



Reconstruction of Path using Compressive Sensing in Dynamic Wireless Sensor Network

Anand M

Assistant Professor, Department of Information Science and Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, India

Abstract: This paper presents CSPR, a compressive sensing based approach for path reconstruction in wireless sensor networks. By viewing the whole network as a path representation space, an arbitrary routing path can be represented by a path vector in the space. As path length is usually much smaller than the network size, such path vectors are sparse, i.e., the majority of elements are zeros. By encoding sparse path representation into packets, the path vector can be recovered from a small amount of packets using compressive sensing technique. CSPR formalizes the sparse path representation and enables accurate and efficient per-packet path reconstruction. CSPR is invulnerable to network dynamics and lossy links due to its distinct design. A set of optimization techniques are further proposed to improve the design. We evaluate CSPR in both testbed-based experiments and large scale trace-driven simulations. Evaluation results show that CSPR achieves high path recovery accuracy and outperforms the state-of-the-art approaches in various network settings.

Keywords: Packet path reconstruction; wireless sensor networks; compressive sensing; bloom filter.

I. INTRODUCTION

The per-packet routing path serves as the meta-information for understanding detailed Wireless Sensor Networks (WSNs) behaviours in many network maintenance and diagnosis situations, e.g., routing dynamics [33], detections on wormholes [9] or packet loss holes [32], end-to-end packet transmission delay [29] or even per-hop per-packet transmission delay [13], network diagnosis [18] [26], etc. Reconstructing per-packet routing path information, however, has been known non-trivial. WSNs are self-organized and usually deployed in dynamic environments. The underlying network topology constantly changes and no fixed routing path can be expected for each node [30]. A straightforward solution to reveal the packet path is to record the complete path during packet forwarding, e.g., storing the ID sequence of all relay nodes, in each packet. The introduced overhead linearly grows with the routing path length, far scalable.

The key insight of our design is as follows. The length of a routing path is usually much smaller than the network size. As a concrete example, the maximum path length reported in CitySee [22] is only 20 hops in comparison with its network size of 1200 nodes. Therefore, we can construct a path representation space, the number of whose dimensions equals to the total number of nodes in the network. In such a representation space, an arbitrary routing path can be represented by a path vector, where each element

corresponds to a node in the network. The path vector sets the hop numbers for nodes on the path and zeros for those not involved in the path. As the path length is much smaller than the network size, such path vectors are thus sparse, i.e., the majority of elements are zeros. The path reconstruction becomes a problem of unveiling all existing path vectors hidden in the representation space. If all nonzero elements of a path vector can be encoded (with few bytes) into the packets forwarded along the path, we can recover the path vector (and thus the represented routing path) based on a small amount of packets using compressive sensing technique [5] [12].

In this paper, we propose a Compressive Sensing based Path Reconstruction method, CSPR, which formalizes the sparse path representation and leverages compressive sensing to recover perpacket routing path. CSPR lets intermediate nodes briefly annotate the transmitted packets and classifies packets travelling along different paths into different groups. For a particular path, the forwarded packets encode independent observations and CSPR performs compressive sensing to recover the path when a certain amount of packets (and the annotations) are collected at the sink. The path reconstruction by CSPR requires no inter-packet correlations and utilizes only a small number of received packets. CSPR is thus invulnerable to topology dynamics and lossy links. On the protocol level, CSPR introduces only small and fixed overhead in annotating each packet, which could be optimized accordingly for practical WSNs (e.g., 8 bytes per packet for a network with 245 nodes). In addition to the basic design, we further propose a set of optimization techniques to gradually shrink the representation space and heuristically scan possible paths for all unrecovered path vectors through the network topology learnt from already reconstructed routing paths. The numbers of packets needed for remaining path reconstructions are lowered and processing is thus accelerated. To examine the performance of



CSPR, we first evaluate our method using a 29 TelosB mote tested. The experiment results validate the feasibility and applicability of CSPR in practice. We further conduct extensive and large-scale trace-driven simulations to examine the efficiency and scalability of CSPR. Compared to the state-of-the-art methods, CSPR achieves higher path recovery accuracy (i.e., 100% and 96% for experiments and simulations, respectively) with comparable overhead (8 extra bytes per packet).

II. PROBLEM STATEMENT

In a WSN, all sensor nodes generate and relay packets to the sink along some routing paths [28]. At the sink, a path reconstruction method is desired to recover the routing path each packet traveled. One packet path is an ID sequence from the source of the packet to the sink, including IDs of all intermediate nodes relaying this packet and their hop numbers as well. There have been many efforts made to address the path reconstruction problem (as reviewed in Section 5). Two state-of-the-art methods, MNT [16] and Pathfinder [14], have been recently proposed. MNT [16] reconstructs per-packet path by exploiting interpacket correlation, i.e., a relayed packet and its adjacent packets locally generated at any node i are usually forwarded to the same next hop. Such local packets serve as anchors of the relayed packet at node i . As the first-hop receiver is recorded in packets, the path of a packet can be obtained by concatenating the first-hop receivers of all its anchors. Improving on MNT, Pathfinder [14] tolerates certain inconsistency in inter-packet correlation via explicitly recording inconsistency in packets. The reconstruction failure occurs once the inconsistency exceeds the tolerance capacity. To accurately locate anchors, Pathfinder further imposes the packet generation rate of each node to be identical and fixed. Both MNT and Pathfinder require stable network topology such that inter-packet correlation can be captured. The practical WSNs, however, behave dynamically and the wireless links are far from stable [17]. Both network dynamic and packet loss have strong impacts on the anchor identifications, and thus deteriorate the performances of MNT and Pathfinder.

III. RELATED WORK

Pathfinder exploits temporal correlation between a set of packet paths and efficiently compresses the path information using path difference. At the PC side, Pathfinder infers packet paths from the compressed information and employs intelligent path speculation to reconstruct the packet paths with high reconstruction ratio. 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. Routing in wireless sensor networks differs from routing in fixed networks in various ways: there is no infrastructure, wireless links are unreliable, sensor nodes may fail, and routing protocols have to meet strict energy saving requirements. Many routing protocols have been proposed for WSN and power efficiency is a major issue in network.

IV. METHODS

- A. CSPR overview
- B. In-network path information encoding
- C. Compressive sensing based path reconstruction
- D. Optimization

A. CSPR Overview

Several fields in the packet header are used by CSPR to carry packet information. SEQ is the packet sequence number. sArr is source address of the packet. pLen records the path length. bFlt is a bloom filter to space-efficiently record the IDs and corresponding hop count information of all relay nodes. aMsr stores the encoded measurement along the path. All the five fields are initialized at the source node.

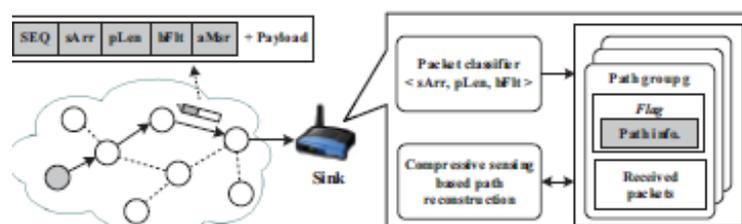


Figure 1.1: System architecture of CSPR.



In particular, SEQ and sArr keep unchanged after initialization, whereas pLen, bFlt and aMsr are updated at each intermediate hop. Note that only two fields, bFlt and aMsr, are additionally introduced by CSPR. SEQ, sArr and pLen can be usually found in the default packet header.

B. In-network path information encoding

In this subsection, we introduce the in-network updating of the last three fields pLen, bFlt and aMsr in turn.

- Updating of pLen.

The pLen field of each packet is initialized to 0 by the source and increased by one at each intermediate hop along the path. At each intermediate hop, pLen is updated prior to both bFlt and aMsr as the updating of latter two fields relies on the new pLen value.

- Updating of bFlt.

Bloom filter is an L-bit array associated with H independent hash functions, where L and H are two parameters to be determined. The bFlt field of each packet accommodates an L-bit array, and sensor nodes use the same set of H independent hash functions $f_i(\cdot)$, $i=1,2,\dots,H$, to update bFlt.

- Updating of aMsr.

The aMsr field in each packet is also initialized as 0 by the source and updated along the path. At each intermediate hop, the node encodes its hop number along the path in aMsr.

C. Compressive sensing based path reconstruction.

We first present the packet classification mechanism in CSPR, and then detail the compressive sensing based path reconstruction with path verification scheme to ensure the reconstruction

- Packet classification.

For each received packet, CSPR extracts

the 3-tuple key, $\langle sArr, pLen, bFlt \rangle$, from the packet header and then classifies it into a path group. A path group is designed to contain packets traveling the same path. At the sink, CSPR manages all path groups with a database.

- Path reconstruction.

Based on the encoded measurements in received packets, CSPR recovers the path for a path group using compressive sensing technique. CSPR executes path verification component to further ensure its correctness.

- Path verification.

Given a recovered path and a packet, the path verification components verify whether the recovered path is valid for the packet via path vector s of the recovered path and aMsr value of the packet. The recovered paths will benefit all future packets traveling on them. When a group with recovered path receives a packet, CSPR just simply invokes the path verification component to check whether the packet truly traveled the recovered path. If yes, this packet obtains its path immediately. In CSPR, for a given 3-tuple key, path groups are matched with the packet following the ascending order of gIdx keys. At worst, the packet might be assigned a newly formed path group.

D. Optimization

We propose two optimization techniques to improve the performance of our basic design.

- Reduction of path representation space.

This optimization aims to reduce the number of elements in a representation space by continuously monitoring the network topology. The minimum number of packets required to recover a length of k path is $ck \log(Nk)$, which is proportional to the space size N . The basic design utilizes all N nodes in the network to form the space. This component tries to reduce the space size for each path group and thus reduces its needed packets for path reconstruction.

- Heuristic path scanning.

For those path groups with insufficient accumulated packets even after a long time, this component is designed to scan possible paths for them based on the learnt network topology. It is triggered when the path reconstruction deadline of a path group is approaching. As this scheme is relatively computation intensive, it recovers path not only for the group which triggers its execution, but also for other unrecovered groups whose paths have the same source, sArr, as this group at the same time.



V. RESULTS AND DISCUSSIONS

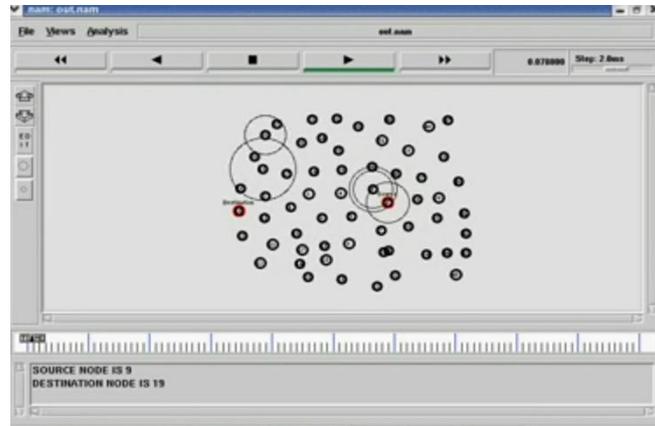


Figure2.1: Average number of paths for each node. The distribution of packet volumes for all path groups.

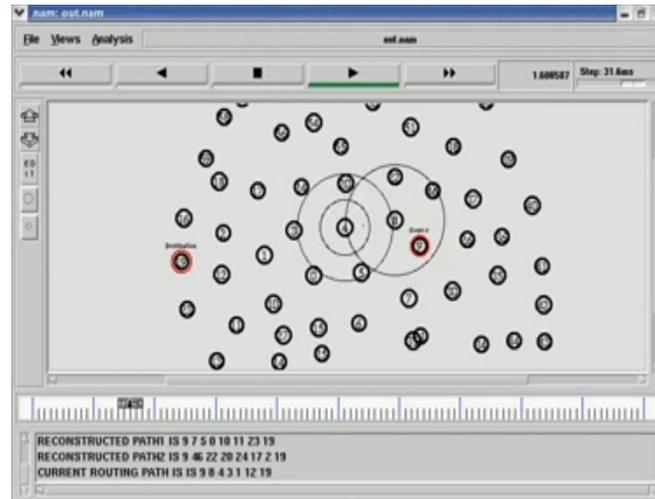


Figure2.2: The path information encoding process for a packet p. The SEQ, sArr, pLen, aMsr and bFlt fields are all initialized at source node, and the last three fields are updated hop-by-hop as depicted in the attached box at each intermediate hop. CSPR can extract a 3-tuple key from packet p at the sink.

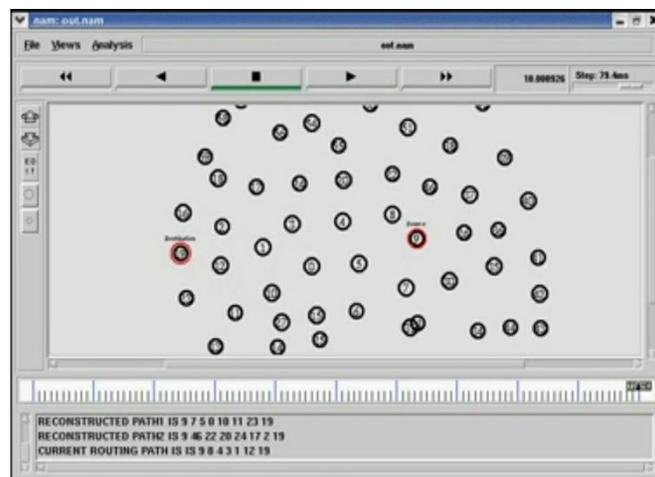


Figure2.3: The portion of packets benefited from already recovered routing paths. The packet path reconstruction progresses with the core.



CONCLUSION

In this paper, we present the CSPR, a compressive sensing based path reconstruction approach. Different from the state-of-the-art approaches, CSPR is inherently insensitive to network dynamics and lossy links. Extensive evaluations through both tested-based experiments and trace-driven simulations show that CSPR outperforms the state-of-the-art approaches in various network settings.

REFERENCES

- [1] TelosB mote datasheet. http://www.willow.co.uk/TelosB_Datasheet.pdf.
- [2] S. Alam and S. Fahmy. A practical approach for provenance transmission in wireless sensor networks. *Ad Hoc Networks*, 2013.
- [3] A. Broder and M. Mitzenmacher. Network applications of bloom filters: a survey. *Internet Mathematics*, 1(4):485–509, 2004.
- [4] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.
- [5] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.
- [6] Z. Chen and K. G. Shin. Post-deployment performance in wireless sensor networks. In *IEEE RTSS*, 2009.
- [7] M. Cheraghchi, A. Karbasi, S. Mohajer, and V. Saligrama. Graph-constrained group testing. *IEEE Transactions on Information Theory*, 58(1):248–262, 2012.
- [8] A. Coates, A. O. Hero III, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal Processing Magazine*, 19(3):47–65, 2002.
- [9] D. Dong, M. Li, Y. Liu, X.-Y. Li, and X. Liao. Topological detection on wormholes in wireless ad hoc and sensor networks. *IEEE/ACM Transactions on Networking*, 19(6):1787–1796, 2011.
- [10] W. Dong, C. Chen, J. Bu, X. Liu, and Y. Liu. D2: anomaly detection and diagnosis in networked embedded systems by program profiling and symptom mining. In *IEEE RTSS*, 2013.
- [11] W. Dong, Y. Liu, Y. He, and T. Zhu. Measurement and analysis on the packet delivery performance in a large scale sensor network. In *IEEE INFOCOM*, 2013.
- [12] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [13] Y. Gao, W. Dong, C. Chen, J. Bu, T. Chen, M. Xia, X. Liu, and X. Xu. Domo: passive per-packet delay tomography in wireless ad-hoc networks. In *IEEE ICDCS*, 2014.
- [14] Y. Gao, W. Dong, C. Chen, J. Bu, G. Guan, X. Zhang, and X. Liu. Pathfinder: robust path reconstruction in large scale sensor networks with lossy links. In *IEEE ICNP*, 2013.
- [15] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *ACM SenSys*, 2009.
- [16] M. Keller, J. Beutel, and L. Thiele. How was your journey?: uncovering routing dynamics in deployed sensor networks with multi-hop network tomography. In *ACM SenSys*, 2012.
- [17] X.-Y. Li, Y. Wang, H. Chen, X. Chu, Y. Wu, and Y. Qi. Reliable and energy-efficient routing for static wireless ad hoc networks with unreliable links. *IEEE Transactions on Parallel and Distributed Systems*, 20(10):1408–1421, 2009.
- [18] Y. Liu, K. Liu, and M. Li. Passive diagnosis for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 18(4):1132–1144, 2010.
- [19] X. Lu, D. Dong, X. Liao, and S. Li. PathZip: packet path tracing in wireless sensor networks. In *IEEE MASS*, 2012.
- [20] C. Luo, F. Wu, J. Sun, and C. W. Chen. Compressive data gathering for large-scale wireless sensor networks. In *ACM MobiCom*, 2009.
- [21] L. Ma, T. He, K. K. Leung, D. Towsley, and A. Swami. Efficient identification of additive link metrics via network tomography. In *IEEE ICDCS*, 2013.
- [22] X. Mao, X. Miao, Y. He, X. Li, and Y. Liu. CitySee: urban CO2 monitoring with sensors. In *IEEE INFOCOM*, 2012.
- [23] X. Miao, K. Liu, Y. He, Y. Liu, and D. Papadias. Agnostic diagnosis: discovering silent failures in wireless sensor networks. In *IEEE INFOCOM*, 2011.
- [24] D. Needell and J. A. Tropp. CoSaMP: iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.
- [25] H. X. Nguyen and P. Thiran. Using end-to-end data to infer lossy links in sensor networks. In *IEEE INFOCOM*, 2006.
- [26] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estrin. Sympathy for the sensor network debugger. In *ACM SenSys*, 2005.
- [27] D. Sy and L. Bao. CAPTRA: coordinated packet traceback. In *ACM/IEEE IPSN*, 2006.
- [28] S. Tang and J. Wu. Qute: quality-of-monitoring aware sensing and routing strategy in wireless sensor networks. In *ACM MobiHoc*, 2013.
- [29] J. Wang, W. Dong, Z. Cao, and Y. Liu. On the delay performance analysis in a large-scale wireless sensor network. In *IEEE RTSS*, 2012.
- [30] J. Wang, Y. Liu, Z. Li, W. Dong, and Y. He. QoF: towards comprehensive path quality measurement in wireless sensor networks. In *IEEE INFOCOM*, 2011.
- [31] W. Xu, E. Mallada, and A. Tang. Compressive sensing over graphs. In *IEEE INFOCOM*, 2011.
- [32] Y. Yang, Y. Xu, X. Li, and C. Chen. A loss inference algorithm for wireless sensor networks.