International Journal of Scientific Research in Computer Science, Engineering and Information Technology
 | Volume 4 | Issue 3| ISSN: 2456-3307

iPath : Path Inference in Wireless Sensor Networks

P.Florance Rincy[1], Mrs. K.Sumalatha[2]

[1]M.Phil Research Scholar,Dept. Of Computer Science,Kamban College of Arts & Science for Women Tiruvannamalai, Tamil Nadu, India

[2]Head ofthe Department, Dept. Of Computer Science,Kamban College of Arts & Science for Women Tiruvannamalai, Tamil Nadu, India

ABSTRACT

Recent wireless sensor networks (WSNs) are becoming increasingly complex with the growing network scale and the dynamic nature of wireless communications. Many measurement and diagnostic approaches depend on per-packet routing paths for accurate and fine-grained analysis of the complex network behaviors. In this paper, we propose iPath, a novel path inference approach to reconstructing the per-packet routing paths in dynamic and large-scale networks. The basic idea of iPath is to exploit high path similarity to iteratively infer long paths from short ones. iPath starts with an initial known set of paths and performs path inference iteratively. iPath includes a novel design of a lightweight hash function for verification of the inferred paths. In order to further improve the inference capability as well as the execution efficiency, iPath includes a fast bootstrapping algorithmto reconstruct the initial set of paths. We also implement iPath and evaluate its performance using traces from large-scale WSN deployments as well as extensive simulations. Results show that iPath achieves much higher reconstruction ratios under different network settings compared to other state-of-the-art approaches.

Keywords: Measurement, Path Reconstruction, WirelessSensor Networks

I. INTRODUCTION

Wireless sensor networks (WSNs) can be applied in many application scenarios, e.g., structural protection [1], ecosystem management [2], and urban CO monitoring [3]. In a typical WSN, a number of self-organized sensor nodes report the sensing data periodically to a central sink via multihop wireless.Recent years have witnessed a rapid growth of sensor network scale. Some sensor networks include hundreds even thousandsof sensor nodes [2], [3]. These networks often employ dynamic routing protocols [4]–[6] to achieve fast adaptation to the dynamic wireless channel conditions. The growing network scale and the dynamic nature of wireless channel make WSNs become increasingly complex and hard to manage. Reconstructing the routing path of each received packet at the sink side is an effective way to understand the network's complex internal behaviors [7], [8].

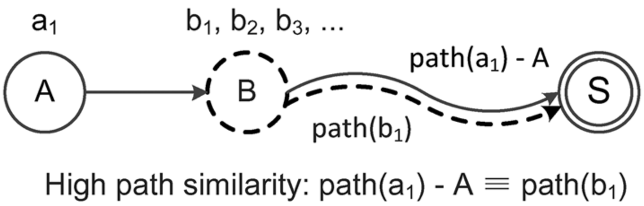


Fig.1. Example to illustrate the basic idea of iPath.

With the routing path of each packet, many measurement and diagnostic approaches [9]–[13] are able to conduct effective management and protocol optimizations for deployed WSNs consisting of a large number of unattended sensor nodes. For example, PAD [10] depends on the routing path information to build a Bayesian network for inferring the root causes of abnormal phenomena. Path information is also important for a network manager to effectively manage a sensor network. For example, given the per-packet path information, a network manager can easily find out the nodes with a lot of packets forwarded by them, i.e., network hop spots.

The contributions of this work are the following.

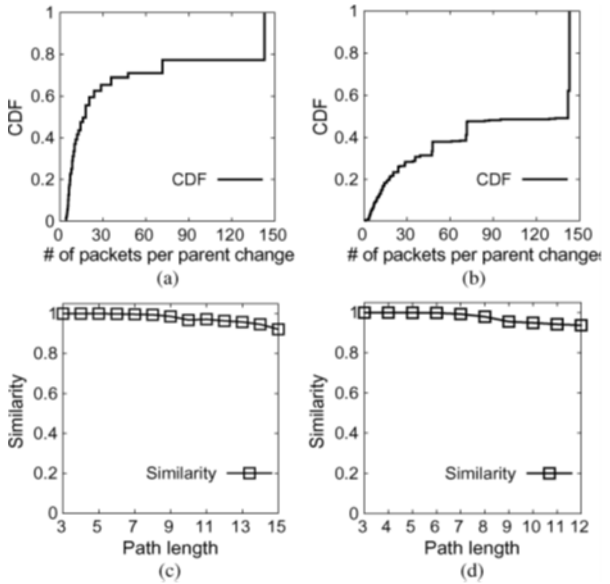
We observe high path similarity in a real-world sensor network. Based on this observation, we propose an iterative boosting algorithm for efficient path inference.We propose a lightweight hash function for efficient verification within iPath. We further propose a fast bootstrapping algorithm to improve the inference capability as well as its execution efficiency.

We propose an analytical model to calculate the successful reconstruction probability in various network conditions such as network scale, routing dynamics, packet losses, and node density.

II. MEASUREMENT STUDY

In order to quantify the path similarity in real-world deployment, we conduct a measurement study on two deployed networks—citysee[3] and greenorbs [2]. The cityseeproject is deployed in an urban area for measuring carbon emission. All nodes are organized in four subnets. Each subnet has one sink node, and sink nodes communicate to the base station through 802.11 wireless links. We collect traces from one sink of a subnet with 297 nodes. The greenorbs project includes 383 nodes in a forest area for measuring the carbon absorbance. These two networks use the Collection Tree Protocol [4] as its routing protocol. In order to reduce the energy consumption and prolong the network lifetime, all nodes except the sink node. Work at low-power listening states. The wakeup interval of the low power setting is 512 ms.Each node reports data packets to a sink with a period of 10 min. Each data packet carries the routing path information directly for offline analysis. We first look at the routing dynamics of the networks.

We measure a quantity that is defined to be the average number of periods (i.e., local packets) between two parent changes by a node. It is simply the inverse of the number of parent changes per period at a node. A smaller means more frequent parent changes. Fig. 2(a) and (b) shows the cumulative distribution function (CDF) of for all nodes in the two networks. We can see that these two network have different degrees of routing dynamics. On average, there is a parent change every 46.9 periods in cityseeand 89.1 periods in greenorbs. As a comparison, the MNT paper [8] reports a parent change every 88.2-793.3 periods of the networks tested, which have less frequent parent changes. We see that cityseeand greenorbs have high routing dynamics, making per-packet path inference necessary for reasoning about complex routing behaviors. On the other hand, we observe high path similarity in the networks, i.e., it is highly probable that a packet from node and one of the packets from's parent will follow the same path startingfrom's parent toward the sink. To quantitatively measure path similarity, wedefine sim (len) such that among all packets with path length len, there are sim(len)ratio of packets that follow the same path as at least one(len-1) hop packet. Fig. 2(c) and (d) shows the sim(len) values with varying len . We see that the values of sim (len) are close to 1, indicating that a high path similarity in both the cityseenetwork and greenorbs network. Note that the paths shown in these two figures include more than 99% of the total packet paths in these two traces. Therefore, the path similarity observation is not biased. The above results show that although there are severe routing dynamics, the path similarity can still be very high. This key observation gives us important implications for efficient path inference: If a similar short path is known, it can be used toreconstruct a long path efficiently.



III. NETWORK MODEL

In this section, we summarize the assumptions made and data fields in each packet. We assume a multihop WSN with a number of sensor nodes. Each node generates and forwards data packets to a single sink. In multisink scenarios, there exist multiple routing topologies.The path reconstruction can be accomplished separately based on the packets collected at each sink.In each packet,there are several data fields related to iPath.

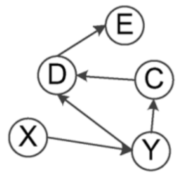
We summarize them as follows.

- The first two hops of the routing path, origin o(k) and parent p(k).Including the parent information in each packet is common best practice in many real applications for different purposes like network topology generation orpassive neighbor discovery [8], [22].
- The path length (k). It is included in the packet header in many protocols like CTP [4]. With the path length, iPath is able to filter out many irrelevant packets during the iterative boosting (Section V-A).
- A hash value h(k) of packet 's routing path. It can makethe sink be able to verify whether a short path and a long path are similar. The hash value is calculated on the nodes along the routing path by the PSP-Hashing (Section V-B).

IV. IPATH DESIGN

A. Iterative Boosting

iPath reconstructs unknown long paths from known short paths iteratively. By comparing the *recorded hash value* and the *calculated hash value*, the sink can verify whether a long path and a short path share the same path after the short path's original node. When the sink finds a match, the long path can be reconstructed by combining its original node and the shortpath.



path(c_1) = (C, D, E) (known)

path(y_1) = (Y, C, D, E) (unknown)
path(x_1) = (X, Y, C, D, E) (unknown)
path(x_2) = (X, Y, D, E) (unknown)

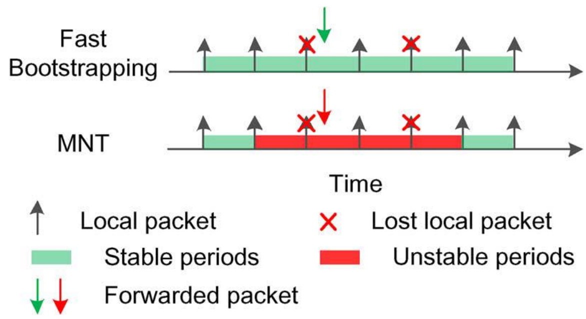
Case 1: hash(Y, path(c_1)) ≡ h(y_1) → path(y_1) = (Y, C, D, E)

Case 2: hash(X, Y, path(c_1)) ≡ h(x_1) → path(x_1) = (X, Y, C, D, E)

Case 3: hash(X, Y, path(c_1) - C) ≡ h(y_2) → path(y_2) = (X, Y, D, E)

B. Fast Bootstrapping

The iterative boosting algorithm needs an initial set of reconstructed paths. In addition to the one/two-hop paths, the fast bootstrapping



algorithm further provides more initial reconstructed paths for the iterative boosting algorithm. These initial reconstructed paths reduce the number of iterations needed and speed up the iterative boosting algorithm. The fast bootstrapping algorithm needs two additional data fields in each packet, parent change counter ***Pc (K)*** and global packet generation time. The parent change counter records the accumulated number of parent changes, and the global packet generation time can be estimated by attaching an accumulated delay in each packet [12]. For packet, there are an upper bound and a lower bound of the difference between the estimated packet generation time and the real value . The basic idea is to reconstruct a packet's path by the help of the local packets at each hop.

V. ANALYSIS

In order to quantify the reconstruction performance of ipathand two related approaches, we analyze these approaches by a novel analytical model. Here, the performance means the probability of a successful reconstruction, which is the most important metric. We use the following definitions for analysis.

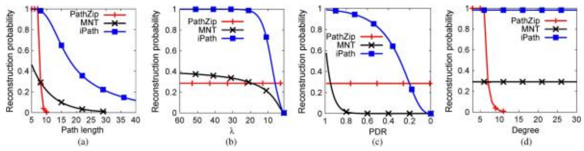
- Local packet generation period. Ipathdoes not require all nodes have the same local packet generation period. In order to simplify the presentation, we assume all nodes have the same packet generation period in this analysis section.
- Routing dynamics, which is the number of parent changes in a single period. On average, there is one parent change every local packets. We call these consecutive periods as one *cycle* for analysis.
- Packet delivery ratio PDR of packet. It can be calculated as the product of the packet reception ratios (PRR) along the routing path of packet.The average node degree.

A. Performance of ipath

The fast bootstrapping algorithm reconstructs an initial set of paths for the iterative boosting algorithm. Therefore, we first analyze the performance of the fast bootstrapping algorithm.

Performance of Iterative Boosting:

The iterative boosting algorithm reconstructs long paths based on short paths. Specifically, a path with length can be reconstructed by another path with length or (three cases in Section V-A). That path can be reconstructed by another even shorter path .A path can be successfully reconstructed only when the path helping to reconstruct it can also be successfully reconstructed. In other words, a



B. Methodology

Ipathis implemented in tinyos2.1. In the trace-driven study, we use traces collected from the citysee[3] project and the greenorbs project [2]. As mentioned in Section III, cityseeis a large-scale deployed network in an urban area for monitoring the carbon emission. Greenorbs is a large-scale sensor network for forest monitoring. A customized Collection Tree Protocol [4] is used as the routing protocol in these two projects. The cityseeand greenorbs traces include the first 10 hops in each packet for further offline analysis. Therefore, in the trace-driven study, we can use the collected routing information to reproduce the local operations on each node for each approach. Take pathzipas an example, we calculate the hash value according to the path included in each received packet at the sink side. Then, we run path zip'salgorithm to reconstruct paths and compare them to the collected ones to calculate the error ratio.

VI. CONCLUSION

In this paper, we propose iPath, a novel path inference approach to reconstructing the routing path for each received packet. iPath exploits the path similarity and uses the iterative boosting algorithm to reconstruct the routing path effectively. Furthermore, the fast bootstrapping algorithm provides an initial set of paths for the iterative algorithm. We formally analyze the reconstruction performance of iPath as well astwo related approaches. The analysis results show that iPath achieves higher reconstruction ratio when the network setting varies. We also implement iPath and evaluate its performance by a trace-driven study and extensive simulations.

Compared to states of the art, iPath achieves much higher reconstruction ratio under different network settings.

VII. REFERENCES

[1] M. Ceriotti et al., "Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment," in Proc. IPSN, 2009, pp. 277–288.

[2] L. Mo et al., "Canopy closure estimates with GreenOrbs: Sustainable sensing in the forest," in Proc. SenSys, 2009, pp. 99–112.

[3] X. Mao et al., "CitySee: Urban CO2 monitoring with sensors," in Proc. IEEE INFOCOM, 2012, pp. 1611–1619.

[4] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in Proc. SenSys, 2009, pp. 1–14.

[5] S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A highthroughput path metric for multi-hop wireless routing," in Proc. MobiCom, 2003, pp. 134–146.

[6] Z. Li, M. Li, J. Wang, and Z. Cao, "Ubiquitous data collection for mobile users in wireless sensor networks," in Proc. IEEE INFOCOM, 2011, pp. 2246–2254.

[7] X. Lu, D. Dong, Y. Liu, X. Liao, and L. Shanshan, "PathZip: Packet path tracing in wireless sensor networks," in Proc. IEEE MASS, 2012, pp. 380–388.

[8] M. Keller, J. Beutel, and L. Thiele, "How was your journey? Uncovering routing dynamics in deployed sensor networks with multi-hop network tomography," in Proc. SenSys, 2012, pp. 15–28.

[9] Y. Yang, Y. Xu, X. Li, and C. Chen, "A loss inference algorithm for wireless sensor networks to improve data reliability of digital ecosystems.," IEEE Trans. Ind. Electron., vol. 58, no. 6, pp. 2126–2137, Jun. 2011.

[10] Y. Liu, K. Liu, and M. Li, "Passive diagnosis for wireless sensor networks," IEEE/ACM Trans. Netw., vol. 18, no. 4, pp. 1132–1144, Aug. 2010.