International Journal of Research
Available at
https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 03
March 2017

Routing Path Inference in Dynamic and Large scale Networks

M Sindhu[1] **, N Dharani** [2] **,R Dhanalakshmi** [3]
[1]B.Tech Student, Dept of CSE, R M K College of Engineering, Chennai, Tamilnadu., India
[2]B.Tech Student, Dept of CSE, R M K College of Engineering, Chennai, Tamilnadu., India
[3]Assistant Professor, Dept of CSE, R M K College of Engineering, Chennai, Tamilnadu., 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 algorithm to 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, wireless sensor networks.

INTRODUCTION

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 algorithm to 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.

SYSTEM ANALYSIS

Existing System

With the routing path of each packet, many measurement and diagnostic approaches are able to conduct effective management and protocol optimizations for deployed WSNs consisting of a large number of unattended sensor nodes. For example, PAD 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. Then, the manager can take actions to deal with that problem, such as deploying more nodes to that area and modifying the routing layer protocols.

Furthermore, per-packet path information is essential to monitor the fine-grained per-link metrics. For example, most existing delay and loss measurement approaches assume that the routing topology is given as a priori.

The time-varying routing topology can be effectively obtained by per-packet routing path, significantly improving the values of existing WSN delay and loss tomography approaches.

Proposed System:

In this paper, we propose iPath, a novel path inference approach to reconstruct routing paths at the sink side. Based on a real-world complex urban sensing network with all node generating local packets, we find a key observation: It is highly probable that a packet from node and one of the packets from ‘s parent will follow the same path starting from ‘s parent toward the sink. We refer to this observation as high path similarity.

The basic idea of iPath is to exploit high path similarity to iteratively infer long paths from short ones. iPath starts with a known set of paths (e.g., the one-hop paths are already known) and performs path inference iteratively. During each iteration, it tries to infer paths one hop longer until no paths can be inferred.

In order to ensure correct inference, iPath needs to verify whether a short path can be used for inferring a long path. For this purpose, iPath includes a novel design of a lightweight hash function. Each data packet attaches a hash value that is updated hop by hop. This recorded hash value is compared against the calculated hash value of an inferred path. If these two values match, the path is correctly inferred with a very high probability.

In order to further improve the inference capability as well as its execution efficiency, iPath includes a fast bootstrapping algorithm to reconstruct a known set of paths.

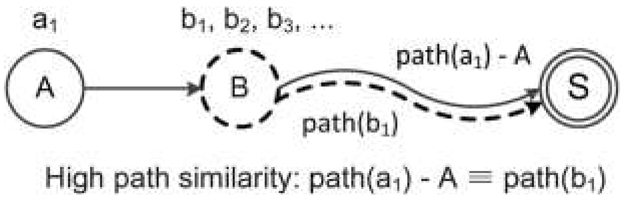


Fig.1 Example illustrating ipath

Related Work

In wired IP networks, fine-grained network measurement includes many aspects such as routing path reconstruction , packet delay estimation, and packet loss tomography. In these works,probes are used for measurement purpose [15]–[18]. Traceroute is a typical network diagnostic tool for displaying the path multiple probes. DTrack [18] is a probe-based path tracking system that predicts and tracks Internet path changes. According to the prediction of path changes, DTrack is able to track path changes effectively. FineComb [15] is a recent probe-based network delay and loss topography approach that focuses on resolvingpacket reordering. In fact, a recent work [19] summarizes the design space of probing algorithms for network performance measurement. Using probes, however, is usually not desirable in WSNs. The main reason is that the wireless dynamic is hard to be captured by a small number of probes, and frequent probing will introduce high energy consumption.A recent work [20] investigates the problem of identifying per-hop metrics from end-to-end path measurements, under the assumption that link metrics are additive and constant. Without using any active probe, it constructs a linear system by the end-to-end measurements from a number of internal monitors. Path information is assumed to exist as prior knowledge to build the linear system. Therefore, this work is orthogonal to iPath, and combining them may lead to new measurement techniques in WSNs.There are several recent path reconstruction approaches for WSNs [7], [8], [10], [21]. PAD is a diagnostic tool that includes a packet marking scheme to obtain the network topology. PAD [10] assumes a relatively static network and uses each packet to carry one hop of a path. When the network becomes dynamic,the frequently changing routing path cannot be accurately reconstructed.MNT [8] first obtains a set of reliable packets from the received packets at sink, then uses the reliable packet set to reconstruct each received packet's path. When the network is not very dynamic and the packet delivery ratio is high, MNT is able to achieve high reconstruction ratio with high reconstruction accuracy.However, as described in Section V-C, MNT is vulnerable to packet loss and wireless dynamics. PathZip [7] hashes the routing path into an 8-B hash value in each packet. Then,the sink performs an exhaustive search over the neighboring nodes for a match. The problem of PathZip is that the search space grows rapidly when the network scales up. Pathfinder [21] assumes that all nodes generate local packets and have a common interpacket interval (i.e., IPI). Pathfinder uses the temporal correlation between multiple packet paths and efficiently compresses the path information into each packet. Then, at the PC side, it can infer packet paths from the compressed information.Compared to PathZip, iPath exploits high path similarity between multiple packets for fast inference, resulting in much better scalability. Compared to MNT, iPath has much less stringent requirements on successful path inference: In each hop,iPath only requires at least one local packet following the same path, while MNT requires a set of consecutive packets with the same parent (called reliable packets). Compared to Pathfinder,iPath does not assume common IPI. iPath achieves higher reconstruction ratio/accuracy in various network conditions by exploiting path similarity among paths with different lengths.

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 and parent . Including the parent information in each packet is common best practice in many real applications for different purposes like network topology generation or passive neighbor discovery
- The path length . It is included in the packet header in many protocols. With the path length, iPath is able to filter out many irrelevant packets during the iterative boosting .
- A hash value of packet 's routing path. It can make the 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 .

- The global packet generation time and a parent change counter . These two fields are not required in iPath. However, with this information, iPath can use a fast bootstrapping algorithm to speed up the reconstruction process as well as reconstruct more paths.

SYSTEM DESIGN

The design of iPath includes three parts: iterative boosting, PSP-Hashing,and fast bootstrapping.

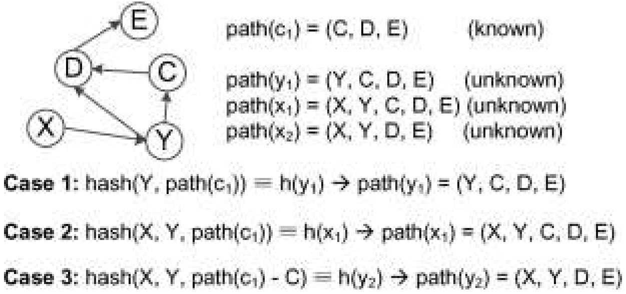


Fig 2. Example to illustrate three cases of reconstructing long paths based on short paths in the iterative boosting algorithm. X, Y, etc., are nodes, and xl, y1,etc., are packets originated from different nodes.

The iterative boosting algorithm is the main part of iPath. It uses the short paths to reconstruct long paths iteratively based on the path similarity. PSP-Hashing provides a path similarity preserving hash function that makes the iterative boosting algorithm be able to verify whether two paths are similar with high accuracy. When the global generation time and the parent change counter are included in each packet, a fast bootstrapping method is further used to speed up the iterative boosting algorithm as well as to reconstruct more paths.

PSP-HASHING

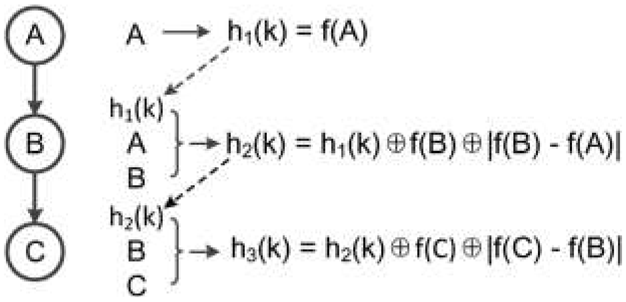


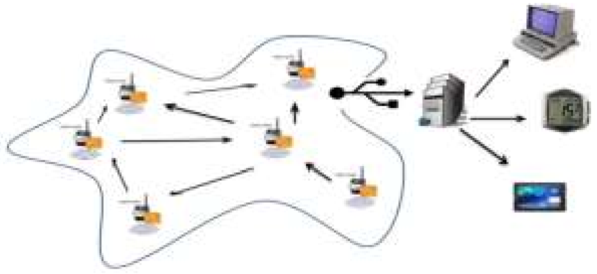
Fig 3 . PSP-Hashing function. Each node in the routing path takes three inputs and updates the hash value in packet . Note that we use to denote the hash value in packet at the th hop, instead of a function defined on packet itself.

As mentioned in the iterative boosting algorithm, the PSPHashing (i.e., path similarity preserving) plays a key role to make the sink be able to verify whether a short path is similar with another long path. There are three requirements of the hash function.

- The hash function should be lightweight and efficient enough since it needs to be run on resource-constrained sensor nodes.
- The hash function should be order-sensitive. That is, hash(A, B) and hash(B, A) should not be the same.
- The collision probability should be sufficiently low to increase the reconstruction accuracy.

FASTBOOTSTRAPPING

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 boostingalgorithm.



SYSTEM ARCHITECHTURE

IMPLEMENTATION

SOURCE MODULE

In this module, service provider browses the file; enter the file name and sends to the iPath router. Service provider encrypts the data and send to the router.

iPath ROUTER MODULE

In this module, router receives the file packets from the source, if packets size is greater than node BW then congestion occurs and then path inference will take place in order to find an alternative path. It takes another node and reaches the destination and load balancing takes place. When congestion occurs node band width can be increased.

RECEIVER MODULE

In this module, receiver receives the file. Calculates the time delay to reach the file from source to destination. Receiver stores the data details.

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

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] D. 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.
[11] W. Dong, Y. Liu, Y. He, T. Zhu, and C. Chen, “Measurement and analysis on the packet delivery performance in a large-scale sensor network,” IEEE/ACM Trans. Netw., 2013, to be published.
[12] J. Wang, W. Dong, Z. Cao, and Y. Liu, “On the delay performance analysis in a large-scale wireless sensor network,” in Proc. IEEE RTSS,2012, pp. 305–314.
[13] Y. Liang and R. Liu, “Routing topology inference for wireless sensor networks,” Comput. Commun. Rev., vol. 43, no. 2, pp. 21–28, 2013.
[14] Y. Gao et al., “Domo: Passive per-packet delay tomography in wireless ad-hoc networks,” in Proc. IEEE ICDCS, 2014, pp. 419–428.
[15] M. Lee, S. Goldberg, R. R. Kompella, and G. Varghese, “Fine-grained latency and loss measurements in the presence of reordering,” in Proc. ACM SIGMETRICS, 2011, pp. 329–340.
[16] Y. Shavitt and U. Weinsberg, “Quantifying the importance of vantage points distribution in internet topology measurements,” in Proc. IEEE INFOCOM, 2009, pp. 792–800.
[17] M. Latapy, C. Magnien, and F. Oudraogo, “A radar for the internet,” in Proc. IEEE ICDMW, 2008, pp. 901–908.
[18] I. Cunha, R. Teixeira, D. Veitch, and C. Diot, “Predicting and tracking internet path changes,” in Proc. SIGCOMM, 2011, pp. 122–133.
[19] A. D. Jaggard, S. Kopparty, V. Ramachandran, and R. N. Wright, “The design space of probing algorithms for network-performance measurement,” in Proc. SIGMETRICS, 2013, pp. 105–116.
[20] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, “Identifiability of link metrics based on end-to-end path measurements,” in Proc. IMC, 2013, pp. 391–404.
[21] Y. Gao et al., “Pathfinder: Robust path reconstruction in large scale sensor networks with lossy links,” in Proc. IEEE ICNP, 2013, pp. 1–10.
[22] A. Woo, T. Tong, and D. Culler, “Taming the underlying challenges of reliable multihop routing in sensor networks,” in Proc. SenSys, 2003, pp. 14–27.
[23] Y. Gao et al., “iPath: Path inference in wireless sensor networks,” Tech. Rep., 2014 [Online]. Available: http://www.emnets.org/pub/gaoyi/tech-ipath.pdf
[24] A. Liu and P. Ning, “TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks,” in Proc. IPSN, 2008, pp. 245–256.
[25] V. Handziski, A. Köpke, A. Willig, and A. Wolisz, “TWIST: A scalable and reconfigurable testbed for wireless indoor experiments with sensor networks,” in Proc. REALMAN, 2006, pp. 63–70.

[26] R. Lim, C. Walser, F. Ferrari, M. Zimmerling, and J. Beutel, “Distributed and synchronized measurements with FlockLab,” in Proc. SenSys, 2012, pp. 373–374.
[27] Z. Li, M. Li, and Y. Liu, “Towards energy-fairness in asynchronous duty-cycling sensor networks,” Trans. Sensor Netw., vol. 10, no. 3, pp.38:1–38:26, 2014.
[28] P. Levis, N. Lee, M. Welsh, and D. Culler, “TOSSIM: Accurate and scalable simulation of entire TinyOS applications,” in Proc. SenSys,2003, pp. 126–137.