International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 4 | ISSN : 2456-3307

A Novel Per-Hop Per-Packet Delay Tomography in Wireless Ad Hoc Networks

S. Ulaganathan, G. Raja Raja Cholan
Department of Computer Science, Prist University, Vallam, Thanjavur, Tamil Nadu, India

ABSTRACT

Packet delivery delay is one of the most important performance metrics in multi-hop wireless ad-hoc networks. Though numerous research efforts have been spent on measuring and optimizing the end-to-end delay performance, there frequently lack precise and lightweight methods for decomposing the end-to-end delay into the per-hop delay for every packet. The per-hop per-packet delay is able to deeply improve the network visibility and create probable network management and measurement. This thesis proposes Domo, an accurate and lightweight delay tomography method for decomposing the packet end-to-end delay into every hop. The basic idea is to formulate the problem into a set of optimization problems by carefully considering the constraints among various timing quantities. At the network side, Domo attaches a small overhead to each packet for constructing constraints for the optimization problems. At the PC side, Domo employs semi definite relaxation and several other methods to efficiently solve the optimization problems. This thesis implements Domo and evaluates its performance extensively using large-scale simulations. Results show that Domo significantly outperforms two existing methods, nearly tripling the accuracy of the state-of-the-art.

Keywords: Wireless ad-hoc Network, Network Measurement, Delay Tomography.

I. INTRODUCTION

Overview of Wireless Ad Hoc Network:

The term ad hoc means “for this (only)” and is from Latin. The ad hoc network means wireless network without infrastructure, they can be called spontaneous network. One technique to identify with ad hoc networks is by comparing them with infrastructure based wireless networks, such as cellular network and WLAN. In the infrastructure based wireless networks a node can only transmit a packet to a destination node only via access point (in cellular network like GSM, it is called base station). The access point establishes a network region and only the nodes in this region can use access point’s services. There are a few unidentified events, which cause access point’s malfunction. The nodes lose their network and they are quasi not working. It is the largest infrastructure’s drawbacks.

There are also several explanations to sacrifice or not to use access point’s services. These can be cost factor, impossibility to install access point in short time, etc. In this case the nodes have to build its own network. This network is called wireless ad hoc network.

The wireless ad hoc networks only consist of nodes equipped with transceiver. The networks are created to be independent from an infrastructure. Therefore, the nodes must be able to arrange their own networks. Keep in mind that a node can now communicate only with other nodes in its transmission range. In the infrastructure based wireless network, the nodes can communicate with a node, which is situated in one more network region, by transmitting data to destination access point and this access point relay the data to the preferred node.

It seems like, that the ad hoc networks are not powerful enough. Every node has its individual transmission range, if these little transmission areas are combined;

they will outline a much superior transmission region. The nodes broadcast their data with single or multiple hopping techniques. Now an appropriate routing algorithm must be implemented, so the process of transmitting data will be more effective.

Problem Definitions:

In existing, the researchers only focus on End-to-End Delay. It is difficult to pinpoint the problematic nodes with only the end-to-end delay information. In contrast, per-hop delay information enables efficient detection of the problematic nodes. To tackle this problem, Domo, a passive, lightweight, and accurate delay tomography approach needed for decomposing the packet end-to-end delay into each hop.

Objectives:

In wireless ad-hoc networks, there are several existing works focusing on recording the end-to-end delay of each collected packet at the sink.

Some existing researchers propose to timestamp packets at the MAC layer to precisely record packet sending/receiving. Then the packet sojourn (means waiting) time at each hop can be accumulated and recorded in a particular field in the packet.

When the packet reaches the sink, that field will store the whole end to-end delay.

Different with these approaches, a novel lightweight and accurate delay tomography approach, Domo proposed for reconstructing per-hop per packet delay.

II. METHODS AND MATERIAL

1. Literature Review:

Literature survey is the most important step in software development process. Before developing the project it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps are to determine which operating system language, algorithms and techniques can be used for developing the project. Once the programmers start building the project the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

1.3 Wireless ad hoc networks and related topologies: applications and research challenges [2]

Wireless ad hoc networks consist of nodes that communicate over a common wireless channel, thus forming an all-wireless network. Contrary to cellular networks, the nodes are not supported by any type of additional infrastructure, such as base stations, a wired backbone, a central network controller, etc. Therefore, the establishment of the network and its operation must be exclusively over the wireless channel, and in a distributed and decentralized manner. Despite the obvious technical challenges, these networks have attracted significant research interest in recent years, as they are very well suited for many challenging settings, such as in search and rescue operations, sensing applications, mesh networks, vehicular communications, etc. In this overview article, S. Toumpis et al. [2] introduced the concept of wireless ad hoc networks, they discuss their advantages over the cellular topology, they review their history, they go over their most important current and future applications, and they discuss the ongoing research on the field, with an emphasis on the critical theoretical and technical challenges that must be overcome for this network paradigm to achieve its full potential.

1.4 Delay Measurement in Wired IP Network

Delay reconstruction is widely studied in wired IP networks for various network analysis [14], [16], [17]. LDA [16] is a compact data structure that efficiently computes the statistical delay of a link or a path, in a coordinated streaming environment. RLI [17] obtains per-flow delay by performing linear interpolation based on the data collected from the reference packets being injected to the sender. Besides the end-to-end delay measurement approaches, there are also approaches aiming at reconstructing per-hop delay. A recent work [14] proposes a very effective algorithm to reconstruct additive link metrics (e.g., delay) by path measurements. It assumes the link metrics are constant during the measurement, which is not the case in wireless ad-hoc networks. Another work [19] measures and analyzes the single-hop packet delay on an IP backbone network. It requires global clock

synchronization for all routers in the networks. These works cannot be directly adapted to wireless ad-hoc networks due to several reasons, such as the constrained resource and the lack of global synchronization.

1.4.1 Efficient Identification of Additive Link Metrics via Network Tomography [14]

L. Ma et al. [14] investigated the problem of identifying individual link metrics in a communication network from accumulated end-to-end metrics over selected measurement paths, under the assumption that link metrics are additive and constant during the measurement, and measurement paths cannot contain cycles. They know from linear algebra that all link metrics can be uniquely identified when the number of linearly independent measurement paths equals n, the number of links. It is, however, inefficient to collect measurements from all possible paths, whose number can grow exponentially in n, as the number of useful measurements (from linearly independent paths) is at most n. The aim of this paper is to develop efficient algorithms for constructing linearly independent measurement paths and calculating link metrics. They showed that whenever there exists a set of n linearly independent measurement paths, there must exist a set of three pair wise independent spanning trees. They exploited this property to develop an algorithm that can construct n linearly independent, cycle-free paths between monitors without examining all candidate paths, whose complexity is quadratic in n. A further benefit of the proposed algorithm is that the generated paths satisfy a nested structure that allows linear-time computation of link metrics without explicitly inverting the measurement matrix.

1.4.2 Every microsecond counts: tracking fine-grain latencies with a lossy difference Aggregator [16]

Many network applications have stringent end-to-end latency requirements, including VoIP and interactive video conferencing, automated trading, and high-performance computing where even microsecond variations may be intolerable. The resulting fine-grain measurement demands cannot be met effectively by existing technologies, such as SNMP, NetFlow, or active probing. R. R. Kompella et al. [16] proposed incrementing routers with a hash-based primitive that they call a Lossy Difference Aggregator (LDA) to measure latencies down to tens of microseconds and losses as infrequent as one in a million. Such measurement can be viewed abstractly as what they refer to as a coordinated streaming problem, which is fundamentally harder than standard streaming problems due to the need to coordinate values between nodes. They described a compact data structure that efficiently computes the average and standard deviation of latency and loss rate in a coordinated streaming environment. Our theoretical results translate to an efficient hardware implementation at 40 Gbps using less than 1% of a typical 65-nm 400-MHz networking ASIC.

1.4.3 Not all microseconds are equal: fine-grained per-flow measurements with reference latency interpolation [17]

New applications such as algorithmic trading and high-performance computing require extremely low latency (in microseconds). Network operators today lack sufficient fine-grain measurement tools to detect, localize and repair performance anomalies and delay spikes that cause application SLA violations. A recently proposed solution called LDA provides a scalable way to obtain latency, but only provides aggregate measurements. However, debugging application-specific problems requires per-flow measurements, since different flows may exhibit significantly different characteristics even when they are traversing the same link. To enable fine-grained per-flow measurements in routers, M. Lee et al. [17] proposed a new scalable architecture called reference latency interpolation (RLI) that is based on our observation that packets potentially belonging to different flows that are closely spaced to each other exhibit similar delay properties.

1.4.4 Measurement and analysis of single-hop delay on an IP backbone network [19]

K. Papagiannaki et al. [19] measured and analyzed the single-hop packet delay through operational routers in the Sprint Internet protocol (IP) backbone network. After presenting our delay measurements through a single router for OC-3 and OC-12 link speeds, they proposed a methodology to identify the factors contributing to single-hop delay. In addition to packet processing, transmission, and queuing delay at the output link, they observed the presence of very large delays that cannot be explained within the context of a

first-in first-out output queue model. They isolated and analyzed these outliers. Results indicate that there is very little queuing taking place in Sprint's backbone. As link speeds increase, transmission delay decreases and the dominant part of single-hop delay is packet processing time. They showed that if a packet is received and transmitted on the same line card, it experiences less than 20 minutes of delay. If the packet is transmitted across the switch fabric, its delay doubles in magnitude. They observed that processing due to IP options results in single-hop delays in the order of milliseconds. Milliseconds of delay may also be experienced by packets that do not carry IP options. They attribute those delays to router idiosyncratic behavior that affects less than 1% of the packets.

2. Delay Measurement in Wireless Ad-Hoc Network

2.1 End-to-End Delay:

In wireless ad-hoc networks, there are several recent works [11], [12] focusing on recording the end-to-end delay of each collected packet at the sink. Wang et al. [12] propose to timestamp packets at the MAC layer to precisely record packet sending/receiving. Then the packet sojourn time at each hop can be accumulated and recorded in a particular field in the packet. When the packet reaches the sink, that field will store the whole end-to-end delay. Different from these approaches, Domo aims at reconstructing more fine-grained delay, i,e., per-hop per-packet delay.

2.1.1 Reconstruction of the correct temporal order of sensor network data [11]

Collecting highly accurate scientific measurements asks for highest data quality and yield. But, satisfying these requirements is non-trivial, when considering phenomena common to wireless sensing systems such as clock drift, packet duplicates, packet loss and device reboots. Previous experience shows that these problems have not been resolved sufficiently by system design. In this paper, M. Keller et al. [11] introduce an offline approach to improve data quality by (a) providing a formal system model, (b) verifying conformance of packets received to this model, (c) providing the corrected packet sequence, and (d) providing additional information on packet generation inferred from temporally adjacent packets. They applied this method to a substantial amount of data from a real-world deployment and show the usefulness of this new intermediate packet processing step. In our validation of the proposed algorithm, they find that our approach successfully reconstructs the correct order of packet data streams. On application of the proposed data cleaning only a single violation is found when cross-validating a sequence of more than 4.6 million packets with ground truth derived from duplicate sensor data recovered from external storage post-deployment.

2.1.2 On the delay performance analysis in a large-scale wireless sensor network [12]

Wang et al. [12] presented a comprehensive delay performance measurement and analysis in a large-scale wireless sensor network. They built a lightweight delay measurement system and present a robust method to calculate the per-packet delay. They showed that the method can identify incorrect delays and recover them with a bounded error. Through analysis of delay and other system metrics, they saw to answer the following fundamental questions: What are the spatial and temporal characteristics of delay performance in a real network? What are the most important impacting factors, and is there any practical model to capture those factors? What are the implications to protocol designs? In this paper, they identified important factors from the data trace and show that the important factors are not necessarily the same with those in the Internet. Furthermore, they proposed a delay model to capture those factors.

2.2 Per-Hop Delay:

There are also several approaches [20], [21] related to fine-grained delay reconstruction in wireless ad-hoc networks. MNT [20] is the state-of-the-art approach to reconstruct per-hop packet arrival order in sensor networks. For each packet p, MNT is able to infer the two local packets right before and after packet p at each hop. Since the generation time of each local packet is known, the per-hop delay of the packet p can be bounded by the two corresponding local packets at each hop. Further, the bounds can be improved by correlating information from packets passing through the same forwarding nodes as packet p. The delay estimation accuracy of MNT mainly depends on the number of packets passing through each forwarding node per unit time, which further mainly depends on the inter-packet interval (IPI). Therefore, increasing

the time stamping granularity of MNT cannot improve the delay estimation accuracy. MessageTracing [21] records every packet sent and received into the local storage of each node. It does not reconstruct per-hop per-packet delay directly. However, by the information it records, it can infer the order of many packet sending/receiving events. From the perspective of revealing the internal behaviors of the networks, MessageTracing provides another possible method (i.e., local logging). In the evaluation section, it will show that Domo significantly outperforms these related approaches in terms of the reconstruction accuracy.

2.2.1 How was your journey?: uncovering routing dynamics in deployed sensor networks with multi-hop network tomography [20]

In the context of wireless data collection, a common application class in wireless sensor networks, this paper presents a novel, non-intrusive algorithm for the precise reconstruction of the packet path, the per-hop arrival order and the per-hop arrival times of individual packets from partial in-band information at runtime. Information is reconstructed outside the network immediately after a packet is received at the sink. After establishing the correctness of our proposed algorithm, M. Keller et al. [20] evaluated its performance in testbed experiments using CTP and Dozer, two well-known data collection protocols. Foremost interested in obtaining a better understanding of the performance of long-term real-world deployments, Multi-hop Network Tomography (MNT) is applied to in total more than 140 million packets that have been obtained from three multi-year WSN deployments of the PermaSense project.

2.2.2 Lightweight message tracing for debugging wireless sensor networks [21]

Wireless sensor networks (WSNs) deployments are subjected not infrequently to complex runtime failures that are difficult to diagnose. Alas, debugging techniques for traditional distributed systems are inapplicable because of extreme resource constraints in WSNs, and existing WSN-specific debugging solutions address either only specific types of failures focus on individual nodes, or exhibit high overheads hampering their scalability. Message tracing is a core issue underlying the efficient and effective debugging of WSNs. V. Sundaram et al. [21] proposed a message tracing solution which addresses key challenges in WSNs — besides stringent resource constraints, these include out-of-order message arrivals and message losses — while being streamlined for the common case of successful in-order message transmission. Our approach reduces energy overhead significantly (up to 95% and on average 59% smaller) compared to state-of-the-art message tracing approaches making use of Lamport clocks.

3. Network Model

3.1 Network Model:

The network model of Domo is explained in this chapter. It also summarizes the assumptions made and notations used in Domo. It assumes that Domo works in a wireless ad-hoc network running data collection task to a single sink node. Each node generates and sends data packets periodically to the sink. All nodes in the network can forward packets for other nodes. The network includes common phenomena in wireless ad-hoc networks such as routing dynamics, packet loss and no global timing information.

3.1.1. Modeling

FIFO Send Queue:

Each node includes an FIFO send queue for all outgoing packets. This queue keeps the packets generated by the node as well as packets to be forwarded. The node will keep sending the same packet at the queue head till a successful acknowledgement or the maximum number of retransmission is reached. FIFO send queue is widely used in routing protocols (e.g., CTP [22], MiniRoute) for wireless ad-hoc networks. Note that the FIFO send queue does not guarantee that the packet arrival order at sink is the same as the packet arrival order at each node. In fact, existing works [20] have shown that packets could be reordered in the network due to the dynamic routing in sensor networks.

Routing Path Information:

Since the wireless network topology is usually dynamic. Per-hop delays are only useful when the packet path is already known. Thanks to the recent progress in path reconstruction [20], [23], [24], it is able to reconstruct

the packet path with small overhead, even in large scale networks with hundreds of nodes. For example, a recent path reconstruction approach Pathfinder [23] achieves above 96% path reconstruction ratio in a network with 900 nodes. In case of imperfect path information, i.e., the routing paths of some received packets cannot be recovered, the per-hop delays of those packets are not defined. In fact, a packet without routing path information is viewed as a lost packet in Domo. As it will show in the evaluation, Domo is able to achieve satisfactory performance even with severe packet losses.

Node Delay:

Since wireless signal propagates very fast in the air, a packet delay from the source to the sink actually consists of the sojourn times the packet stays on the source node and all forwarding nodes. The sojourn time is the time between the node's radio starts to receive the packet and starts to transmit the packet. Since the time a node starts to transmit a packet is very close to the time the next hop starts to receive the packet. In the rest of this paper, it will use the difference between the time a node A starts to receive a packet and the time the next hop B starts to receive the packet to be the node delay of this packet at node A. The sojourn time at each node can be measured accurately on that node. In the implementation section, it will give a detailed description of measuring the node delay at that node.

End-to-End Delay:

Recent works [11], [12] show that the end-to-end delay can be obtained accurately after a packet reaches the sink. MAC layer timestamping technique is used to record the actual time a packet is transmitted by the radio at the physical layer. Based on this technique, the end-to-end delay can be recorded in the packet by accumulating all the node delays [12]. Note that this accumulating process is done on forwarding nodes of the packet. When the packet reaches the sink, the end-to-end delay is recorded in the packet while the node delays are not recorded.

4. Implementation Of Domo

4.1 Domo:

Domo, a Delay tomography approach for achieving lightweight and accurate per-hop per-packet delay reconstruction in wireless ad-hoc networks. Domo consists of two parts. At the node side, Domo attaches a small overhead to each packet for constructing constraints for the optimization problems. More specifically, each forwarder records the per hop delays of packets passing through it between its two local packets (originated from the forwarder), and attaches the sum of these per-hop delays to its latter local packet. At the PC side, Domo employs multiple techniques to efficiently solve the optimization problems.

First, semi-definite relaxation is used to facilitate problem solving. After relaxation, the optimization is turned into a convex problem and can thus be solved efficiently.

Second, a time window based method is used to further reduce the computation time while preserving the accuracy.

Third, a sub-graph extraction method is used to efficiently calculate the upper bound and lower bound of the per-hop per packet delay.

Finally, based on the calculated bounds of each per-hop per-packet delay, Domo further improves the accuracy of the estimated values of the unknown fine-grained delays.

4.2 Implementation:

In wireless ad-hoc networks, the sink node keeps receiving packets from other nodes. Each packet contains the routing information as well as the end-to-end delay between the source and sink. Then the sink would like to obtain the per-hop delay for every packet. This section formulates this per-hop per packet delay reconstruction problem into a set of optimization problems. It develops three convex constraints from network and packet transmission properties. With sufficient packets, the sink node is able to shrink the bounds of all per-hop delays for every packet by solving these optimization problems. Since per-hop delays and the arrival times can be transformed to each other easily.

For each packet p, the packet generation time $t_0(p)$ and the arrival time at the sink $t_{|p|-1}(p)$ are already known. The rest of its arrival times at other hops are the unknowns which need to be reconstructed. When the unknown arrival times of each packet are considered

separately, it is difficult to reconstruct them at the sink side. However, it observes that the arrival times of multiple packets are not independent, but highly correlated. By exploiting the correlation among per-hop arrival times of multiple packets, it can accurately reconstruct the arrival times.

Concretely, it constructs three kinds of constraints of the unknown arrival times to represent the correlation among them. When it needs to calculate the estimated values or the bounds of the arrival times, different optimization problems can be constructed and solved.

Constraints Construction

FIFO Constraint:

The first kind of constraint is the FIFO constraint. Since there is a FIFO send queue on each node, packets always leave a node in the same order as they arrive. For any two packets x, y with a common node n (not the sink) in their paths ($n \in path(x)$ and $n \in path(y)$), let $n = Nix(x) = Niy(y)$, then it have the following constraint.

$$(tix(x) - tiy(y))(tix+1(x) - tiy+1(y)) > 0.$$

Order Constraint:

The second kind of constraint is the order constraint. The arrival times on all hops of a packet should be in order. That is, $t0(p) < t1(p) < ... < t|p|-1(p)$. In practice, the per-hop delay has a minimum value due to the software processing delay. It use ω to denote this minimum software processing delay, and the second kind of constraint becomes the following. $t0(p) < t1(p) - \omega < \cdots < t|p|-1(p) - (|p| - 1)\omega$.

Sum-of-Delays Constraint:

The last kind of constraint is related to the sum of node delays S(p). S(p) can be calculated on node N0(p) accurately, but cannot be represented accurately by the arrival time t at the sink side. The reason is that the sink does not know which packets were forwarded by packet p's source node N0(p) between packet p and its previous packet q from the same source node. In order to relate the S(p) in each packet p to the arrival times, it first define a candidate set of packets C(p) as follows. For a particular packet p, it define its candidate set C(p) as a set of packets that satisfy the following three conditions.

1) For any packet $x \in C(p)$, N0(p) is in packet x's path;
2) For any packet $x \in C(p)$, its packet generation time t0(x) is earlier than packet p's generation time t0(p);
3) For any packet $x \in C(p)$, its arrival time at sink $t|x|-1(x)$ is later than packet q's generation time t0(q), where packet q is the previous local packet from the same source node as packet p.

4.2.2. Estimated Values of Arrival Times

There may be multiple solutions which satisfy all these constraints. This work needs an optimization goal to obtain a single solution as the estimated values of all unknown arrival times. It observes that within a relatively short period of time, the node delays of multiple packets on the same hop are similar. It measures the node delay in a network with 80 TelosB nodes. The "Per hop short term" line shows the standard deviation of the node delays at the same node within a short period of time, i.e., 60 seconds. The "Overall" line shows the standard deviation of the node delays at all node in the network. It can see that the node delays at the same node within a short period of time are much more stable than the node delays at all nodes. There are several parameters which have the largest relative importance to the delay, including the queue length, the MAC back-off time, link quality, etc. Within a short period of time, the queue lengths of packets at the same hop are usually similar. Further, the MAC back-off time of a node mainly depends on the number of its neighbors. Therefore, packets at the same hop within a short period of time usually have similar MAC back off times. In summary, several important factors that affect the node delay are similar at a certain node within a short period of time.

Nevertheless, with increasing number of packets arriving at the sink, the reconstruction process will also grow significantly. Thus, this work divides the original trace into multiple time windows to improve the efficiency. One problem left is that the packets near the time window boundaries do not have enough number of constraints to limit their possible values. This will lower the accuracy of the arrival times of these packets. Therefore, it uses the following improved time window approach instead. For packets in a time window, it obtains their estimated per-hop arrival times by solving

the optimization problem constructed by the packets in that time window. Due to the mentioned accuracy concern, it drops the estimated values near the time window boundaries and keeps the ones far from the boundaries. Then it considers the next time window which has an overlapped area with the previous time window. It applies the same strategy on the second time window and drop some estimated values near the boundaries. The time windows are chosen so that after dropping a number of estimated values, the remaining values can also cover all unknown arrival times. Figure 1 illustrates this improved time window approach.

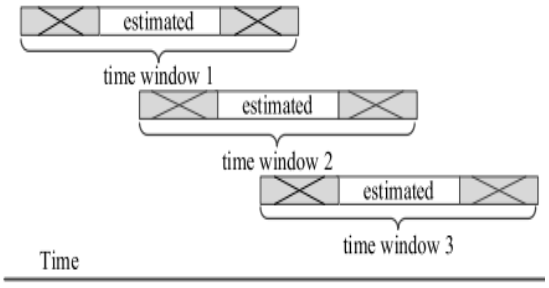

Figure 1: The improved time window approach to prevent the inaccuracy caused by the information loss near the boundaries. The arrival times in the grey areas are dropped.

The remaining arrival times in the white areas of multiple time windows cover all the unknown arrival times.

There is a key parameter in this improved time window approach, the ratio of the number of remaining values and the number of all values in one time window. It refers this ratio as effective time window ratio. A larger effective time window ratio means more values are kept in each time window and the accuracy is lower. A small effective time window ratio will increase the number of time windows and prolong the computation time. In practice, it set this ratio to be 0.5 which achieves both satisfactory accuracy and efficiency. In the evaluation section, it tunes this ratio to validate its impact on the accuracy of Domo.

4.2.3. Bounds of Arrival Times

In some scenarios, the upper bounds and lower bounds of the arrival times are more useful than the estimated values. In this case, this work uses a different method to calculate the bounds of all unknown arrival times. The basic idea is to find the upper bound and lower bound for each arrival time which satisfies the three kinds of constraints. For each unknown arrival time t, it constructs two optimization problems to obtain its lower bound and upper bound, Minimize t and Maximize t, considering the same constraints. The problem is that it needs to solve twice number of optimization problems as the number of unknown arrival times. Although the problem is a convex optimization problem, it still needs to improve its efficiency. In fact, it is not necessary to take all unknown arrival times and related constraints into account when it only want to optimize one arrival time. Intuitively, the arrival times near the one being optimized should be sufficient. Therefore, how to select the arrival times and the related constraints for the optimization problems becomes important. it use a graph model to select the proper arrival times and the related constraints. Consider each unknown arrival time as a vertex. An edge between two vertices indicates that there is at least one constraint involving these two arrival times. Then it extracts a sub-graph for each arrival time for optimization. Figure 2 gives an example.

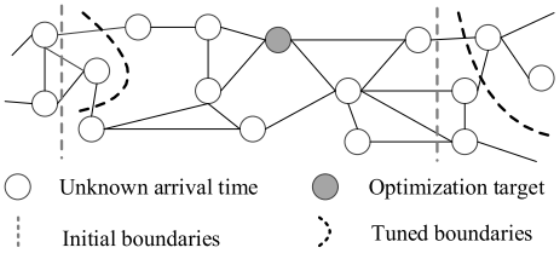

Figure 2: An example to illustrate how to extract a sub-graph without cutting too many edges. The initial solution cut 7 edges and the tuned solution only cut 4 edges.

An initial solution is generated based on two simple criteria. First, the extracted sub-graph contains a predetermined number of vertices. Second, the boundaries of the sub-graph need to be far from the vertex being calculated. Then, it employs a recently proposed massive graph cut algorithm BLP (balanced label propagation) to tune the cutting boundaries so that the number of cut edges is minimized. The BLP algorithm iteratively tunes the cutting boundaries till optimal. Here, the number of vertices in each extracted sub-graph is an important parameter. Clearly, there is a tradeoff between the accuracy of bounds and the computational efficiency. In the evaluation section, it will show the impact of the graph cut size. For each unknown arrival time, the extracted sub-graph includes

a number of vertices and edges. The upper bound and lower bound can be obtained by solving the two optimization problems considering the constraints related to these vertices and edges in the sub-graph. Since the number of constraints is much smaller, each optimization problem can be solved efficiently and the overall computing time is shortened significantly. It is worth noting that the bounds obtained by Domo are hard bounds. For the lower bound of an unknown t, Domo solves the optimization problem with optimization goal "Minimize" t and the three kinds of constraints. As long as the constraints hold, the lower bound is hard bound. In order to improve the computation efficiency, Domo includes relaxations to the constraints. These relaxations will cause Domo to obtain a smaller lower bound. Similarly, Domo will obtain a large upper bound after relaxations. Therefore, the bounds obtained after relaxations are looser bounds, but still hard bounds. Since the bounds obtained by Domo and MNT are both hard bounds, it compare their performances in terms of the tightness of the bounds in the evaluation.

4.2.4. Further Refinement of Arrival Time Estimation

After it has calculated the upper/lower bounds of each unknown arrival time, it can further refine the estimated values of the arrival times. It uses an optimization goal to estimate arrival times. The goal is to minimize the sum of variances of the node delays. The observation behind this goal is that within a relatively short period, the node delays of multiple packets on the same hop are similar. However, it also observes that a small number node delays can be much larger than others, which is possibly caused by transient link failure. These large node delays may affect the effectiveness of the optimization-based arrival time estimation. Therefore, it uses the calculated lower bound of each node delay to further refine the node delay estimation. Algorithm 1 gives the refinement process.

Algorithm 1 Delay Estimation Refinement

```
1: NODE-DELAY-ESTIMATION()
2: for each node delay D_n(x) do
3:     if min(D_n(x)) < AVG(D_n) + α·STD(D_n) then
4:         remove D_n(x) from the optimization problem
5: NODE-DELAY-ESTIMATION()
```

After Domo solves the optimization problem to estimate node delays (line 1), Domo will check whether a node delay is much larger than others. When the lower bound of a certain node delay is much larger than the average node delay on the same node (line 3), it is removed from the optimization problem (line 4). The constant α in the refinement algorithm (line 3) affects how it defines "much larger". In practice, α is set to be 1.5. Then Domo solves the optimization problem again with some nodes removed (line 5).

4.2.5. Sum of Node Delays Recording

In order to reconstruct the per-hop delay of each received packet, each node needs to record the sum of node delays in its local packets. A two-byte buffer is used to record the sum of node delays measured by the above technique. Algorithm 2 describes how the sum of node delays is recorded.

Algorithm 2 Sum of Node Delays Recording

```
1: sum-hop-delays = 0
2: event generates a local packet
3:     records current local time t_1
4: event receives a packet
5:     records current local time t_1 in SFD_RX handler
6: event forwards a packet or sends a local packet
7:     records current local time t_2 in SFD_TX handler
8:     sum-hop-delays += t_2 − t_1
9:     if sends a local packet then
10:        write sum-hop-delays to the transmission RAM
11:        sum-hop-delays = 0
```

For a local packet, its first node delay is recorded as the time difference between the generation time and the transmit SFD time (line 2, 3 and line 7). For a forwarded packet, its node delay is recorded as the time difference between two SFD interrupts (line 4, 5 and line 7). Whenever a new local packet is being transmitted, the sum of node delays is attached at the end of the new local packet (line 10). The buffer is cleared when the new local packet is transmitted (line 11). Note that the sum of node delays is stored at the end of each local packet. The reason is that the sum of node delays is calculated in the transmit SFD of the new local packet which is being transmitted. Writing the sum of node delays to the transmit FIFO RAM is required to be done before the radio actually transmits that part of the packet.

III. RESULTS AND DISCUSSION

Experimental Results

This chapter evaluates the performance of the Domo in terms of accuracy of both the estimated value and the bounds, by extensive simulations. Domo is implemented in Netbeans 8.0 and jdk 1.7.

First, wireless ad hoc network is formed. This network contains one sink node and many wireless ad hoc nodes. These wireless ad hoc nodes are located in various distances from sink. First Each Ad hoc nodes are connected with sink. At the connection time, sink forward each neighbor details to all connected nodes.

If a source node wants to forward any data to sink, it finds a path. Here path generation in 2 types. One way is random path, another way is shortest path.

Compared with Random Path, shortest path is better for transmission. Each node maintains FIFO Queue & a Routing Table. After path generation each node forwards the data through this path.

Each Ad hoc Node has 2 equipments. One is receiver another one is transmitter. Receiver used for receive packet and transmitter used for transmit packet.

In this project, one sink node and 5 ad hoc nodes are created and connected. A source node 1 sends the packet to sink node through intermediate nodes 3 and 5. It takes end-to-end delay is 13910 milliseconds. Furthermore, Domo achieves accurate per-hop per-packet delay reconstruction. It provide node 1 is per-hop per packet delay is 623 milliseconds, node 3 takes 558 milliseconds and node 5 takes 513 milliseconds. These Details are showed in Table 1.

Ad hoc Node Id	Per-Hop Per-Packet Delay (in ms)
1	623
3	558
5	513

Table 1: Per-Hop Per-Packet Delay (in ms)

IV.CONCLUSION

This thesis proposes Domo, a passive, lightweight and accurate delay tomography approach to decomposing the packet end-to-end delay into each hop. Domo reconstructs per-hop per-packet delays by solving a set of optimization problems. SDP relaxation technique is used to improve the efficiency of Domo. Further, an improved time window method and a graph cut method are used to deal with large number of unknowns in the optimization problem. It implements Domo in TinyOS and evaluates its accuracy in both simulations and test-bed experiments. Evaluation results show that Domo achieves higher accuracy.

V. Future Enhancement:

Domo does not consider Data security. For data security source node apply encryption for cryptography. At the encryption time, delay is occurred. After encryption data size is too large. So per hop per packet delay is increased. To tackle this problem, a novel Security with Domo approach needed.

VI. REFERENCES

[1]. J. Schiller, Mobile communication. Addison-Wesley Verlag, 2000.

[2]. S. Toumpis and D. Toumpakaris, "Wireless ad hoc networks and related topologies: applications and research challenges," e & i Electro technique und Information’s technique, vol. Volume 123, pp. 232-241, 2006.

[3]. M. Günes, B. Blywis, and F. Juraschek, "Concept and design of the hybrid distributed embedded systems testbed," August 2008.

[4]. Perkins and C. E., Ad hoc networking. Addison-Wesley Verlag, 2001.

[5]. "The official bluetooth."

[6]. L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X. Li, and G. Dai, "Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest," in Proceedings of SenSys, 2009.

[7]. M. Ceriotti, L. Mottola, G. P. Picco, A. L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon, "Monitoring heritage buildings

with wireless sensor networks: The torre aquila deployment," in Proceedings of IPSN, 2009.

[8]. X. Mao, X. Miao, Y. He, T. Zhu, J. Wang, W. Dong, X. Li, and Y. Liu, "Citysee: Urban CO2 monitoring with sensors," in Proceedings of INFOCOM, 2012.

[9]. 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 Proceedings of SenSys, 2010.

[10]. M. Patel and J. Wang, "Applications, challenges, and prospective in emerging body area networking technologies," IEEE Wireless Communications, vol. 17, no. 1, pp. 80-88, 2010.

[11]. M. Keller, L. Thiele, and J. Beutel, "Reconstruction of the correct temporal order of sensor network data," in Proceedings of IPSN, 2011.

[12]. J. Wang, W. Dong, Z. Cao, and Y. Liu, "On the delay performance analysis in a large-scale wireless sensor network," in Proceedings of RTSS, 2012.

[13]. Y. Gu and T. He, "Data forwarding in extremely low dutycycle sensor networks with unreliable communication links," in Proceedings of SenSys, 2007.

[14]. L. Ma, T. He, K. K. Leung, D. Towsley, and A. Swami, "Efficient identification of additive link metrics via network tomography," in Proceedings of ICDCS, 2013.

[15]. P. Sommer and B. Kusy, "Minerva: distributed tracing and debugging in wireless sensor networks," in Proceedings of ACM SenSys, 2013.

[16]. R. R. Kompella, K. Levchenko, A. C. Snoeren, and G. Varghese, "Every microsecond counts: tracking fine-grain latencies with a lossy difference aggregator," in Proceedings of ACM SIGCOMM, 2009.

[17]. M. Lee, N. Duffield, and R. R. Kompella, "Not all microseconds are equal: fine-grained per-flow measurements with reference latency interpolation," in Proceedings of ACM SIGCOMM, 2010.

[18]. M. Lee, S. Goldberg, R. R. Kompella, and G. Varghese, "Fine-grained latency and loss measurements in the presence of reordering," in Proceedings of the ACM SIGMETRICS, 2011.

[19]. K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, "Measurement and analysis of single-hop delay on an IP backbone network," IEEE J. Sel. Areas Commun., vol. 21, no. 6, pp. 908-921, Aug. 2003.

[20]. 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. ACM SenSys, 2012, pp. 15-28.

[21]. V. Sundaram and P. Eugster, "Lightweight message tracing for debugging wireless sensor networks," in Proc. IEEE/IFIP DSN, Jun. 2013, pp. 1-12.

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

[23]. Y. Gao et al., "Pathfinder: Robust path reconstruction in large scale sensor networks with lossy links," in Proc. IEEE ICNP, Oct. 2013, pp. 1-10.

[24]. Z. Liu, Z. Li, M. Li, W. Xing, and D. Lu, "Path reconstruction in dynamic wireless sensor networks using compressive sensing," IEEE/ACM Trans. Netw., vol. 24, no. 4, pp. 1948-1960, Aug. 2016.

[25]. X. Lu, D. Dong, X. Liao, and S. Li, "PathZip: Packet path tracing in wireless sensor networks," in Proc. MASS, 2012, pp. 380-388