fontana and S. J. Gunderson. Ultra-wideband precision asset location system. In *IEEE Conference on Ultra Wideband Systems and Technologies*, pages 147–150, May 2002.

[12] R. J. Fontana, E. Richley, and J. Barney. Commercialization of an ultra wideband precision asset location system. In *IEEE Conference on Ultra Wideband Systems and Technologies*, pages 369– 373, November 2003.

[13] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-Free Localization Schemes for Large Scale Sensor Networks. In *9th ACM MOBICOM*, pages 81–95, 2003.

[14] X. Ji. Sensor Positioning in Wireless Ad-hoc Sensor Networks with Multidimensional Scaling. In *Proceedings of IEEE InfoCom*, volume 4, pages 2652–2661, 2004.

[15] P. Krishnan, A. Krishnakumar, W. H. Ju, C. Mallows, and S. Ganu. A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks. In *IEEE Infocom*, volume 2, pages 1001–1011, March 2004.

[16] J. Krumm et al. Multi-camera multi-person tracking for Easy Living. In *3rd IEEE Int'l Workshop on Visual Surveillance*, pages 3–10, Piscataway, NJ, 2000.

[17] A. M. Ladd, K. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-Based Location Sensing using Wireless Ethernet. In *8th ACM MobiCom*, pages 23–28, Atlanta, GA, September 2002.

[18] J. C. Navas and T. Imielinski. Geocast- Geographic Addressing and Routing. In *3rd ACM MOBICOM*, volume 3, pages 66–76, 1997.

[19] D. Niculescu and B. Nath. Ad-hoc Positioning System (APS). In *IEEE GlobeCom*, volume 5, pages 2926–2931, 2001.

[20] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS) Using AoA. In *Proceedings of IEEE Infocom*, volume 3, pages 1734– 1743, 2003.

[21] R. J. Orr and G. D. Abowd. The Smart Floor: A Mechanism for Natural User Identification and Tracking. In *Conference on Human Factors in Computing Systems (CHI 2000)*, pages 1–6, The Hague, Netherlands, April 2000.

[22] V. N. Padmanabhan and L. Subramanian. Determining the Geographic Location of Internet Hosts. In *Proceedings of ACM SIGMETRICS*, pages 324–325, 2001.

[23] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *6th ACM MOBICOM*, pages 32–43, Boston, MA, August 2000.

[24] T. S. Rappaport. *Wireless Communications Principles and Practice*. Pearson Education, second edition, 2002.

[25] A. Savvides, C. C. Han, and M. Srivastava. Dynamic Fine-Grained Localization in Ad-hoc Networks of Sensors. In *7th ACM MOBICOM*, pages 166–179, 2001.

[26] A. Savvides, H. Park, and M. B. Srivastava. The Bits and Flops of The n-hop Multilateration Primitive for Node Localization Problems. In *the first ACM international workshop on Wireless Sensor Networks and Applications*, pages 112–121, 2002.

[27] C. Schurgers, G. Kulkarni, and M. Srivastava. Distributed On-Demand Address Assignment in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(10):1056–1065, January-March 2002.

[28] S. Tekinay. Special issue on Wireless Geolocation Systems and Services. *IEEE Communications Magazine*, April 1998.

[29] The Institute of Electrical and Electronics Engineers, Inc. IEEE Standard 802.11 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. 1999.

[30] R. Want, A. Hopper, V. Falco, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.

[31] M. Welborn. System considerations for ultra-wideband wireless networks. In *Proceedings of IEEE Radio and Wireless Conference (RAWCON)*, pages 5–8, 2001.

[32] J. Werb and C. Lanzl. Designing a Positioning System for Finding Things and People Indoors. *IEEE Spectrum*, 35(9):71–78, September 1998.

[33] A. Youssef, M. Youssef, M. Younis, and A. Agrawala. The Overlapped K-hop (OK) Clustering Algorithm. Technical Report CS-TR-4735, Department of Computer Science, University of Maryland College Park, July 2005.

[34] M. Youssef and A. Agrawala. The Horus WLAN Location Determination System. In *Third International Conference on Mobile Systems, Applications, and Services (MobiSys 2005)*, pages 205–218, June 2005.