INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume3, Issue4)
Available online at www.ijariit.com

Effectively Reconstructing the Routing Paths in Sensor Networks

Mohammad Peer M. Shaikh
Sri Dharmasthala Manjunatheshwara College of Engineering and Technology, Karnataka
azharshaikh181@gmail.com

Prof. Anand D. Vaidya
Sri Dharmasthala Manjunatheshwara College of Engineering and Technology, Karnataka
Vaidya.anand@rediffmail.com

Abstract: ***In wireless sensor networks, sensor nodes are usually self-organized, delivering data to a central sink in a multi-hop manner. Reconstructing the per-packet routing path enables fine-grained diagnostic analysis and performance optimizations of the network. The performances of existing path reconstruction approaches, however, degrade rapidly in large scale networks with loss links. We present Pathfinder, a robust path reconstruction method against packet losses as well as routing dynamics. At the node side, Pathfinder exploits temporal correlation between a set of packet paths and efficiently compresses the path information using path difference. At the sink side, Pathfinder infers packet paths from the compressed information and employs intelligent path speculation to reconstruct the packet paths with high reconstruction ratio. We propose a novel analytical model to analyze the performance of Pathfinder. We further evaluate Pathfinder compared with two most related approaches using traces from a large scale deployment and extensive simulations. Results show that Pathfinder outperforms existing approaches, achieving both high reconstruction ratio and low transmission cost.***

Keywords: ***Pathfinder, Reconstruction, Path Recording, Path Speculation MNT.***

I. INTRODUCTION

The routing path of each packet can be very useful for understanding the internal network behaviours [8],[9], [10]. Sensor nodes are usually self-organized, delivering data to a central sink in a multi-hop manner. Reconstructing the per-packet routing path enables fine-grained diagnostic analysis and performance optimizations of the network. The difficulty in obtaining per-packet routing path is due to two reasons sensor networks are usually self organized and dynamically changing. There is usually no prior knowledge about the underlying routing topology. It is costly to directly attach path information in each packet since the overhead increases as the network scales up. Pathfinder, a robust path reconstruction method against packet losses as well as routing dynamics. At the node side, Pathfinder exploits temporal correlation between a set of packet paths and efficiently compresses the path information using path difference. Multi-hop data collection network with a single sink. The network employs a data collection protocol to collect data. Each sensor node generates and sends packets to the sink via multihop wireless. All nodes have a common packet generation period. Each data packet includes origin (original node where the packet is generated), parent (first hop receiver after a packet leaves its original node) and seq no (sequence number which increases each time a data packet is transmitted).At the sink side, Pathfinder infers packet paths from the compressed information and employs intelligent path speculation to reconstruct the packet paths with high reconstruction ratio. We propose a novel analytical model to analyze the performance of by recording the path difference at the node side, Pathfinder reconstructs the routing path of each packet effectively at the sink side by accurate reference packet locating. Analysis and evaluation show that Pathfinder can achieve high reconstruction ratios under difference network settings.

II. PROPOSED SYSTEM

The The paper proposes pathfinder, a novel path reconstruction approach: at the node side, Pathfinder exploits temporal correlation between a set of packet paths and efficiently compresses the path information using path difference; at the sink side, Pathfinder infers packet paths from the compressed information and employs intelligent path speculation to reconstruct the packet paths with high reconstruction ratio. We propose a novel analytical model to quantitatively analyze the reconstruction performance of Pathfinder.

We implement Pathfinder and compare its performance with two most related approaches using traces from a large scale real-world sensor network as well as extensive simulations. Results show that Pathfinder significantly outperforms MNT and Path- Zip in various network settings.

Advantages:

- High energy efficient.
- High performance.
- Maximum network period.
- High path reconstruction ratio.

III.DESIGN

In Pathfinder consists of two components the node side and the sink side shown in Fig 1. At the node side, the path recording component exploits temporal correlation among a set of packet paths and efficiently compresses the path information using path difference. At the sink side, the path reconstruction component infers packet paths from the compressed information and employs intelligent path speculation to reconstruct the packet paths with high reconstruction ratio.

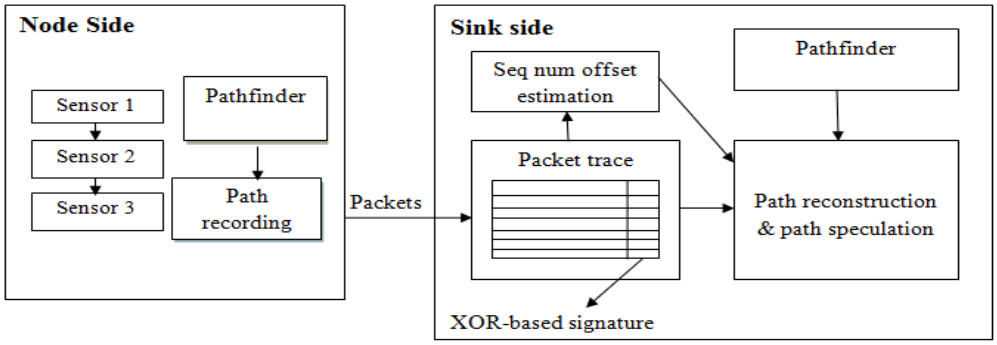

Fig 1 Architecture design

A. Packet Transmission

The entire The phase consist of source node which generates packets to the next hop in the format of (origin,format,seq no).The fig 2 depicts the flow of the packets

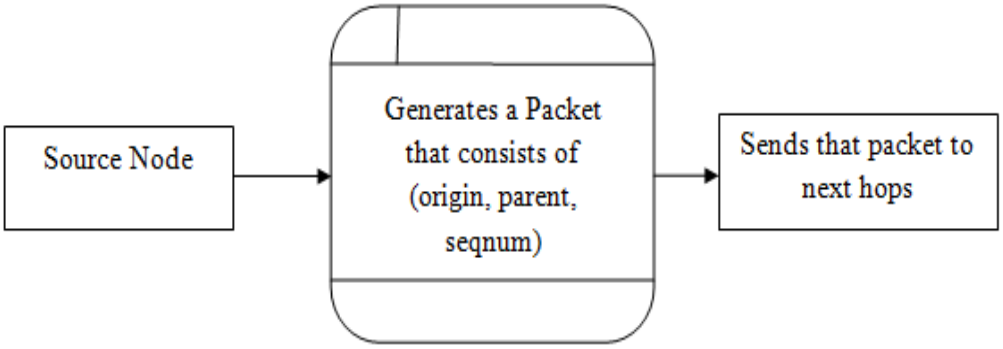

Fig 2 Packet Transmission Flow

B. Path Recording

At the node side, path difference of each packet is recorded in three data structures, bit vector, container, and XOR-byte, which are updated hop-by-hop. When a packet pktS originated from node S is delivered to a forwarding node fi, fi compares the next hop of pktS after fi with the parent of pktS's reference packet.If they are the same, fi appends a 0 to pktS's path bit vector. Otherwise, fi appends a 1 to the bit vector, and, at the same time, records the actual next hop after fi to pktS's path container if the size of the container does not exceed its limit shown in fig 3. In cases that the path difference is larger than the container's limit due to severe routing dynamics, the path difference cannot be recorded completely.

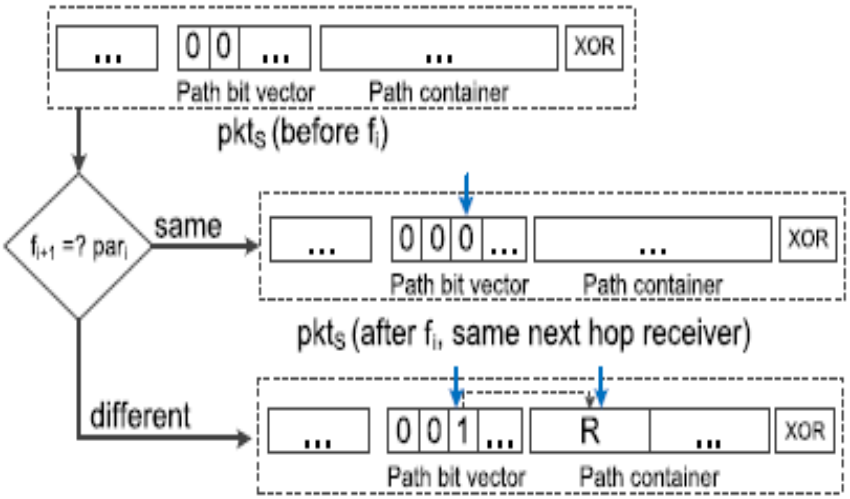

Fig 3 Updating Bit Vector

In order to record the path difference efficiently, Path- finder includes a path bit vector and a path container in each data packet. One bit in the path bit vector indicates whether the next hop of a forwarded packet at a particular forwarder is the same as the parent of its reference packet (i.e., Ii 0 in Equation (1)). If yes, no information needs to be recorded in the path container. Otherwise, Pathfinder records the actual next hop of the forwarded packet. Pathfinder additionally adds an XOR-byte for path verification. In order to reduce the message overhead, Pathfinder exposes a limit on the size of the path container, e.g 2 in our current implementation. Besides, Pathfinder employs Huffman encoding to compress the bit vector.

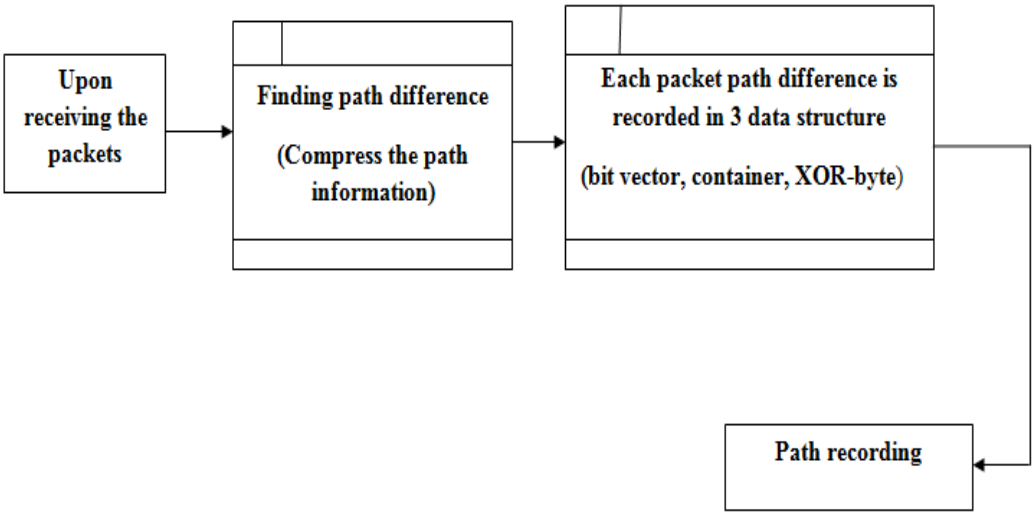

Fig 4 Path Recording

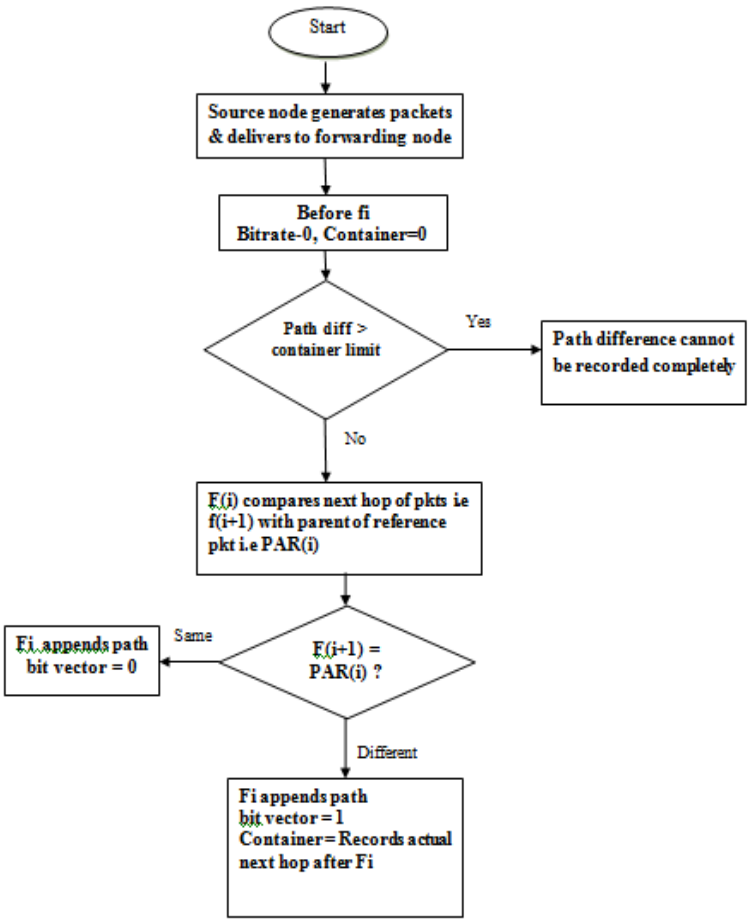

Fig 5 Flow of Path Recording

Hence, the bit vector can further be compressed. We employ Huffman encoding which an entropy is an encoding algorithm used for lossless data compression. The Huffman encoding algorithm encodes a stream of symbols according to their frequencies of occurrence. A symbol which occurs frequently will be assigned to a short code, hence the total size of the symbol stream can be greatly reduced on average. In our case, the input of the algorithm is an uncompressed bit vector. Since the bit vector is updated on each hop, each forwarder first decodes the bit vector, then updates it and encodes the updated bit vector again.

C. Path Reconstruction

For a received packet pktS originated from node S, the sink scans the uncompressed bit vector from the left to the right in each iteration. In the ith iteration, Pathfinder tries to reconstruct the ith hop forwarder. If Pathfinder finds a zero in the bit vector, it needs to locate pktS's reference packet shown in flow diagram fig 7.The parent of the reference packet is the ith hop forwarder of pktS. If Pathfinder finds a one in the path bit vector, it gets the ith hop forwarded from the path container if it is recorded.

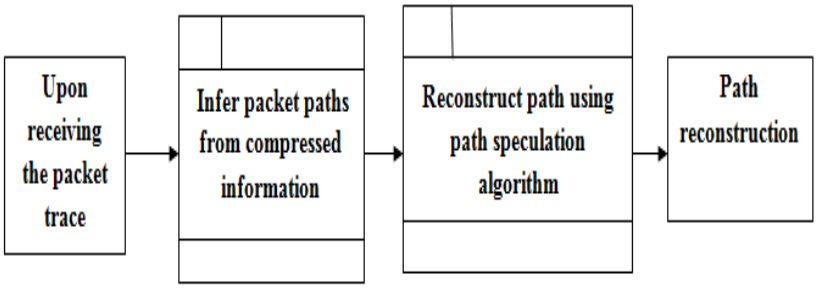

Fig 6 Upon Receiving Packets From Recorded Information

A key problem in Pathfinder is how to accurately localize reference packets for a given packet. If all reference packets are accurately identified, the reconstructed path is guaranteed to be correct. However, it is challenging due to several practical reasons.

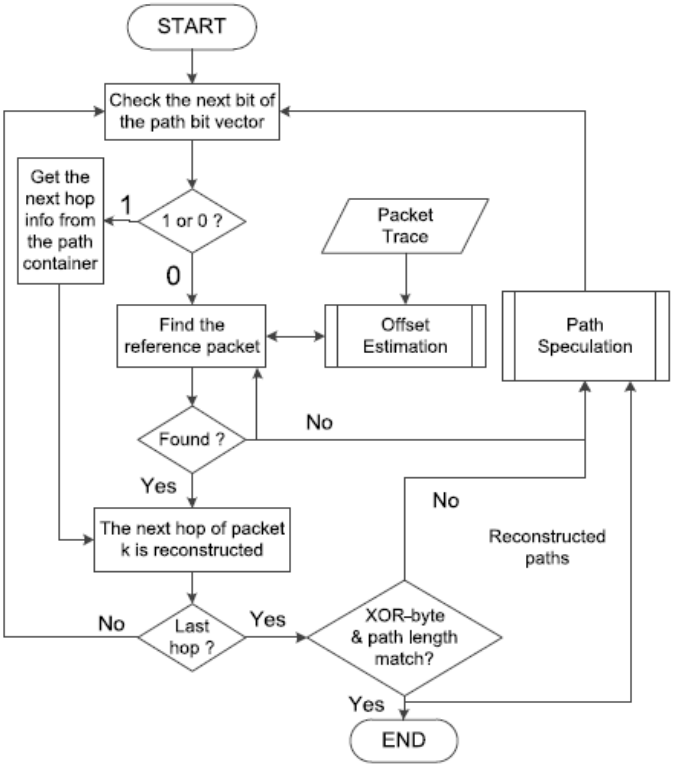

Fig 7 General Flow of Path Reconstruction

First, the reference packet may even be lost. In this case, Pathfinder will try to use a sub-reference packet to help reconstruct the path. The sub-reference packet is not the actual reference packet. It is the latest received packet at the sink from the same origin node as the actual reference packet. The sink uses this sub-reference packet to opportunistically reconstruct the path.

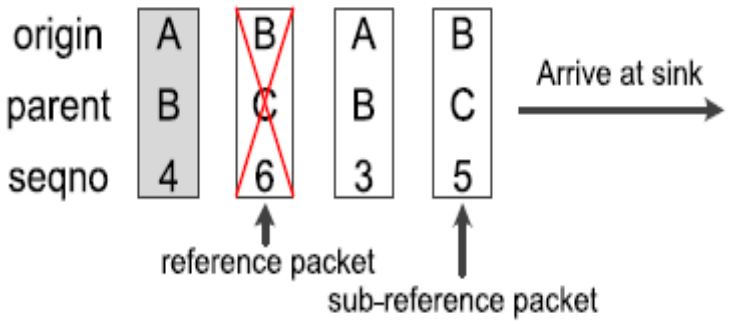

Fig 8 example of sub reference packet

We use a tuple (origin, parent, seqno) to represent a packet in this example shown in fig 8. The actual reference packet of the packet (A, B, 4) is (B, C, 6). If (B, C, 6) is lost, Pathfinder can use the sub reference packet (B, C, 5) opportunistically to infer the next hop of the packet (A, B, 4). If the sub-reference packet has the same parent as the reference packet, the reconstruction of the current hop is correct.Otherwise, the reconstruction of the current hop will be wrong. In that case, we tackle this problem by path speculation described in the next subsection.

Second, the arriving order of a forwarded packet and its reference packet may be different at the forwarder and the sink. Therefore, the latest local packet from the forwarder may not be the real reference packet at the sink side

D. Path Speculation

When path reconstruction fails (i.e., the XOR values do not match), Pathfinder employs path speculation to further enhance the reconstruction capability. There are several challenges to speculate the path. First, if the reconstructed path is not the actual one (since XOR values do not match), we do not get fine-grained knowledge which nodes match the actual ones. Therefore, we cannot enumerate neighbors of the matched nodes for efficient speculation. Second, enumerating all possible neighbors from the original node like PathZip causes a significant computation overhead, making it less scalable to large scale networks. The key insight of our approach is to exploit the already reconstructed paths. Due to temporal correlation, packets tend to follow links consisting of already seen forwarders towards the sink. Based on this insight, Pathfinder performs path speculation as follows.

This observation motivates us to devise Algorithm 1 for accurately estimating the real offset. Pathfinder first calculates an offset sequence at the sink. Each offset in the offset sequence is calculated by the sequence numbers of a packet and its reference packet located by the method in Path- finder-simple. We introduce a confidence value which is increased whenever two consecutive offsets are the same (line 5, 6). Then the estimated offset vi is the one with the maximum confidence value $c(v_{i)}$.

ALGORITHM 1: Sequence Number Offset Estimation

Input: A recent offset sequence
Output: vi: the estimated offset of the source and forwarder
1: procedure OFFSET-ESTIMATOR
2: Let $\omega = (\mathbf{s}_{1,\ldots.s_{n)}}$ be the offset sequence and forwarder
3: $c(v_{i)=}0$
4: for i = 2 to length(ω) do
5: if $s_i=s_{i-1}$ then
6: increase $c(v_{i)}$ where $v_{i=}s_i$
7: return v_i where $c(v_{i)}$ is the maximum

Within a time window, we use F(i) to denote the set of nodes that have forwarded packets for node i. After we have the F(i) for each node, we try to add directed edges in each F(i) according to the received packets. For two nodes a and b in F(i) we add a directed edge from a to b if there exists at least one packet from a to b. We then have a directed graph G(i) for each node. Finally, we try to enumerate all possible paths in G(i).

ALGORITHM 2: PATH SPECULATION

Input: P: received packets set; R: all reconstructed paths;
k: the packet whose path is being reconstructed
Output: x: packet k's reconstructed path
1: procedure PATH-SPECULATION
2: for each reconstructed path ri with origin node i do
3: if f ∈ ri and f ∉ F(i) then
4: insert f to F(i)
5: G(i) = CONSTRUCT-G(P, R, F(i))
6: for each path x from i to sink in G(i) do
7: if length, XOR-byte, bit vector or
8: container of x does not match k then
9: continue
10: if maximum time limit exceeds then
11: break
12: return x
13: procedure CONSTRUCT-G (P, R, F(i))
14: let G(i) be a graph
15: insert each node in F(i) to G(i)
16: for p in P do
17: if source(p) ∈ F(i) and parent(p) ∈ F(i)) then
18: add an edge from source(p) to parent(p)
19: for r in R do
20: for each two consecutive hops u, v in r do
21: if u ∈ F(i)and v ∈ F(i) then
22: add an edge from u to v
23: return G(i)

Algorithm 2 describes the above process. First, F(i) is constructed according to the reconstructed paths (lines 2, 3, 4). Then the Construct-G subroutine constructs the directed graph G(i) (lines 13-23). Specifically, an edge is added to the graph whenever there exists one packet, local packet (lines 16-18) or forwarded packet (lines 19-22), which has been delivered on that edge. Finally, all possible paths for node i are enumerated (line 6) considering the constraints on path length, XOR-byte, bit vector and container (lines 7, 8, 9).The bit vector and contained in the packet can be used to ensure the correctness of the reconstructed path. For example, a bit vector {0010000} and a container {20} indicate the third hop (after the parent) of the packet is node 20.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of Pathfinder using traces from a large scale deployment and extensive simulations.All nodes are uniformly distributed in a square area. The transmission range is configured to generate different network settings without breaking network connectivity. By simulations in networks with different number of nodes and different packet loss rates, we can evaluate the impact of network scale and packet losses to the approaches

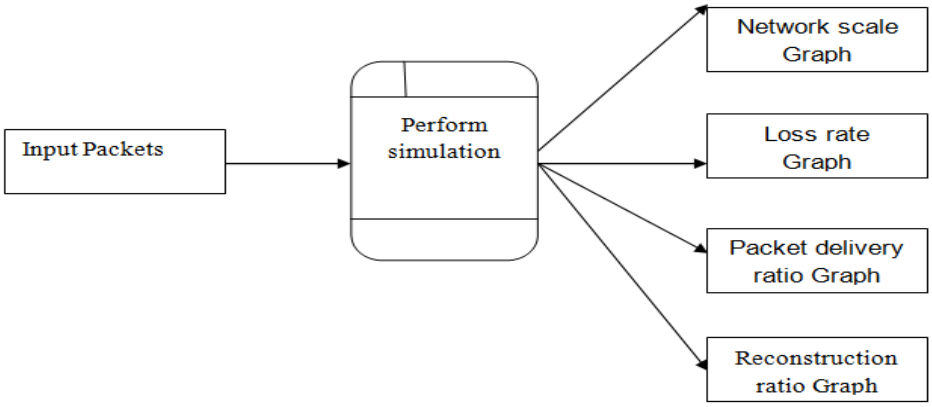

Fig 9 Performances Evaluation

A. *Effect on System Scale:*

All approaches perform well when the network size is 225 nodes. When the network scales up, MNT and PathZip's performance degrade and Pathfinder still has high reconstruction ratio. Note that Pathfinder-simple and Pathfinder-accurate have a small false ratio, which is the ratio of false reconstructed paths. The reason is that in the current implementation, the XOR protection field is only one byte. In scenarios which require extremely low false ratio (e.g., 0.01 percent), a two-byte XOR field will be sufficient.

For a 1-byte XOR field, we observe about 1 percent false reconstruction ratio. If more accurate path reconstruction is required, a longer XOR field can be used. We tune the length of the XOR field to evaluate its impact on the reconstruction accuracy

B. *Effect on Packet Loss:*

In order to evaluate the impact of packet loss, 10 to 40 percent of the packets are randomly removed from the original trace. We then evaluate the reconstruction ratios of the four methods in these traces. As shown in Fig. 10, the reconstruction ratio of Pathfinder is still very high when there are severe packet losses. The reconstruction ratio of MNT drops rapidly when packet loss increases. As described in Section 3.2, MNT cannot determine the next hop in the case of packet losses. PathZip has a stable performance when loss rate increases. The performance of Pathfinder-accurate performs significantly better than Path- finder-simple. The reason is that a simple approach to identifying reference packets cannot yield accurate results, especially when there are high routing dynamics

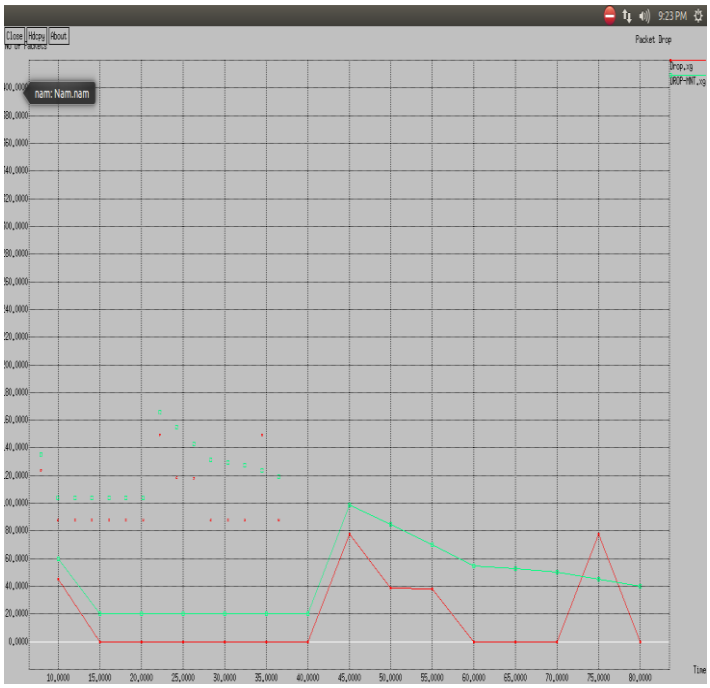

Fig 10 Packet loss rate

C. *Effect on Packet Delivery Ratio:*

Pathfinder accomplishes the most astonishing recreation proportion and a low wrong or false reproduction proportion for all the misfortune rates. Pathfinder-basic additionally accomplishes high recreation proportion, however the false reproduction proportion increments quickly when misfortune rate increments. Pathfinder basic's execution in the reproductions shown in fig 11.

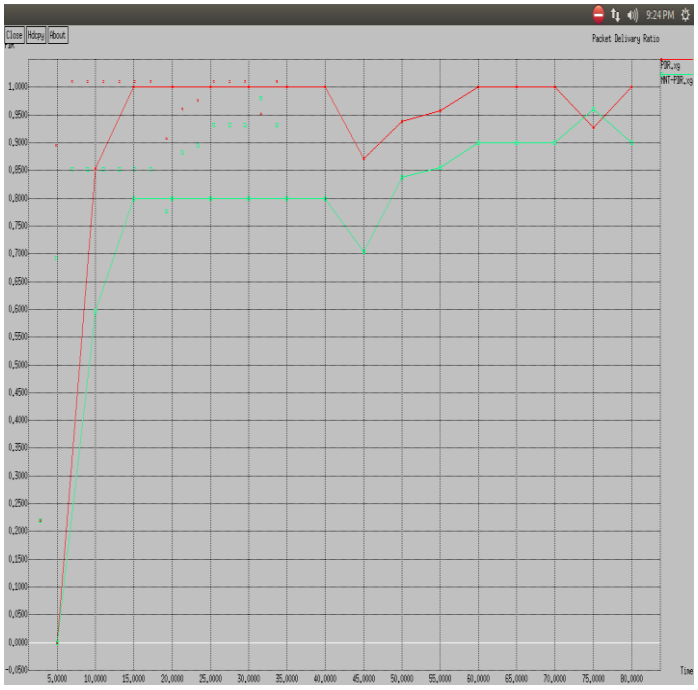

Fig 11 Packet Delivery Ratio

D. *Effect on Reconstruction Ratio:*

The ratio of reconstruction can be measured in terms of energy and throughput, As you can see in the graph the energy consumed by the model is very low compared to MNT shown in fig 12.

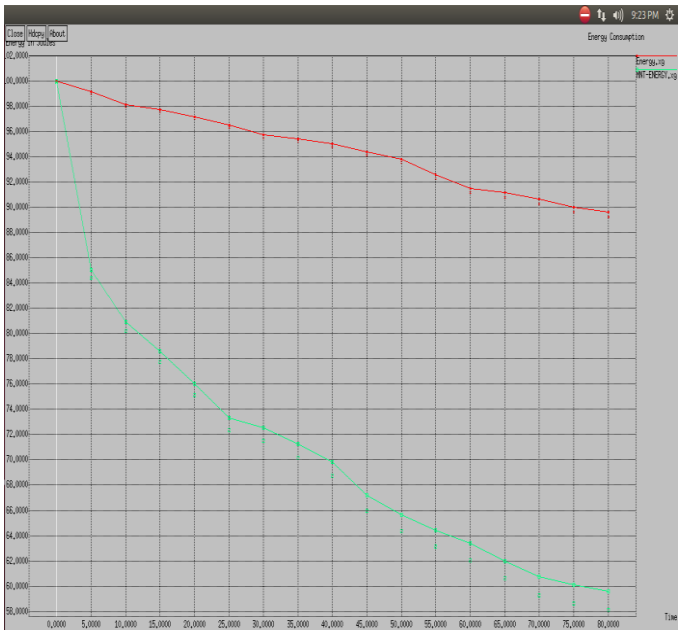

Fig 12 Energy consumption

The throughput will be high enough as we are using a Multihop networking where all the nodes participate at a same time and the packets are sent from the sink to the destination at the same time thus increased output compared to MNT in fig 13.

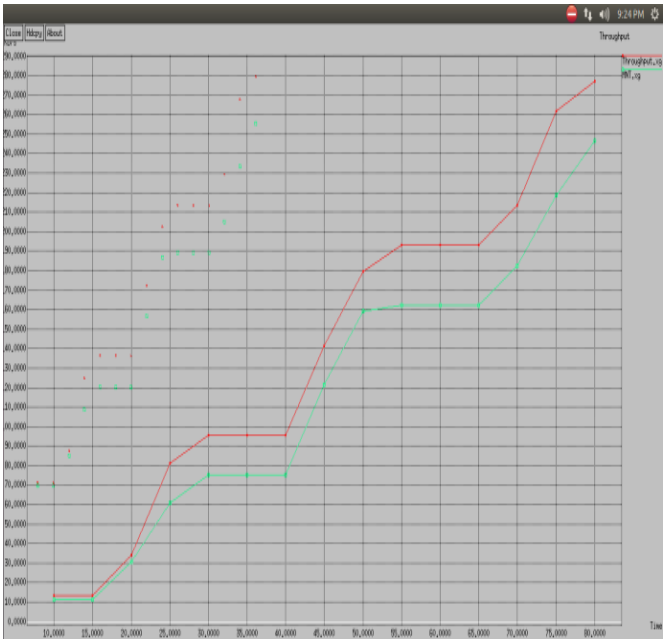

Fig 13 High Throughput

CONCLUSIONS

This paper presents Pathfinder, a robust path reconstruction method, for large scale wireless sensor networks with lossy links. By recording the path difference at the node side, Pathfinder reconstructs the routing path of each packet effectively at the sink side by accurate reference packet locating. Analysis and evaluation show that Pathfinder can achieve high reconstruction ratios under different network settings compared to MNT. There are multiple directions of future work of Path- finder. First, in some cases, we may be interested in the partial routing paths of packets dropped in the network due to node failures or poor link quality. Therefore, reconstructing the partial path of packets lost in the network is considered as future work. Second, in some cases (e.g., multiple node failures), the routing dynamic of the network will become very high in a short period of time. When the high routing dynamic causes the path difference value of some packets to be larger than 2, the path reconstruction of these packets may fail. Therefore, improving the design of Pathfinder to make it be able to handle temporary high routing dynamic is also considered as future work.

REFERENCES

[1] X. Mao, X. Miao, Y. He, T. Zhu, J. Wang, W. Dong, X. Li, and Y. Liu, "City See: Urbanmonitoring with sensors," in Proc. IEEE INFOCOM, 2012, pp. 1611–1619J.

[2] L.Mo, Y.He, Y. Liu, J. Zhao, S. Tang, X. Li, and G. Dai, "Canopy closure estimates with green orbs: Sustainable sensing in the forest," in Proc. 7thACMConf. Embedded Netw. Sensor Syst., 2009, pp. 99–112.

[3]] J. Zhou, Y. Chen, B. Leong, and B. Feng, "Practical virtual coordinates for large wireless sensor networks," in Proc. IEEE 18th Int. Conf. Netw. Protocols, 2010, pp. 41–51.

[4] L. Zhang, Q. Cheng, Y. Wang, and S. Zeadally, "A novel distributed sensor positioning system using the dual of target tracking," IEEE Trans. Comput., vol. 57, no. 2, pp. 246–260, Feb. 2008.

[5] P. Biswas and S. Phoha, "Self-organizing sensor networks for integrated target surveillance," IEEE Trans. Comput., vol. 55, no. 8, pp. 1033–1047, Aug. 2006.

[6] A. Ibrahim, Z. Han, and K. J. Liu, "Distributed energy-efficient cooperative routing in wireless networks," IEEE Trans. Wireless Commun., vol. 7, no. 10, pp. 3930–3941, Oct. 2008.

[7] A. E. A. A. Abdulla, H. Nishiyama, J. Yang, N. Ansari, and N. Kato, "HYMN: A novel hybrid multi-hop routing algorithm to improve the longevity of WSNs," IEEE Trans. Wireless Commun., vol. 11, no. 7, pp. 2531–2541, Jul. 2012.

[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. 10th ACM Conf. Embedded Netw. Sensor Syst., 2012, pp. 15–28.

[9] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in Proc. 1st Int. Conf. Embedded Netw. Sensor Syst., 2003, pp. 1–13.

[10] 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.

[11] K. Liu, M. Li, Y. Liu, M. Li, Z. Guo, and F. Hong, "Passive diagnosis for wireless sensor networks," in Proc. 6th ACM Conf. Embedded Netw. Sensor Syst., 2008, pp. 113–126.

[12] W. Dong, Y. Liu, Y. He, and T. Zhu, "Measurement and analysis of the packet delivery performance in a large scale sensor network," in Proc. IEEE INFOCOM, 2013, pp. 2679–2687.

[13] X. Lu, D. Dong, Y. Liu, X. Liao, and L. Shanshan, "PathZip: Packet path tracking in wireless sensor networks," in Proc. IEEE 9th Int. Conf. Mobile Adhoc Sensor Syst., 2012, pp. 380–388.

[14] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in Proc. 1st Int. Conf. Embedded Netw. Sensor Syst., 2003, pp. 126–137.

[15] Y. Gao, W. Dong, C. Chen, G. Guan, X. Zhang, and X. Liu, "Pathfinder: Robust path reconstruction in large scale sensor networks with lossy links," in Proc. IEEE Int. Conf. Netw. Protocols, 2013, pp. 1–10.

[16] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in Proc. 7th ACM Conf. Embedded Netw. Sensor Syst., 2009, pp. 1–14.

[17] M. Keller, L. Thiele, and J. Beutel, "Reconstruction of the correct temporal order of sensor network data," in Proc. 10th Int. Conf. Inf. Process. Sensor Netw., 2011, pp. 282–293.

[18] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in Proc. 4th Int. Symp. Inf. Process. Sensor Netw., 2005, pp. 364–369.

[19] O. Chipara, C. Lu, T. C. Bailey, and G. Catalin Roman, "Reliable clinical monitoring using wireless sensor networks: Experiences in a step-down hospital unit," in Proc. 8th ACM Conf. Embedded Netw. Sensor Syst., 2010, pp. 155–168.

[20] I. Talzi, A. Hasler, S. Gruber, and C. Tschudin, "PermaSense: Investigating permafrost with a WSN in the Swiss Alps," in Proc. 4th Workshop Embedded Netw. Sensors, 2007, pp. 8–12.

[21] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: Out-of-the-box environmental monitoring," in Proc. Int. Conf. Inf. Process. Sensor Netw., 2008, pp. 332–343.

[22] D. Moss, and P. Levis, "BOX-MACs: Exploiting physical and link layer boundaries in low-power networking," Stanford Inf. Netw. Group, Tech. Rep. SING-08-00, 2008.

