Asset Localization in Data Centers Using WUSB Radios

N. Udar[1], K. Kant[2], and R. Viswanathan[1]

[1] Southern Illinois University, Carbondale, IL
[2] Intel Corporation, Hillsboro, OR

Abstract. Asset tracking is a critical problem in modern data centers with modular and easily movable servers. This paper studies a wireless asset tracking solution built using wireless USB (WUSB) radios that are expected to ubiquitous in future servers. This paper builds on our previous work on direct ranging using WUSB radios and studies algorithms for data center wide localization of all rack mounted servers. The results show that it is possible achieve very low error rates in localization in spite of very stringent constraints (i.e., each server being just 1.8" high) by exploiting the properties of the data center environment.

Keywords: Ultra Wideband (UWB), Wireless USB, data centers, channel model, localization, Maximum Likelihood Estimation.

1 Introduction

Data centers form the backbone of modern commerce and continue to grow both in terms of number of servers and number of servers per unit volume of data center space. Modern data centers sport rows upon rows of "racks", each with certain number of standard size "slots" where various "assets" (servers, routers, switches, storage bricks, etc.) can be inserted. Fig. 1 shows a typical row of a data center. Tracking these assets has repeatedly been cited as among the top 5 issues facing IT administrators in large data centers and other IT environments. As assets become easier to move – by virtue of smaller sizes, modular structure, hot plug-in/plug-out capabilities – they indeed tend to change locations more frequently. There are several reasons for assets to move around: (a) replacement of old/problematic equipment and/or addition of new equipment and resulting reorganization, (b) manual reorganization of assets to handle evolving needs and applications, (c) reorganization driven by power and thermal issues which keep becoming more and more severe, (d) removal followed by reinsertion for miscellaneous reasons including SW patching, etc. One whole, the assets don't necessarily move very much, but in a large data center, even a limited movement could become very cumbersome to manage manually. In particular, trivial solutions such as those requiring personnel to log each moved asset in a spreadsheet/database have not worked well in the past. Several vendors including Sun and HP have devised new solutions, which points to the importance of the problem.

A. Das et al. (Eds.): NETWORKING 2008, LNCS 4982, pp. 756–767, 2008.

Fig. 1. Racks in a data center

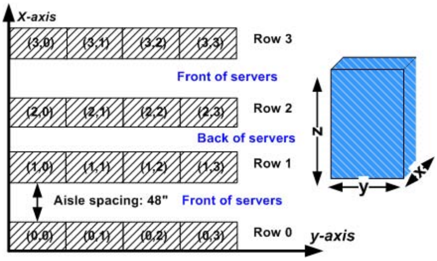

Fig. 2. Schematic Rack Layout

From the survey paper [1] on indoor positioning techniques, it can be concluded that techniques based on wireless local area network (WLAN) are perhaps reasonable to locate racks in the data center, but too crude for locating servers. Techniques based on surface acoustic wave (SAW) can provide good accuracy but would require a network of SAW sensors to be deployed and managed. Ultra wideband (UWB) based solutions can provide good accuracy, but any external infrastructure would add to cost and management issues. In this paper, a wireless universal serial bus(WUSB) based solution is exploited that uses UWB radios integrated into the server and does not rely on any external infrastructure. HP has developed a solution based on an array of passive RFID tags attached to each server [2]. The solution requires one radio frequency identification (RFID) reader per server which communicates with the rack level data collector. Each RFID reader has a directional antenna mounted on a motorized rack and each rack has a sensor controller aware of its position. Although an accuracy exceeding 98% is claimed for this system, the complexity and cost of the system are expected to be prohibitive.

In this paper we explore an automated asset tracking solution by exploiting the WUSB that is expected to replace *wired USB* in the near future. The basic assumption is that WUSB radios will be integrated with various assets and could form a sort of mesh fabric which can be exploited for a variety of low-bandwidth applications including asset tracking. There are many reasons why such an approach is attractive:

1. WUSB uses UWB radio as its physical layer which is known to have excellent localization properties.
2. Given its role as a wired USB replacement, WUSB radio cost should come down rapidly, which makes the solution inexpensive.
3. Since WUSB radios are assumed to be integrated in each server, such a solution does not require any external infrastructure. Additional infrastructure is usually highly undesirable from an IT personnel's perspective.

In this paper we focus on "plugged-in" assets that have our integrated solution implemented. The assets don't necessarily have to be "booted up" or even powered, just plugged in. To allow for this, we propose to implement asset localization in a server's baseboard management controller (BMC), which is supposed to be always operational irrespective of the state of the server.

The most challenging aspect of the data center localization problem is the need for very high accuracy localization. *The typical "one meter" accuracies that localization methods can typically achieve with significant effort are about 2 orders of magnitude worse than what we need in this application.* Furthermore, we would like to do this without any additional infrastructure. The outline of the paper is as follows. In section 2, we present the configuration of modern data centers and discuss how a wireless USB based scheme can walk through the entire data center and localize individual plugged in assets. In section 3 we propose the method of maximum likelihood identification (MLI) for localization of servers and show that the performance of proposed method far exceeds the performance of traditional hyperbolic positioning (HBP). The novelty of the MLI solution is the exploitation of geometric properties of the data center environment to obtain high accuracy localization. The performance of MLI is analyzed in section 4. The simulation results show that it is possible to achieve good accuracy in a data center environment using MLI. Section 5 concludes the discussion.

2 Configuration and Assumptions

The geometric aspects of server arrangement in a data center are crucial to our solution; hence are described briefly here. The racks, as shown in Fig. 1 are 78" high, 23-25" wide and 26-30" deep and are arranged in rows. The rack rows are arranged in pairs so that the servers in successive odd-even row pairs face one another. This creates alternating hot and cold aisles (backs and fronts of servers) and helps make cooling more efficient. The width of these hot and cold aisles is generally different and needs to be accounted for. Fig. 2 shows a simplified top view of this arrangement where the wider aisles are the cold aisles (fronts of servers). The Fig. 2 also establishes a coordinate system that we will be using in our analysis. The x-axis denotes the row index and the y-axis denotes the rack position in the row. The z-axis (not shown in the figure) is along the height of the rack. For example,$(0, 1, 1)$ denotes that the server is in row 0, rack 1 and at position 1 in the rack. The positions in the rack are labeled as $1, 2, ..N$ starting from top of the rack.

In this paper, we focus on the popular "rack mount" servers that go directly into the racks as shown in Fig. 1. Obviously, the server thickness decides how many servers can fit in a rack. The standard measure of thickness is "U" (about 1.8"). Consequently, a single rack can take up to 42 1U "assets". The increasingly popular "Blade servers" go vertically in chassis that in turn fit into racks will be addressed in future work.

It is assumed that each plugged asset has an integrated WUSB radio accessible from the management controller in the pre boot or post boot environments. We assume that each rack has at least 2 plugged in servers and the racks are arranged in a rectangular pattern – localization with geometries other than a rectangle is beyond the scope of the current paper.

We use a domain dependent medium access (MAC) protocol for localization based on Wimedia [3] where MAC domain is referred as a Piconet and the master of Piconet is called a Piconet Controller (PNC). The PNC guarantees access to the medium on a TDMA basis and up to 255 slaves are allowed in the protocol and slaves are allowed to interact with each other in the contention access period.

We assume that there are at least 3 servers with known locations. The WUSB radios of these servers are assumed to be manually configured and these servers are prohibited from being moved around. (If even the BMC of one of these servers fails, it is possible to "elect" a suitable substitute from other already localized servers – however, this is just a matter of detail that we omit here.) In particular two known servers in rack(0,0) and one in rack(0,1) is considered. One of the known server in rack(0,1) is hard coded to be the PNC. All other radios, when turned on, are in the listening mode. We assume that every WUSB radio could operate in a range of power between a minimum and maximum value. The PNC starts with minimum power and gradually increases it (up to the maximum power) so that it can reach a few servers around it in the same or adjacent rack.

For a distance estimate between two nodes time of arrival(ToA), time difference of arrival or receiver strength may be used [4]. The fundamental mechanism that we used for for distance estimation is the ToA of the strongest signal. Here, we briefly summarize our previous results on achievable ranging accuracy [5]. The results show that it is possible to achieve accuracies of 0.05-0.65 meter with direct ranging [5]. In any case, such accuracies are far from adequate for our purposes; therefore, we need additional mechanisms to reduce the error substantially. These mechanisms include bias estimation, multiple measurements, exploitation of geometry and effective estimation procedures. In particular, based on a large number of measurements, we found that the standard deviation of ToA determined distance to be less than 0.32 meters in all cases (i.e., variance less than 0.1 m^2). We also found that the variance is largely independent of the distance for the range of values that we are interested in (a few meters). In particular, we conservatively assume the worst case variance of 0.2 (double the maximum observed in our experiments). Consequently, we expect the errors from our simulation results to be significantly overstated than those expected from real measurements.

2.1 Localizing Multiple Nodes

In this section we discuss how we can systematically *all* nodes in a data center starting with the known location of a few servers. There are in fact two distinct phases here: (a) cold start, and (b) steady state. The cold start localization phase occurs initially when none of the servers other than the known ones have their locations known. It may also be invoked by the operator on major changes or perhaps automatically once per day (or more frequently) to ensure that no changes are missed. The steady state phase involves localization of one (or at most a few) servers when the presence of new radios is detected or some radios are removed or shut down.

The cold-start localization starts with localization of unknown servers in rack (0,0). For rack (0,0) localization, two servers at known locations (known servers) in rack (0,0) and one known server in rack (0,1) is used as reference nodes. During rack(0,0) localization, all the unknown nodes in rack(0,0) and one unknown node in rack(0,1) and 2 unknown nodes in rack(1,1) are localized. These unknown nodes are then used as reference nodes for corresponding rack localization. Rack(0,1) onwards during localization 2 reference nodes from the current rack and one reference node from the previous rack. Rack(0,1) localization onwards exactly 2 nodes in the next rack and 2 nodes in the opposite rack are localized. Thus the row 0 localization is complete. The odd numbered racks need not localize the servers in the next rack except the ones present in the beginning of the first rack. For example, rack(1,1) uses two reference nodes from its rack and one from rack(0,0) and localizes one server in rack(2,0). Thus, all the servers in the data center are localized from right to left one row at a time.

Each server maintains a local grid map of its neighbors. Local grid map refers to the map of position of neighbors. As the localization is performed, each server updates its map to indicate which neighbors are plugged-in.

A critical element in cold-start localization is the avoidance of servers in racks that we are currently not interested in localizing. The fundamental assumption in the approach above is that the transmit power is chosen low enough so that the signal does not reach across more than one rack in either direction. Obviously, this cannot be guaranteed by power control alone. Instead, we need to examine each distance estimate and determine whether the responder is more than one rack away. That is, the problem requires doing “macro localization” in order to enable systematic localization discussed above. This is another place where we take advantage of the regular geometry of the racks. Since the rack dimensions are known and the racks are regularly placed, macro-localization is relatively straightforward. In particular, if we have 3 known nodes and distances from them to an unknown node, simple geometric calculations can easily tell us with high probability which rack the unknown node belongs to. In cases where there is ambiguity, multiple measurements can be used for disambiguation much in the same way as discussed in section 4.2 for correctly locating new set of reference nodes. For brevity, we omit the details of this procedure.

Once the cold start localization is finished, monitoring in steady state is relatively simple because several Piconets operating simultaneously monitor their surrounding nodes in order to update any changes in the locations of the nodes.

3 Localization Methods

In this section, we consider the issue of accurately estimating the position of a single unknown node based on distance measurements from a number of known nodes. We propose the maximum likelihood identification method and compare with the traditional hyperbolic positioning method.

3.1 Hyperbolic Positioning

Hyperbolic Positioning technique requires common time reference only between reference nodes and does not rely on the synchronization between references nodes and the target node [6]. The position of the target node is determined based on the time difference of arrival from two reference nodes. Let us determine the 2-D position of the target node $T(x_0, y_0)$ using k reference nodes. Each of the reference node makes the range measurement based on TOA with the target node denoted as $D_{1T}, D_{2T}....D_{kT}$. Let us assume that the reference node share a common time reference and the clock at the target node T is delayed by δ. Then it is observed that the difference of range measurements between any pair of node removes the delay. The range measurement between any two nodes is given as

$$D_{kT} - D_{lT} \ = c(\tau_{kT} + \delta) - c(\tau_{lT} + \delta) \tag{1}$$

where τ_{kT} denotes the TOA measured by reference node k from the target node. The position of target node T in the 2-D space is determined by the intersection of hyperbola using 3 reference nodes. However, when range measurements incur errors due to multi path and noise HBP could show large errors in localization [7]. In fact, for our problem ML identification shows much better accuracies than HBP localization, as will be shown in section 4.

3.2 Maximum Likelihood Identification

Maximum likelihood (ML) testing is a well-known technique for deciding on which one of the hypothesized models a measurement came from. Since in our problem, we only have 42 discrete possible positions for a server in a rack, the maximum likelihood identification (MLI) deals with identifying which one of the several discrete positions is the correct one. For a more formal description of MLI, consider two nodes at the known locations in rack 1 and one in rack 2 (this can be easily extended to p transmitters). Each of the known nodes measure the distances using TOA from an unknown node at (m, l, n) where m denotes the row number, $m \in (0, 1..M)$, l denotes the rack number $l \in (0, 1, ..L)$, and n denotes the position of the node in the rack, $n \in (0, 1, 2...N - 1)$.

Let V out of N-2 (N-2 possible unknown positions for rack 1) possible positions are filled by plugged-in servers. Each of the V nodes determines its position based on its range estimates from the three known nodes.

Let us consider the detection of the location of one of these V nodes say, node u. Since the location of node u is unknown , it hypothesizes its location to be any one of the possible locations and forms $N - 2$ likelihood functions, one for each hypothesis, based on the range measurements (r_{1u}, r_{2u}, r_{3u}). For forming the likelihood function, we assume that a range estimate r_{iu} is distributed as Gaussian with zero bias (that is, mean equal to the true distance between the reference node i and the node u, d_{iu}) and variance $\sigma^2 = N_0/2$. This model assumes line of sight propagation, which may be reasonable when the transmitters and the receivers are in close proximity of each other. In future work we relax this assumption and introduce bias in the range estimates. The node u estimates its location based on the maximum likelihood rule, i.e., decide location $(m, l, \hat{n})$,

$$(m, l, \hat{n}) = \underset{n}{arg} \quad max \quad p\left(r_{1u}, r_{2u}, r_{3u} | H_n\right) \tag{2}$$

where the row index m is assumed to be correct as we localize one row at a time and the rack index l is decided based on the geometry of the racks and the three range estimates as mentioned in section 2.1. Hence, MLI searches only over the rack index n.

Since each plugged in server is classified as occupying one of the allowed server positions, several types of errors could happen. A server could be misclassified to be occupying a wrong position where (i) there actually exists another server or (ii) there exists no server. Another situation occurs when two or more nodes claim the same location. This situation could be resolved by subsequent range measurements. For the proposed location estimation system to be successful, it is imperative that the probability of error in the classification of a node is extremely small, especially because the estimation of nodes occurs in successive stages, relying on previous estimates, one rack at a time and one row at a time. We briefly mention the calculation of probability of correctly identifying a node location for the maximum likelihood rule. By expanding eq. 2 and by throwing out the terms that do not depend on n, the equivalent rule picks the maximum of $Z_{n,u} = \langle \mathbf{r}_u, \mathbf{d}_n \rangle - E_n/2$, where $\mathbf{r}_u = \left(r_{1u}, r_{2u}, r_{3u}\right)^T$, $\mathbf{d}_n = (d_{1n}, d_{2n}, d_{3n})$.

Here, $\langle -, - \rangle$ denotes the inner product between two vectors and $E_n = \langle \mathbf{d}_n, \mathbf{d}_n \rangle$. Given that the true location of unknown node say, *(m,l,u)*, the probability of correct classification is the probability that $Z_{u,u}$ is the largest among all possible $Z_{n,u}$, Finally, a closed expression for error probabilities is similar to calculating the error probabilities, in the detection of M-ary signals in additive white Gaussian noise with arbitrary signal set [8]. It is well-known that the exact error probability is difficult to calculate but tight upper and lower bounds can be obtained.

Notice that the maximum likelihood rule does not require the knowledge of the variance, as long as the variance is the same for any estimated range. If the variances of the range estimates depend on the distance, then the maximum likelihood rule would require the values of these variances. However, for short distances, spanning the height of a rack or an adjacent rack, a single variance assumption may be valid as a first approximation [5]. For assessing the effectiveness of the algorithm we will use MATLAB simulations and union bound discussed in section 3.3.

3.3 Error Bound

For any countable set of events, $A_i, i = 1, 2, ...$, with corresponding probabilities $P[A_i]$, the probability of union of these events is no greater than the sum of the probabilities of individual events [8]. That is, $P(\bigcup_i A_i) \leq \sum_i P[A_i]$.

A node u can be uniquely identified by specifying the distances between the node and the reference nodes, $\mathbf{d}_u$. Then, the "distance" between two nodes u and i, denoted $D_{u,i}$, is characterized by $D_{u,i}^2 = \sum_{k=1}^{3} \left(d_{ku} - d_{ki}\right)^2$

From the detection of known signals in Gaussian noise [8], the probability of error in distinguishing between two locations, u and i, given that the data came from the location u, is given by $P\left(\varepsilon_{ui}|\mathbf{d}_u\right) = Q\left(D_u, i/\sqrt(2N_0)\right)$ where $Q(x)$ is the complementary cumulative distribution function of the standard Gaussian at point x. Since a node u can be misidentified as any one of the other possible nodes $i \neq \; u$, the probability of error in misidentifying u is

$$P\left(\varepsilon|\mathbf{d}_u\right) = P\left[\bigcup_{i\neq u} \varepsilon_{ui}|\mathbf{d}_u\right] \leq \sum_{i\neq u} P\left(\varepsilon_{ui}|\mathbf{d}_u\right) \tag{3}$$

Since N-2 corresponds to the maximum number of possible unknown servers assuming the rack is completely filled, the average probability of error is obviously $P\left(\varepsilon\right) = \frac{1}{N-2}\sum_u P\left(\varepsilon|\mathbf{d}_u\right)$

3.4 Comparison of HBP and MLI

To analyze the performance of HBP and MLI methods, two racks in a data center are considered. The problem of localization of 1U servers in rack(0,0) is studied. The two servers in known locations in the top and middle of rack(0,0) (known servers) and one known server in the middle of rack (0,1) is considered. We analyze the performance of HBP and MLI methods by estimating the probability of incorrect identification of server position, P_e.

To simulate the range estimate obtained via UWB radio, *i.i.d* Gaussian errors are added to the true distance between the two nodes. Given the dimension of the rack, there are 42 IU servers possible in rack(0,0) when rack is completely filled, out of which a maximum of 40 server positions may be unknown. Let the three reference nodes for rack(0,0) simulation are labeled as 1,2,3 respectively and the unknown server is at the position u, then the error distance metric for MLI method is formed as

$$errDist(i) = sqrt((r_{1u} - d_{1i})^2 + (r_{2u} - d_{2i})^2 + (r_{3u} - d_{3i})^2), i = 1, 2, ..., V \tag{4}$$

The position of the unknown node is localized as that position for which the errDist metric is minimum among the V metrics. Ideally, when the range measurements are exact, the correct decision $i = u$ is made with the corresponding errDist metric being zero. In HBP method, the position of each of the unknown server is estimated using the intersection of hyperbolas. The position of the unknown server is determined in the nearest neighbor sense by finding the minimum distance between the estimated position and all the possible server positions.

In Fig. 3 the performance of HBP method is compared with MLI method. Assuming zero bias in the range estimates, the average probability of error in identification of locations of unknown servers in a rack is plotted as a function of variance of ranging errors. In Fig. 3 two graphs are shown for MLI method, one for error bound described earlier and the other based on simulation procedure outlined here. It can be seen that the MLI method significantly outperforms the HBP method. The MLI method requires the calculation of distances between the three known nodes and all possible hypothesized positions for the unknown node whereas the HBP method requires the finding of the nearest neighbor node, nearest to the estimated position obtained through hyperbolic intersection. Hence, the MLI method requires a slightly more computation per sever localization. However, since computations involve mainly the calculations of Euclidean distances and ascertaining the minimum of a set of numbers, these computations can be done quickly with a reasonable inexpensive processor. Certainly, the proposed localization algorithm dictates an accuracy that is not attainable with the HBP method. It is observed that, as long as the variance is below 0.8 (corresponding to a probability of error of less than 0.1), the union bound is extremely close to the simulation estimate. In rest of the paper we analyze the performance of the MLI method only.

4 MLI Simulation Results

4.1 Single Rack Simulation

In this section, we further analyze the performance of MLI method for a single rack. We also introduce the threshold rule based on the magnitude of error distance metric in eqn (2).

Fig. 4 shows the error probability of ML method as a function of node position, for two different variances of ranging errors. It is observed that the nodes at the top and bottom of a racks, experience smaller errors than the nodes at the center of the rack. Of course, each of the two end nodes, the top and the bottom, has only one immediate neighbor, and this could have contributed to the low error rate.

The probability of incorrect identification can be lowered by increasing the number of reference nodes or using multiple measurements. In Fig. 5, the simulations show that by using 2 measurements of range estimates, P_e is lowered by two orders of magnitude. Similar reduction could also be obtained using increased number of reference nodes. However, more reference nodes require that we start

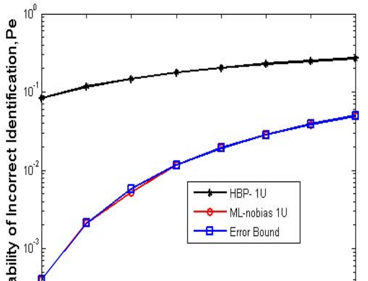

Fig. 3. Comparison of HBP and ML Detection Error

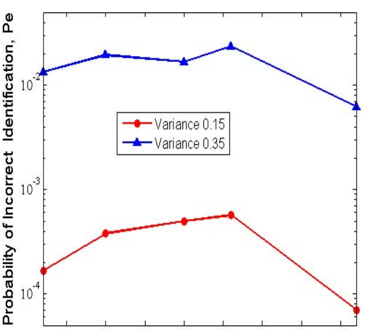

Fig. 4. Prob. of Incorrect Identification vs. Server Position

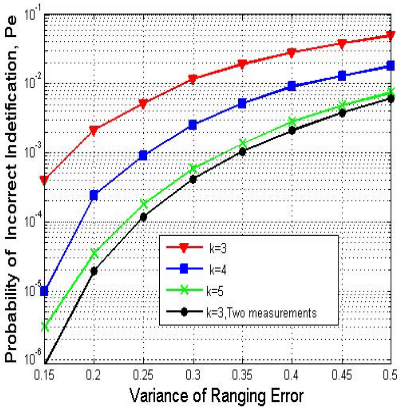

Fig. 5. Performance with More Reference Nodes & Measurements

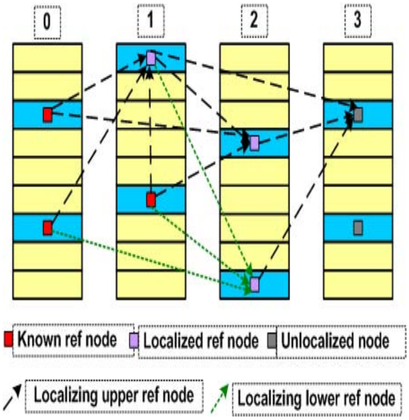

Fig. 6. Illustration of Error Propagation

out with more servers in fixed and known locations, which is undesirable. A downside to more measurements is increase in localization time.

The distance between two adjacent 1U servers is 1.8 inches. When the magnitude of minimum of V errDist metrics exceeds a certain threshold value (which, by simulation study,was fixed at 0.9, for a range estimate variance of 0.2) the probability of incorrect identification increases substantially. Such error events are unacceptable when reference nodes are identified for subsequent localizations. Therefore, during the next rack reference node localization, if the magnitude of the minimum of errDist metrics exceed the threshold value, then two range measurements to each known reference nodes are made in order to increase the ranging accuracy and reduce the errDist metric.

4.2 Error Propagation in Large Data Centers

The previous section showed that we can achieve excellent localization accuracy in single rack. In this section we address the question of how the localization error will change as we proceed from rack to rack.

Fig. 6 shows how the localization proceeds across racks in row 0. For simplicity, we only show localization of reference nodes. We need to choose 2 reference nodes in successive racks starting with known racks in racks 0 and 1 (shown in red squares). Each such localization depends on 3 previously localized servers as shown. The error propagation could then be characterized via a recursive set of equations as discussed below. This simple analysis suggests that we arbitrarily pick two of the localized nodes in a rack as reference nodes. In fact, because of their importance, we pick reference nodes a bit more carefully, by considering error distance metric and position within the rack. Nodes with small values of error distance metric are likely to be located more precisely and hence make a better choice for reference nodes. The position is important since we don't want both reference nodes to be close together. These aspects should only make the reference node choice more robust and hence result in even less error propagation.

Let P_{ei} denote the probability of error in identifying an unknown node in a rack, given i reference nodes in incorrect positions where $i \in \{0, 1, 2, 3\}$. Using simulations and the MLI method we estimate values for P_{ei} as:

$$P_{e0} = 1.920\text{e-}5, \quad P_{e1} = 1.430\text{e-}3, \quad P_{e2} = 7.60\text{e-}1, \quad P_{e3} = 8.20\text{e-}1 \tag{5}$$

Not surprisingly, the probability of error goes up sharply with number of nodes in error. The simulation also examined instances where the estimated server position is off by only ±1 slot. The simulation study shows that, except when all the three reference nodes are in error, whenever an error occurs in the identification of a node, the MLI method picked one of the immediate neighboring nodes.

Let $P_i(n)$, $i = 1, 2$ denote the probability of error in locating the ith reference node in rack n. Now referring to Fig. 6, we can write the following equations. First, the initial conditions are:

$$P_1(0) = P_2(0) = 0, \qquad P_1(1) = P_{e0}, \quad P_2(1) = 0 \tag{6}$$

Now for $n > 1$, we can write the following recursive equation for $P_1(n)$.

$$\begin{aligned} P_1(n) = {} & [1-P_1(n-1)][1-P_2(n-1)]\,\{[1-P_1(n-2)]P_{e0} + P_1(n-2)P_{e1}\} \\ & + \{P_1(n-1)[1-P_2(n-1)] + [1-P_1(n-1)]P_2(n-1)\}\,\{[1-P_1(n-2)]P_{e1} + P_1(n-2)P_{e2}\} \\ & + P_1(n-1)P_2(n-1)\,\{[1-P_1(n-2)]P_{e2} + P_1(n-2)P_{e3}\} \end{aligned}$$

The equation for $P_2(n)$ is almost identical and is omitted. Since $P_1()$ and $P_2()$ differ only at rack 1, we simply assume the worst case scenario (which applies to P_1) and drop the subscript from the above equation. We now show, via an inductive proof, that $P(n)$ remains "small" (i.e., of the order of P_{e0} or smaller) even as n grows. For this, let us inductively assume that both $P(n-2)$ and $P(n-1)$ are of the order of P_{e0}. Certainly, this is true for $P(0)$ and $P(1)$. Therefore,

$$P(n) \approx P_{e0} + P(n-2)P_{e1} + 2P(n-1)\{P_{e1} + P(n-2)P_{e2}\} \approx P_{e0} + 3P_{e1}P(n-1) \tag{7}$$

Clearly, $P(\infty) = P_{e0}/(1 - 3P_{e1})$. Now since P_{e1} is itself quite small (1.43e-3), it follows that $P(n) \approx P_{e0}$ and there is no appreciable error propagation. More generally, so long as P_{e1} can be kept reasonably small, there is no error propagation of any consequence. In particular, for the values given above, $P(\infty)$ =1.9283e-5.

Once row 0 localization is completed, two reference nodes in (rack 0, row 1), are localized by utilizing range measurements from known nodes in (rack 0, row 0). After this, row 1 localization proceeds exactly as in row 0.

In order to verify analytic calculations above, we simulated localization for all 16 racks in a 4x4 data center. In other words, we carried out successive localizations across racks using MLI along with the threshold rule for identifying the reference nodes. We ran the simulation 1 million times. With range estimate variance of 0.2, the resulting localization error in 16th rack was found to be 1.9286e-5, which is pretty close to the expected value.

5 Conclusions and Future Work

In this paper we proposed a wireless USB based solution for localization of all plugged-in servers in data center racks. In particular, we showed how we can systematically locate all other nodes starting with a few known nodes and the associated errors in doing so. We showed how the use of multiple measurements and careful selection of reference nodes can be used to keep the error probabilities low in this process. For estimating the location of individual servers, we considered both Hyperbolic Positioning (HBP) and Maximum Likelihood identification (ML) techniques. Our analysis indicates that (i) the ML method is far superior to HBP in terms of accuracy of localization (ii) the ML method, though computationally more intense than HBP, is executable with cheap processors (iii) even with ML method, the overall accuracy critically depends on good range estimates between a pair of UWB radios.

Future work on the topic would address the issue of reliably estimating bias in distance measurement that arises naturally due to non line of sight components. We also would like to obtain bounds on total localization time as a function of the data center size.

Acknowledgements. We would like to thank Anas Tom for his help with Matlab program for Union Bound.

References

1. Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of Wireless Indoor Positioning Techniques and Systems. IEEE Trans on Systems, Man & Cybernetics 37(6), 1067–1080 (2007)
2. Brignone, C., et al.: Real Time Asset Tracking in the Data Center. Distr. and Parallel Databases 21(2-3), 145–165 (2007)
3. Di Benedetto M.-G., Giancola, G.: Understanding Ultra Wide Band Radio Fundamentals, pp. 474–475. Prentice-Hall, Englewood cliffs (2004)
4. Patwari, N., et al.: Locating the nodes: cooperative localization in wireless sensor networks. IEEE Signal Process. Mag. 22(4), 54–69 (2005)
5. Udar, N., Kant, K., Viswanathan, R., Cheung, D.: Ultra Wideband channel Characterization and Ranging in Data Center. In: ICUWB 2007, September 2007, pp. 322–327 (2007)
6. Darne, C., Macnaughtan, M., Scott, C.: Positioning GSM telephones. IEEE Communications Magazine 36(4), 46–54, 59 (1998)
7. Chan, Y.-T., Hang, Y.C., Ching, H.P.-C.: Exact and approximate Maximum Likelihood Localization Algorithms. Vehicular Technology, IEEE Transactions on 55(1), 10–16 (2006)
8. Seguin, G.E.: A lower Bound on the Error Probability for the Signals in White Gaussian Noise. IEEE trans on information theory 4(7), 3168–3175 (1998)