Improving Estimation of the Nearest Neighbor in Mobile Wireless Sensor Networks

HyungJune Lee, HyukJoon Kwon
Stanford University, Stanford, CA 94305
Email: {abbado, hjkwon}@stanford.edu

I. Introduction

In wireless networks, every node needs to keep its neighbors' link connectivity originated from periodic beacon messages in order to maintain the best route and forward packets along the path. Especially, wireless sensor networks should be working under low power constraint, so retransmission should be restricted in WSN as far as it can [2]. Moreover, low power wireless link makes the connectivity too volatile since low power wireless radio is vulnerable due to temporal interference from higher power radio, e.g., 802.11, Bluetooth, and etc. [3], [4]. Additionally, mobility makes the link connectivity more unstable [5]. These intrinsic wireless link behaviors make the link estimation for best routes far more difficult. Thus, it would be of great significance to estimate the best neighbor by efficiently exploiting the previous link status, i.e., history of beacon time and power so as to forward packets successfully.

In our project, we exploit the *locally weighted linear regression* method to estimate signal power when a node assumes to receive a packet from neighbors at an arbitrary time and rank the proximity of neighbors according to the estimated power. More specifically, the node which has the highest power would be the closest neighbor.

We use NS-2 network simulator to model the mobility of nodes and obtain training data from their own learning tables. We compare the estimation method with naive deterministic schemes such that a node, which has the largest received power in the last beacon time, is selected as the 1st rank. We expect our learning algorithm will reduce the error rate to estimate the best candidate for forwarding compared to the naive methods.

II. System Model

The network model we consider consists of N nodes randomly distributed over two-dimensional region. Each node is assumed to move with its own will under the random waypoint (RWP) model. The actual location of each node, $x^{(i)}$ where $i = 1, \ldots, n$, depends on the speed which is randomly distributed over [0, *max-speed*] and a certain pause time in the RWP model. In order to design the system on this assumption, we consider two typical models used well in a wireless sensor network: two-ray ground propagation model and shadowing propagation model.

A. Two-Ray Ground Propagation Model

The transmitted signal from a sender is delivered to a receiver through multi-ray paths, depending on how many it is reflected, diffracted or scattered on the designated ray-paths. It is assumed that the single ground reflection is dominated among other multi-path components in the two-ray model, which is illustrated in Figure 1.

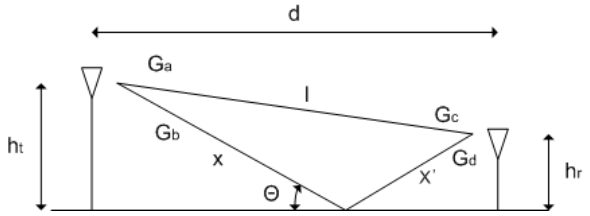

Fig. 1. Two-ray Ground ropagation Model

Then, the received power, P_r, is calculated via the geometric view as follows.

$$P_r = P_t\left(\frac{\lambda}{4\pi}\right)^2\left|\frac{\sqrt{G_l}}{l} + \frac{R\sqrt{G_r}e^{-j\triangle\phi}}{x+x'}\right|^2 \quad (1)$$

P_t is a transmitted power, $G_l = G_aG_b$ is the product of the transmit and receive antenna gains in the LOS direction, $G_r = G_cG_d$ is the product of the transmit and receive antenna gains corresponding to the reflective direction, and R is the ground reflection coefficient. $\triangle\phi = 2\pi(x + x' - l)/\lambda$ is the phase difference between a LOS path and the reflected path.

From the geometry, the distance difference and the phase difference is given by

$$x + x' - l = \sqrt{(h_t + h_r)^2 + d^2} - \sqrt{(h_t - h_r)^2 + d^2} \quad (2)$$

$$\triangle\phi = \frac{2\pi(x + x' - l)}{\lambda} \approx \frac{4\pi h_t h_r}{\lambda d} \quad (3)$$

where it is assumed that d is asymptotically large enough compared to $h_t + h_r$. If this assumption makes sense in the network model, then the parameters can be supposed like $x + x' \approx l \approx d$, $\theta \approx 0$, $G_l \approx G_r$, and $R \approx -1$. Therefore, the Eq. (1) is approximated into

$$P_r \approx P_t\left(\frac{\lambda\sqrt{G_l}}{4\pi d}\right)^2\left(\frac{4\pi h_t h_r}{\lambda d}\right)^2 \quad (4)$$

$$= P_t\left(\frac{\sqrt{G_l}h_t h_r}{d^2}\right)^2 \quad (5)$$

B. Shadowing Propagation Model

The two-ray model is assumed that there is no blockage on the ray-paths and only the ray distance difference is thought of as an important factor for a received power. However, a signal transmitted from a sender can pass through blockage, which gives rise to random variations of the receiver power at a receiver. The shadowing model is considered under the statistical random fluctuation. Typically, the log-normal distribution is used in a shadowing model which is superimposed with the path loss in a wireless sensor network. The receiver power is attenuated according to a distance between two nodes and randomly fluctuated for shadowing. Therefore, the receiver power is given by

$$\frac{P_r}{P_t}dB = 10\log_{10} K - 10\gamma \log_{10}\frac{d}{d_0} - \psi_{dB} \tag{6}$$

where ψ_{dB} is a Gauss-distributed random variable with mean zero and variance σ^2_{dB}. K is the path-loss coefficient and γ is the path-loss exponent.

The routing table on each node is built up based on the learning table of the received power, which is used for delivering the messages at any time. However, the information used to establish a learning table is obtained only at the regular beacon time. Therefore, in order to decide the nearest neighbor at any time, i.e. even at irregular beacon time, we need a new algorithm to decide who the nearest neighbor is.

III. Algorithms

A. Naive Methods

Each mobile node receives beacon packets from its neighbor mobile nodes, which include the beacon time, node ID, and the received power. Naive methods we use deterministically choose the nearest one at a given time based on the previous link status for neighbor nodes. These methods do not execute any further statistical processing.

The first naive method is to use the last received beacon time only as a main criterion to determine the proximity between a node and neighbors. For example, in this method, the neighbor from which a node received a beacon packet the most recently, would be the closest neighbor to the node.

However, this method do not provide a correct answer when a node received beacon packets from multiple neighbors, simultaneously. In this case, the received power of each beacon can be a useful indicator to determine the proximity between a node and neighbor nodes. The second naive method considers the last received power as well as the last received beacon time as criteria to determine the locality.

Both naive methods, however, have some weak points. As time goes by, the network also changes due to mobility of nodes. The previous information might be outdated to be used. For instance, even if a node received a beacon packet from a neighbor the most recently with the highest received power, another neighbor node could approach to the node rapidly in the mean time and be the closest node at the current time. Therefore, it would be required to estimate the received power at a given time by exploiting some statistical prediction methods so that each node can estimate the nearest neighbor with higher accuracy.

B. Locally Weighted Linear Regression

The learning algorithm we exploit is a locally weighted linear regression. In the first step, we try to approximate y, the received power, as a linear function of x, the beacon time with a regression variable, θ. Then, we define $h(x)$ is the approximated value of y given x.

$$h(x) = \theta_0 + \theta_1 x \tag{7}$$

The criteria how the approximation is close to y is determined by the cost function, $J(\theta)$, where the errors are squared and summed.

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \tag{8}$$

However, the normal linear regression gives equal weight for all the training data for approximation. However, depending on the distance from each training data, varying the weighting factor gives the better performance for estimation. In a weighted linear regression, the cost function is changed into

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m} w^{(i)}(h_\theta(x^{(i)}) - y^{(i)})^2 \tag{9}$$

where the weighting factor $w^{(i)}$ is given by

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right) \tag{10}$$

The weighting effect depends on the bandwidth τ, which will be given a proper value in our project. When the cost function approaches to the minimal point, the regression variable, θ, is optimized for estimating non-training data. In order to obtain the minimum of the cost function, we take a derivation of $J(\theta)$ (or the gradient for the case of vector) and obtain a θ which leads to $J(\theta) = 0$. As a result, we find that regression variable θ is

$$\theta = (X^T W X)^{-1} X^T W y \tag{11}$$

where W is a matrix whose diagonal terms are $w^{(i)}$.

With the locally weighted linear regression algorithm, each node can estimate the received power from neighbor nodes at a current time. Therefore, each node can be aware of who is the nearest node at a given time based on this regression method.

IV. Evaluation

To evaluate the performance of estimator, we generated test data from NS-2, which is one of the most widely used network simulators in wireless networks. The topology we used is 120m x 120m grid and 20 mobile nodes are moving around the grid according to random waypoint mobility model. The maximum speed and pause time we set is 5m/s and 3 seconds, respectively. Each node communicates with each other via 802.15.4 wireless PHY/MAC layer and communication range is set to 30m. We evaluated estimators under two different propagation models, i.e., two-ray ground model and shadowing model.

Each node transmits a beacon message with every 5 seconds. The total simulation time is 1000 seconds. For the first 400 seconds, each node updates its learning table which cumulates neighbor ID, received beacon time, and received beacon power over time. During the remaining 600 seconds, each node estimates the best neighbor once per every 20 seconds. The specific time is randomly chosen from uniform random variable over [0, 20] seconds. This test procedure provides 30 different data sets per node, thereby leading to 600 data sets from total 20 nodes. Using NS-2 simulator, we obtain the real nearest neighbor for each data set at the every execution time to estimate. Estimator tries to predict the nearest neighborhood list which is sorted by proximity. We calculated the probability that the real nearest node at each evaluation time successfully matches to one of the best k neighbors of each estimation where k = 1, 2, and 3. We compared our proposed estimation method with the naive methods specified in Section III-A.

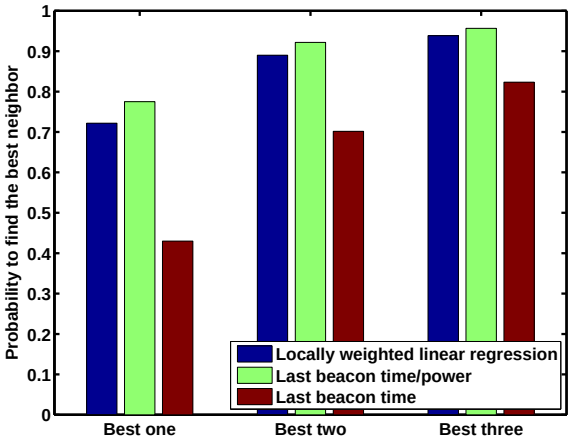

Fig. 2. Probability to find the best neighbor for locally weighted linear regression and naive methods in two-ray ground propagation model

Figure 2 shows that both our proposed scheme and the estimation scheme based on last received beacon time and power work pretty durably in two-ray ground model. Both two methods outperform the estimation scheme based only on beacon time. The proposed scheme successfully predicts the exact best neighbor with 74% and one of our best 3 neighbors from estimation successfully matches to the real best neighbor with 94%. This means that if a mobile node sends a packet to the best three next-hop nodes by multicast, the packet will be successfully delivered to the best neighbor with very high probability.

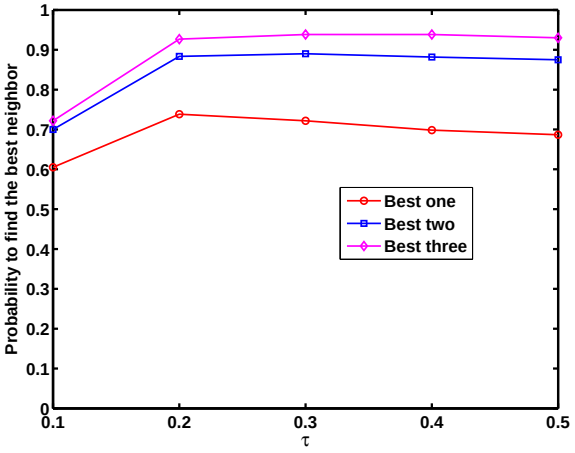

Fig. 3. Probability to find the best neighbor for locally weighted linear regression according to varying τ in two-ray ground propagation model

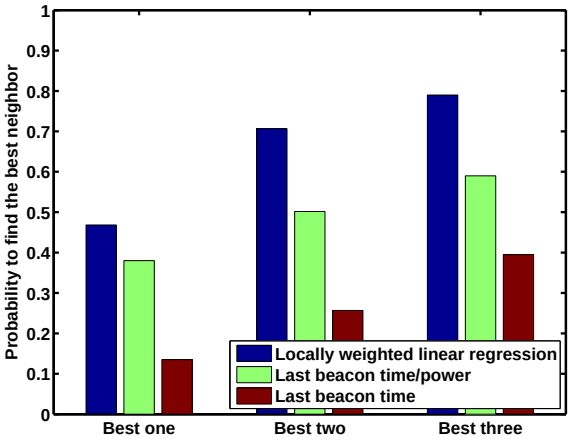

Fig. 4. Probability to find the best neighbor for locally weighted linear regression and naive methods in shadowing propagation model

Figure 3 shows that when τ is 0.2 or 0.3, the maximum prediction is achieved to find the best neighbor with locally weighted linear regression method. Beyond the value, the performance is all about the same or rather slightly decreased.

Figure 4 shows that the proposed scheme outperforms two naive methods with a factor of 1.4 in maximum in shadowing model. The proposed scheme successfully predicts the exact best neighbor with 47% and one of our best 3 neighbors from estimation successfully matches to the real best neighbor with 80%. The fidelity in shadowing model is less than in two-ray ground model since random behavior in shadowing model makes the prediction reliability a bit lower.

Figure 5 shows that τ= 0.6 or 0.7 achieves the maximum prediction level to find the best neighbor in locally weighted linear regression. The point we need to address is that the

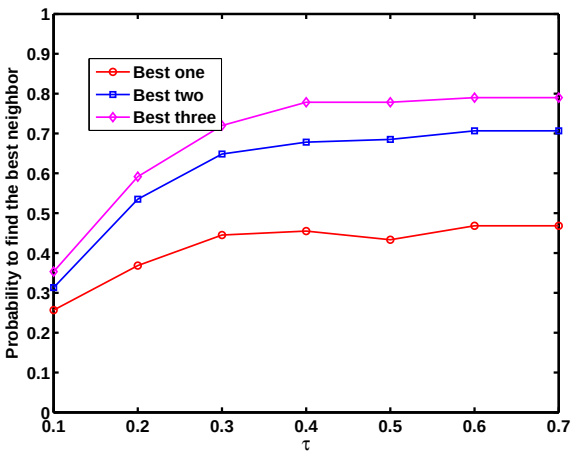

Fig. 5. Probability to find the best neighbor for locally weighted linear regression according to varying τ in shadowing propagation model

larger τ leads to the maximum estimation results in shadowing model compared to two-ray ground model because less penalization for the added random error is desirable.

V. Conclusion

Our proposed locally weighted linear regression method works durably for estimation of the best neighbor in mobile sensor networks even though random nature in shadowing model makes prediction fidelity lower than the deterministic two-ray ground model. The preliminary findings have strong implications to the design of ad-hoc routing protocols in mobile WSN. Routing protocols can choose the best three next-hop nodes and forward packets to them so as to improve reliable packet delivery in mobile networks. We leave the following questions as future works.

- How does the proposed estimator based on locally weighted linear regression work in NS-2 simulator: execution time (especially for pseudo-inverse operation), memory overhead, etc.?
- How can normal/adaptive Kalman filter be applied to the estimation of the nearest neighbor?
- Whether can other kinds of supervised machine learning algorithms be applied?

References

[1] Dongjin Son, Bhaskar Krishnamachari, and John Heidemann, "Experimental Analysis of Concurrent Packet Transmissions in Low-Power Wireless Networks," *ACM Sensys*, November 2006.

[2] Marco Zuniga and Bhaskar Krishnamachari, "An Analysis of Unreliability and Asymmetry in Low-Power Wireless Links," *ACM Transactions on Sensor Networks*, 2007.

[3] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis, "Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks," Technical Report SING-06-00. Stanford, 2006.

[4] HyungJune Lee, Alberto Cerpa, and Philip Levis, "Improving Wireless Simulation Through Noise Modeling," In *Proceedings of the Sixth International Conference on Information Processing in Sensor Networks (IPSN)*, 2007.

[5] Ranveer Chanadra, Christ Fetzer, and H. Karin, "Adaptive topology discovery in hybrid wireless networks," In *Proceedings of Informatics, 1st International Conference on Ad Hoc Networks and Wireless*, Toronto, vol. 16, September 20.22, 2002, pp. 1-16.

[6] Massimo Franceschetti, Lorna Booth, Matthew Cook, Ronald Meester, and Jehoshua Bruck, "Continuum Percolation with Unreliable and Spread out Connections," *Journal of Statistical Physics*, v. 118, N. 3-4, February 2005, pp. 721-734.

[7] Alec Woo, Terence Tong, and David E. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," *ACM SenSys*, 2003, pp.14-27.

[8] D. Kotz, C. Newport, and C. Elliott, "The Mistaken Axioms of Wireless-Network Research," Dartmouth College Computer Science Technical Report, TR2003-67, 2003.

[9] Thomas Moscibroda, "The Worst-Case Capacity of Wireless Sensor Networks," *ACM IPSN*, 2007, pp.1-10.

[10] Paolo Santi, "Topology Control in Wireless Ad Hoc and Sensor Networks," *ACM Comput*. Surv. 37, 2 (Jun. 2005), pp. 164-194.