18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Large-scale, Discrete IP Geolocation Via Multi-factor Evidence Fusion Using Factor Graphs

Sudhanshu Chandekar
Dept. of Electrical & Comp.Eng.
George Mason University
Fairfax, Virginia 22030
Email: schandek@gmu.edu

Bernd-Peter Paris
Dept. of Electrical & Comp.Eng.
George Mason University
Fairfax, Virginia 22030
Email: pparis@gmu.edu

***Abstract*—Traditionally, IP geolocation has been focused on finding the exact location of a given internet host. In this work, we focus on the problem of simultaneously geolocating large numbers of IP hosts to within a discrete set of geographical regions. Towards this objective, we present a fusion algorithm that combines information from multiple, heterogeneous sources of location information. Our algorithm estimates the most likely region for all hosts simultaneously. A key component of the algorithm is a systematic procedure for automatically constructing factor graphs to perform information fusion from a collection of delay measurements between hosts. While the paper focuses on the algorithmic aspects of the problem, we present initial results that demonstrate that hosts are placed in the correct region with very good accuracy.**

I. Introduction

IP geolocation refers to the practice of mapping an Internet host, identified by its IP address, to a physical location, e.g., a street address or geographic coordinates, latitude and longitude. By itself, an IP address does not provide any information about a host's location; therefore information from external sources is required to map an IP address to a geographic location.

Sources of geolocation can be broadly divided into three types: database-based, name-based and measurement-based [1]. Database-based sources [2] build and maintain repositories of IP address to location mappings. Geolocation databases are traditionally focused on pinpointing the location of end-systems, i.e., hosts that act as clients in internet applications. This focus is driven by the desire to provide location based services such as web advertising, content localization, fraud mitigation, and others. Due to this focus on end-hosts, geolocation databases tend to be unreliable in geolocating hosts in the core of the network.

Location information about devices in the core of the network, such as routers, can often be inferred from the names assigned to them. Name-based geolocation [3] uses hints, such as airport codes, encoded in hostnames to geolocate hosts. For example, the hostname `0.ae1.br2.iad8.alter.net` contains the airport code IAD (Washington Dulles Airport) which suggest that the host is located near that airport. Although such location hints tend to be highly reliable they are not always available, especially for hosts that do not reside in the core network.

Measurement-based systems [4] exploit network characteristics, such as path delay and topology, to estimate the location of a node. Measurement-based methods have been used to geolocate individual network hosts fairly accurately. But, such methods depend heavily on distributed "landmark" hosts and on redundant probes to pinpoint a single host. As a result, their ability to geolocate a large number of hosts is severely restricted.

In contrast to the above methods, our research aims to geolocate a large number of hosts simultaneously with reduced geographical resolution. We envision that our methods will directly support research on characterizing the spatial distribution of internet resources [5], efforts towards understanding the infrastructural readiness of a region to store and communicate information [6], and projects such as the Department of Homeland Security initiative to gain macroscopic insight into the global internet infrastructure [7]. In these applications, the emphasis is on determining which hosts lie within a given geographic region, which requires accurate, albeit coarser, geolocation for a large number of IP hosts.

In this paper, we introduce a method for large-scale, discrete IP geolocation; our method geolocates a large number of IP hosts simultaneously to a discrete set of geographic regions. At the core of our method is an algorithm for fusing heterogeneous information about hosts' locations. Specifically, we fuse data from three publicly available sources of information: the GeoIP database [2], the nslookup service [8], and host-to-host delay measurements obtained from the DIMES topological database [9]. The first two sources provide location evidence for individual nodes and therefore are referred to as node-local evidence. The third is the host-to-host delay that provides evidence about the relative separation of directly connected hosts, hence it is referred to as the delay evidence.

We derive statistical models to capture the uncertainty in each type of evidence and use Bayesian methods to fuse information from these models. To support simultaneous geolocation of a large number of hosts, we present a method for automatically constructing factor graphs from the network topology implied by delay measurements. Using factor graphs and the associated sum-product algorithm we derive an information fusion algorithm to find the most likely region for a large number of IP hosts, simultaneously.

978-0-9964527-1-7

This paper is organized as follows: Section II describes our sources of location information and, in particular, the DIMES database as a source of IP topology. Section III formalizes the discrete geographic model and presents models for each piece of location evidence. In section IV our algorithm for combining multi-factor evidence is presented. Section V presents results obtained using simulated data as well as data for an actual IP topology and delay measurements extracted from the DIMES database. Finally, section VI provides conclusions and discusses future work.

II. IP Topology

Of particular value for our work is the DIMES [9] database. The DIMES database is constructed from an ongoing campaign of systematic, distributed `traceroute` measurements [10]. From these measurements, The DIMES database infers links between IP hosts and the delay on these links.

Given a collection of IP links, an IP interface graph can be constructed by connecting links that have common nodes. For example, consider Fig.1, where in the first case a traceroute from host A to host B returns the IP addresses of the hosts (more correctly, interfaces) between A and B, here i, j and k. The linear IP topology of links $i \rightarrow j \rightarrow k$ thus obtained is indicated on the right. In the second case, due to redundant paths between A and B a traceroute probe may traverse routers $R1$, $R2$, $R3$ or just $R1$ and $R3$ returning links i, k, l, m or i, j, m. Connecting the links results in an IP topology that has a loop as shown on the bottom right.

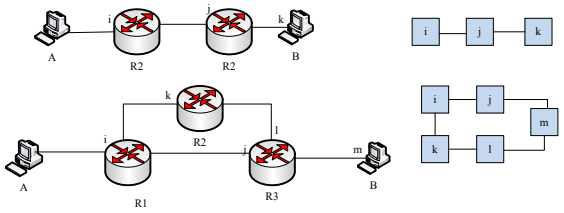


Fig. 1. Inferring IP topology from traceroute

These simple examples can be generalized to infer large IP topologies from the data in the DIMES topological database. In section IV-C, we will show how factor graphs can be synthesized directly from these inferred IP topologies.

Before we can derive our fusion algorithm, we present statistical models to describe the information sources that we seek to fuse.

III. Model

A. Discrete Region Model

To place an IP node in a distinct region, we define a finite set of discrete geographical regions. For example, we can adopt the Metropolitan Statistical Area (MSA) defined in [11]. For this paper, we assume that a finite discrete set of areas A is given and the total number of areas in the set is M.

Given an IP address n, there are M distinct regions where the node may be situated. We define a random variable R_n such that it reflects the index of the region from set A as the location for the node n. Formally, $R_n : A \mapsto S$ is a discrete random variable where $S \subset Z^+$. Subsequently, initial information about n's location is modeled by associating a probability mass function $F_{R_n}(r)$ with R_n. As mentioned in section I, an IP address by itself does not provide any location information and therefore a priori n is equally likely to be in any of the regions in set A, i.e., the prior probability mass function $F_{R_n}(r)$ is uniformly distributed. Throughout, we adopt vector notation in representing probability mass functions such that F_{R_n} is a M-vector defined over the space $[0, 1]^M$

B. Models for node-local evidence

The first piece of node-local evidence is the GeoIP database from Maxmind [2]. Although, such databases are good at mapping the IP address of end-systems to a physical location they are not universally reliable. For example, it has been observed [12] that administrative entities with hosts scattered in different locations register their assigned IP blocks with the geographical location of their headquarters. As a result, geographically dispersed IPs are erroneously geolocated to a single location by such databases. Consequently, such databases tend to be unreliable, especially for hosts in the network core. Let the probability that GeoIP locates an IP address correctly be denoted by α. Empirically, the reliability of the GeoIP database, has been estimated as $\alpha = 0.7$ for hosts residing in the United States [2].

Let R^G be a random variable defined over the same domain as R_n. When the GeoIP database is queried with the IP address n, the realization r of R^G indicates the index of the region where the corresponding host is believed to be located. The confidence in the observed evidence r is given by the conditional pmf $F_{R^G|R_n}(r|R_n) = Pr\{R^G = r|R_n = s\}$ which we model as

$$Pr\{R^G = r|R_n = s\} = \begin{cases} \alpha & \text{if } r = s, \\ \frac{1-\alpha}{M-1} & \text{if } r \neq s. \end{cases} \quad (1)$$

to reflect the accuracy of the GeoIP database.

The second piece of node-local evidence is the hostname associated with an IP address. Hostname lookup services, such as `nslookup` [8], map IP addresses to hostnames. The hostname may contain location hints. However, hostnames are not available for all IP addresses, in which case a query yields an empty result. When a hostname is available and contains a location hint, they provide a highly reliable source of location information. To model name-based location information, in addition to the M discrete regions, the set of possible outcomes must include an additional "none" outcome.

Let R^N be random variable defined on the expanded set of elementary outcomes given by $A \cup \{\text{"none"}\}$. Let the probability that a query results in a "none" reply be denoted as β. Further, let the probability that a naming hint geolocates

the address n correctly be denoted as ϵ. Our conditional pmf $F_{R^N|R_n}(r|R_n)$ for modeling naming information is, thus,

$$Pr\{R^N = r|R_n = s\} = \begin{cases} \beta & \text{if } r = \text{"none"}, \\ \epsilon & \text{if } r = s, \\ \frac{1-\beta-\epsilon}{M-1} & \text{if } r \neq s. \end{cases} \quad (2)$$

From available data [13], [14], we estimate that naming hints are available for half of all IP addresses ($\beta = 0.5$) and that $\epsilon = 0.45$.

C. Model for delay evidence

The third piece of evidence is the host-to-host propagation delay which reflects the relative separation of directly connected nodes. For example, short delays support the hypothesis that the connected hosts are in the same region. Past studies [15], [16] based on path delays measured via `ping`, have shown that a strong positive correlation exists between measured delay and physical distance. In our study of traceroute delays, experimental data have suggested similar positive correlation between relative delays and distance. However, it has been observed that for a given distance, there is considerable variance in observed latency measurements.

The relation between delay and distance can be obtained by fitting a linear regression model to the observed data

$$\hat{\theta} = m \cdot d + b, \quad (3)$$

where $\hat{\theta}$ is the mean delay for a given distance d and m and b are the slope and offset of the linear model respectively. In first approximation, the observed delay θ given distance d is modeled using a normal distribution,

$$f_{\Theta|D}(\theta|D = d) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\theta-\hat{\theta})^2}{2\sigma^2}}. \quad (4)$$

Since two regions uniquely determine distance d between nodes i and j the above function can be rewritten in terms of the regions R_i and R_j where the hosts at either end of the link are located: $f_{\Theta|D}(\theta|D = d) = f_{\Theta|R_i,R_j}(\theta|R_i = s, R_j = t)$. This formulation will allow us to infer in which regions hosts are located from measured delays.

In the data obtained from DIMES, we have noticed inconsistent delay values such as negative delays. It is known [17], that these result are an artifact stemming from the fact that some internet routers delay ICMP replies. As a consequence, estimated link delays may be negatively biased or highly positively biased depending on the position of such routers in the traceroute path. To model this effect, we postulate that the traceroute delays occur from one of three different distributions forming a Gaussian mixture model [18]. For data corresponding to each of the three component we calibrated slopes m and offsets b in the model as are $\underline{m} = (0.0159, 0.0135, 0.0186)$ and $\underline{b} = (0.4603, -14.358, 12.6855)$. The corresponding mixing proportions are $\underline{w} = (0.9, 0.08, 0.02)$. Finally, the standard deviations σ were determined as $\underline{\sigma} = (1.942, 3.552, 4.477)$.

Now that we have established the models for each piece of evidence, in the next section, we present our procedure for fusing information using these models.

IV. Fusing multi-factor evidence

In this section, we develop our algorithms for large-scale discrete geolocation. We begin with simple topologies to demonstrate that the Bayesian inference is correctly described by factor graphs before addressing the automated generation of factor graphs from arbitrary IP topologies.

A. Bayesian Inference

In section III, we have defined an a priori unconditional model for the location of IP nodes and conditional probability models for each piece of evidence. Now, we use Bayesian inference to compute the posterior distribution by updating the prior distribution, in the light of observed data. We adopt a two step approach for evidence fusion.

In the first step, an initial location estimate for a IP node n is formed from node-local evidence by fusing GeoIP and hostname evidence. Let $R^G = s$ and $R^N = t$ be the location evidence for IP address n from GeoIP and hostnaming, respectively. Assuming conditional independence of these variables, the Bayes rule is applied as follows

$$F_{R_n|r,s}(R_n|s,t) \propto F_{R^G|R_n}(s|R_n) \cdot F_{R^N|R_n}(t|R_n) \cdot F_{R_n}. \quad (5)$$

Note that the distribution $F_{R_n|s,t}$ obtained in (5) must be normalized to obtain the exact posterior; for notational simplicity, we do not include the scaling factor. This process can be carried out to obtain an initial location estimate for each IP node in a given topology from node-local evidence.

The second step is to incorporate the delay evidence to update location information about connected nodes. Suppose two IP nodes i and j are connected; then $F_{R_i|s_i,t_i}$ and $F_{R_j|s_j,t_j}$ are the posteriors obtained after incorporating node-local evidence (5). Let $\Theta_{ij} = \theta_{ij}$ be the observed delay on the link connecting nodes i and j. Then, $f_{\Theta_{ij}|R_i,R_j}(\theta_{ij}|R_i, R_j)$ gives the likelihood of observing delay θ_{ij} for the distance between the two regions. The joint conditional distribution of variables R_i and R_j is obtained using the Bayes rule as

$$\begin{aligned} F_{R_i,R_j|\Theta,\underline{R}^G,\underline{R}^N}(R_i, R_j|\theta_{ij}, \underline{s}, \underline{t}) \quad &\propto \\ \underbrace{f_{\Theta_{ij}|R_i,R_j}(\theta_{ij}|R_i, R_j)}_{\text{Delay evidence}} \cdot & \\ \underbrace{F_{R^G|R_i}(s_i|R_i) F_{R^N|R_i}(t_i|R_i)}_{\text{i's node-local evidence}} \cdot & \\ \underbrace{F_{R^G|R_j}(s_j|R_j) F_{R^N|R_j}(t_j|R_j)}_{\text{j's node-local evidence}} \cdot \underbrace{F_{R_i}}_{\text{i's prior}} \cdot \underbrace{F_{R_j}}_{\text{j's prior}}. & \end{aligned} \quad (6)$$

The location estimates for IP nodes i and j is obtained by marginalizing the joint posterior (6). Finding the posterior distributions for a pair of connected nodes is simple, but the problem becomes intractable when dealing with a large number of connected nodes. In such cases, it is important to have a method that systematically captures the dependency between variables and allows for the computation of the marginal pmfs for a large number of connected nodes.

B. Factor graphs for mapping IP topologies

Before, we illustrate the automatic construction of factor graphs from IP topologies, we briefly review factor graphs.

1) Overview: Factor graphs [19] integrate a graph theoretic formalism for large systems with probabilistic modeling. They can represent complex global functions of many variables and encode the relations between them. For example, if a multivariate function, such as the joint probability function (6), can be factored into a product of functions then it can effectively be represented via Factor graphs. Throughout, we use the Forney-style factor graphs (FFGs) [20]. Following is the summary of conventions followed in building the FFG representation.

- Factorized functions are represented by nodes and variables participating in each factor represent the connected edges to the factor.
- A variable that occurs only in one factor is represented as a half edge.
- Generally, a variable can only be shared between two factors but this restriction is easily circumvented by introducing equality constraint nodes.

Factor graphs are associated with the sum-product algorithm [21] that operates by message passing along the edges of the factor graph. The sum-product algorithm can be thought of as an elimination algorithm for multiple variables. Each time a summation is calculated an expression is produced which can be considered as a message to be passed to the next summation. Continuing this process of summation to generate messages (partial sums) the final expression thus obtained is a function of the variable to be marginalized. Following are the rules for message computation.

- A half edge does not carry any message towards the connected factor. Alternatively, a constant function such as a uniform distribution can be assumed on such edges.
- The message out of a factor node along an edge is the summarized (marginalized) product of the local function and the received messages on all connected edges except the edge on which the message is to be sent.
- The marginal distribution of a variable can be obtained by forming the product of messages traveling along the edge in opposite direction.
- Known or observed variables are simply plugged into the corresponding factors.

For detailed treatment of factor graphs and the summary propagation algorithms, the reader is directed to [19], [22].

2) Example 1: Factor Graphs for Geolocating a Single IP Node: We now demonstrate how the Bayesian inference for node local evidence (5) can be captured by a factor graph. Each conditional model along with the a priori model can be represented by a factor node and the random variables involved in each model are represented by edges as per the rules for constructing FFGs. Since the two conditional distributions share a common variable R_n, we use the equality constraint node $f_{=}(r, r^{(1)}, r^{(2)}) = \delta(r - r^{(1)})\delta(r - r^{(2)})$ where $\delta(\cdot)$ is a Kronecker delta function. The equality constraint node "clones" the variables so as to share them between at most two factors. The factor graph capturing node-local evidence for an IP node n is illustrated in Fig. 2.

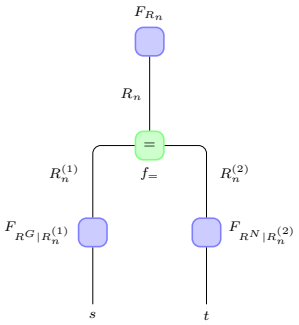


Fig. 2. Factor graph for combining node-local evidence for a single IP node

This simple factor graph represents the a posteriori function $F_{R_n|s,t}(R_n|s,t)$ for local evidence at IP node n.

a) Sum-Product Algorithm for Combining Node-Local Evidence: The sum-product algorithm fuses the node-local evidence and we will show that it produces exactly the right hand side of (5). The computation is performed via message passing over the factor graph of Fig. 2. In such factor graphs message passing is initiated from the leaf nodes. F_{R_n} is a leaf node connected to only one edge R_n. The message along edge R_n is the probability vector F_{R_n}. Thus, the message along the edge R_n is given by $\mu_{F_{R_n}\to R_n} = F_{R_n}$. Per the sum-product rule, the observed GeoIP evidence $R^G = s$ is plugged into the factor, $F_{R^G|R_n}(s|R_n)$, giving the message $\mu_{F_{R^G|R_n}\to R_n^{(1)}} = F_{R^G|R_n}(s|R_n)$ along edge $R_n^{(1)}$. Similarly, the observed naming evidence $R^N = t$ for a node n is plugged into the factor $F_{R^N|R_n}(s|R_n)$ giving the message $\mu_{F_{R^N|R_n}\to R_n^{(2)}} = F_{R^N|R_n}(t|R_n)$ along edge $R_n^{(2)}$. Due to the conditional independence of GeoIP and naming evidence, the messages from the factors representing conditional pmfs can be combined simply by forming an elements-wise product of vectors at the equality constraint node

$$\begin{aligned}
\mu_{f_{=}\to R_n} &= \sum_{r^{(1)}}\sum_{r^{(2)}} \delta(r - r^{(1)})\delta(r - r^{(2)}) \\
&\quad \mu_{F_{R^G|R_n^{(1)}}\to R_n^{(1)}}\mu_{F_{R^N|R_n^{(2)}}\to R_n^{(2)}} \\
&= \mu_{F_{R^G|R_n}\to R_n} \cdot \mu_{F_{R^N|R_n}\to R_n} \qquad (7) \\
&= F_{R^G|R_n}(s|R_n) \cdot F_{R^N|R_n}(t|R_n). \qquad (8)
\end{aligned}$$

As discussed earlier, the a posteriori pmf is obtained by forming the product of messages traveling in the opposite direction along the edge R_n and is given by

$$\begin{aligned}
F_{R_n|R^G,R^N} &= \mu_{f_{=}\to R_n} \cdot \mu_{F_{R_n}\to R_n} \\
&= F_{R^G|R_n}(s|R_n) \cdot F_{R^N|R_n}(t|R_n) \cdot F_{R_n}. \qquad (9)
\end{aligned}$$

Obviously, equations (5) and (9) are identical, demonstrating that the posterior pmf is correctly computed via the above factor graph.

3) Example 2: Factor Graphs for Geolocating a Network of Two Nodes: In the next step, we demonstrate that a factor graph can correctly represent two connected nodes, i and j, and the sum product algorithm produces the posterior pmf for this case (6). For each node, node-local evidence is available that is fused as described above and captured in Fig. 2. The additional factor is the delay model given by $f_{\Theta_{ij}|R_i,R_j}(\theta|R_i,R_j)$. This is a function of two discrete variables R_i and R_j since the delay θ is observed between nodes i and j. Therefore each of these variables now need to be shared between the factors representing the three pieces of evidence using the equality constraint node $f_{=}$. The resulting factor graph is shown in Fig. 3.

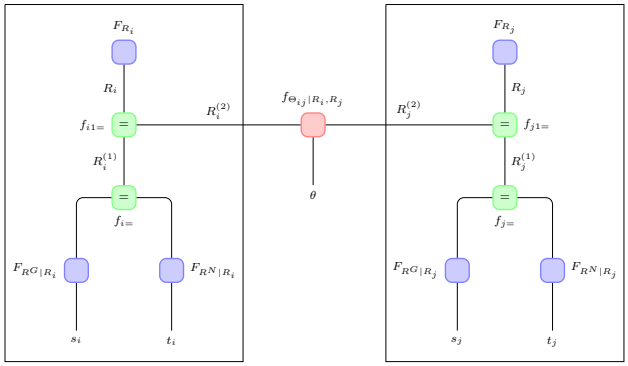


Fig. 3. Factor Graph representation for combining delay and node-local evidence to simultaneously geolocate a pair of nodes i,j

a) Sum-Product Algorithm for Combining Delay Evidence with Node-Local evidence for a Pair of Nodes: Equation (6) gives the conditional joint pmf of two nodes, given the node-local and delay evidence. To find the marginal distribution for a node i or j, the conditional joint pmf needs to be summarized over all variables except the variable to be marginalized. Like the factor graph of Fig. 2, here too the message passing starts from the leaf nodes and the messages are combined at the equality constraint nodes $f_{i1=}$ and $f_{j1=}$. The output messages from these nodes are combined with the local function $f_{\theta|R_i,R_j}$ and subsequently update the other node. Note that the two nodes "trade location information" via the messages passed through the delay factor.

Alternatively, the message passing can be viewed as occurring in two steps. In the first step, the posterior due to the node-local evidence at each node is computed, and in the second step, they are combined with the local function for the link evidence. This is a valid way to pass messages because of the local elimination property [19] of the sum-product algorithm wherein intermediate summaries are formed by grouping factors into subsystems and applying the sum-product rule to each subsystem. This property of forming subsystems has important applications to our problem as now each subsystem corresponds to a physical node in the IP network. The box drawn around the factors represent a subsystem in Fig. 3

The message computation for each node due to node-local evidence is identical to the one shown in section IV-B2. Therefore, we show only messages that illustrate how the new delay evidence is used in computing marginals. For nodes i and j the updated posterior after observing node-local evidence are $F_{R_i|R^G,R^N}$ and $F_{R_j|R^G,R^N}$. Following are the messages that are computed at the factor representing the delay evidence.

$$\begin{aligned}\mu_{f_{\Theta|R_i,R_j}\to R_i^{(2)}} &= \sum_{r_j} f_{\Theta|R_i,R_j}\cdot\mu_{f_{j1=}\to R_j^{(2)}} \\ &= \sum_{r_j} f_{\Theta|R_i,R_j}\cdot F_{R_j|R^G,R^N} \qquad (10)\end{aligned}$$

$$\begin{aligned}\mu_{f_{\Theta|R_i,R_j}\to R_j^{(2)}} &= \sum_{r_i} f_{\Theta|R_i,R_j}\cdot\mu_{f_{i1=}\to R_i^{(2)}} \\ &= \sum_{r_i} f_{\Theta|R_i,R_j}\cdot F_{R_i|R^G,R^N} \qquad (11)\end{aligned}$$

Finally the updated posterior after including the link evidence for nodes with IP address i and j is obtained by forming the product of messages traveling in opposite directions on edges R_i and R_j respectively and is given by

$$\begin{aligned}F_{R_i|\Theta_{ij},\underline{R}^G,\underline{R}^N} &= \mu_{f_{i1=}\to R_i}\mu_{f_{\Theta_{ij}|R_i,R_j}\to R_i} \qquad (12)\\ &= F_{R_i|R^G,R^N}\sum_{r_j} f_{\Theta_{ij}|R_i,R_j}F_{R_j|R^G,R^N}\end{aligned}$$

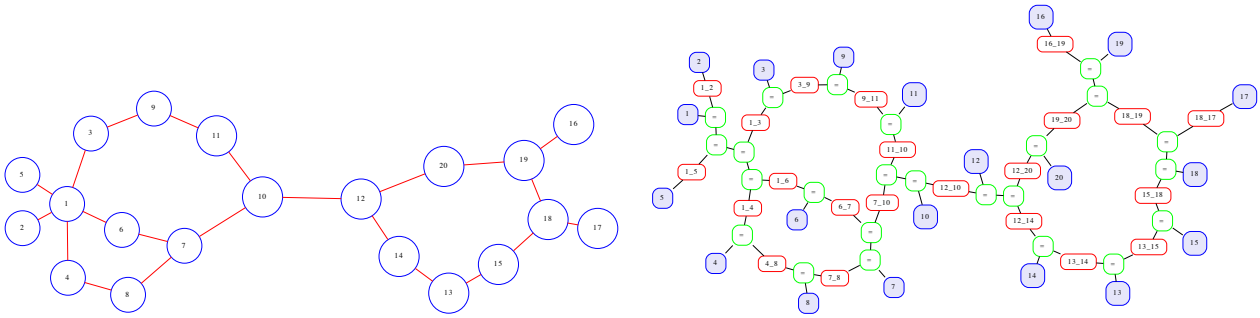
$$\begin{aligned}F_{R_j|\Theta_{ij},\underline{R}^G,\underline{R}^N} &= \mu_{f_{j1=}\to R_j}\cdot\mu_{f_{\Theta|R_i,R_j}\to R_j} \qquad (13)\\ &= F_{R_j|R^G,R^N}\cdot\sum_{r_i} f_{\Theta_{ij}|R_i,R_j}\cdot F_{R_i|R^G,R^N}\end{aligned}$$

Note that equation (12)-(13) are the marginals of the global function. Also, the algorithm stops automatically after computing messages along each direction of each edge in the factor graph. Such factor graphs in which the algorithm halts after computing exact marginals are known as acyclic factor graphs.

C. Factor Graphs for Simultaneous Geolocation of Nodes in Arbitrary IP Topologies

Building on the simple examples in the preceding sections, we will now turn to arbitrary topologies. In the following discussion, we present two algorithms, one for automated construction of factor graphs from an arbitrary network topology and the second for executing the iterative sum-product algorithm on such factor graphs. The first algorithm illustrates the close relationship between the topology structure and the factor graph structure. Each host of the IP network is represented by a node-factor (see section IV-B2) and each link is represented by a delay-factor (see section IV-B3) in the corresponding factor graph. If a host is connected to multiple links then the corresponding node-factor is connected to multiple delay-factors via equality-factors.

Fig.4 shows a network and the corresponding factor graph for a network of 20 nodes and 21 links. Note how the the factor graph preserves the network structure implied by host-to-host delays. A step-by-step procedure for creating a factor graph



(a) Physical IP topology

(b) Factor graph corresponding to the IP topology

Fig. 4. Automatic construction of factor graphs for an arbitrary IP topology

representation for an IP topology implied by a collection of links is as follows:

1) For each link (i, j) create one delay-factor and check if factors corresponding to each node have already been created.
2) If a node-factor does not exist, create a node-factor and point it to the link-factor. Similarly point the link-factor to the node-factor.
3) If a node-factor exists, create an equality-factor and update the existing pointers of the link-factor and the node-factor. Subsequently update the pointers of the equality-factor.
4) Repeat steps 1 to 3 for all links in a given physical topology

In section II, it was shown that due to redundant paths between hosts, IP networks may have loops causing cycles in the corresponding factor graph. For factor graphs with cycles, the sum-product algorithm does not terminate automatically. In such cases, the iterative form of the sum-product algorithm can be used to produce approximate marginal distributions. Such distributions are close to the true distributions whenever the algorithm successfully converges [23].

To understand how the sum-product algorithm applies to arbitrary IP graphs consider the following example. Three IP nodes i, j, and k are connected in a cyclic manner such that each node has links to the other two. Let $\underline{\Theta}$ be the delays observed on all links in the network and $\underline{R}^G$, $\underline{R}^N$ be the node-local evidence for the three nodes. The conditional, joint pmf for this network has the following factorization

$$F_{R_i,R_j,R_k|\underline{\Theta},\underline{R}^G,\underline{R}^N} \propto f_{\Theta_{ij}|R_i,R_j} f_{\Theta_{jk}|R_j,R_k} f_{\Theta_{ki}|R_k,R_i} F_{R_i|R^G,R^N} F_{R_j|R^G,R^N} F_{R_k|R^G,R^N}. \quad (14)$$

This correspondence between nodes and links of the IP network and the nodes of factor graph is illustrated in Fig.5.

For the factor graph of Fig.5, the messages due to node-local evidence are sent along the edges R_i, R_j, R_k. By the sum-product rule, a factor computes an output message along an edge only if there are messages on all other connected edges. In Fig. 5, a deadlock occurs as none of the equality nodes $f_{i1=}, f_{j1=}, f_{k1=}$ have received messages on at least two edges. To break such deadlocks a place-holder message, e.g. a uniform pmf vector is injected along a randomly chosen edge and message passing is initiated. Also, once such deadlocks are broken, the messages circulate endlessly in the factor graph unless a stopping criterion is specified. We use the Kullback-Leibler (KL) divergence, as a measure of the change between two consecutive messages, for this purpose. The message passing stops when the change in the message is below a pre-determined threshold κ.

Following is the outline for implementing our iterative sum-product algorithm.

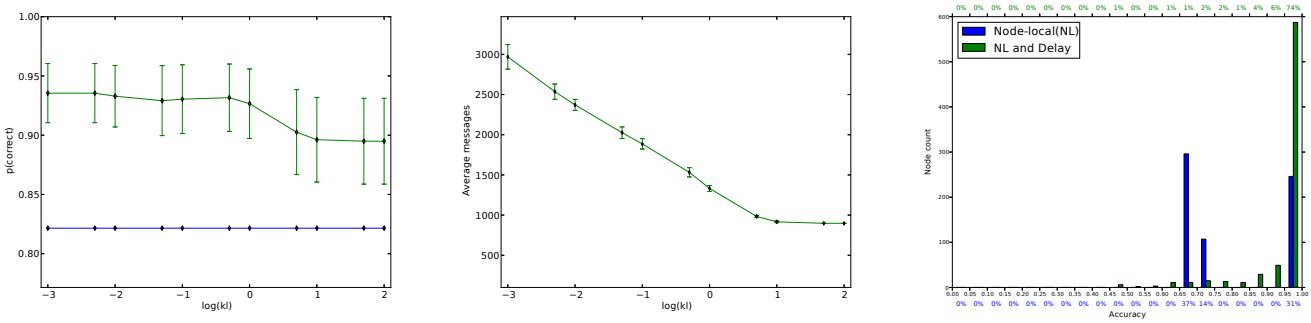
1) Set initial conditions for execution: Compute messages at each node by combining node-local evidence and start message passing.
2) Computation of messages at each node in the factor graph: Compute a new message along an edge if there are messages along all other edges and pass resulting messages to the next node.
3) If a deadlock occurs, randomly choose an equality factor and inject a place-holder message to initiate message passing.
4) Convergence detection: In each iteration find the KL distance between the message in the previous iteration and the message in the current iteration. Pass a new message only if KL distance is above threshold κ.
5) Repeat steps 2 to 4 until no more messages exist.

When the algorithm stops all factor nodes must have received at least one message other than the place-holder message, if any, along each connected edge and the change in the message along an edge is less than the KL threshold κ.

V. Results

A. Simulation study

To test our method, we extracted an IP subnetwork from the DIMES edges file by using the Internet2 [24] address block and matched it against the DIMES database to obtain links with at least one address belonging to that IP block. The topology thus obtained, consisted of 79 IP nodes and



(a) Accuracy vs κ (b) Avg. messages vs κ (c) Node count vs Accuracy

Fig. 6. Simulation Results for an IP topology

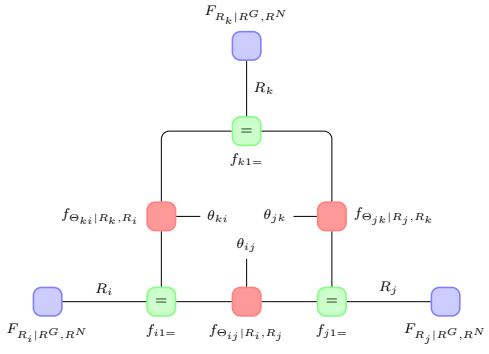


Fig. 5. Factor graphs with cycles for a network of three nodes

132 IP links. Using the algorithms in the previous section we constructed the corresponding factor graph and computed the posterior pmfs.

As discussed above, factor graphs with cycles compute messages endlessly unless a stopping criterion is specified. We rely on the KL distance dropping below threshold κ for this purpose. The parameter κ is the main factor responsible for the convergence of the iterative algorithm. We say the algorithm has converged if in successive iterations the KL distance between messages at all the nodes is less than κ. If κ is chosen to be too large ($\kappa \gg 0$), convergence is insufficient, and location estimates are inaccurate. If κ is chosen too small ($\kappa \approx 0$) then the convergence is delayed without any appreciable change to the marginal distributions of each node. Moreover, this leads to the increase in the number of messages thereby affecting the execution time of the algorithm.

To understand how this parameter affects the accuracy and the execution time of the algorithm we conducted a simulation study on the candidate topology where the node-local and delay evidence were generated from the statistical models presented in section III. The effect of κ on accuracy is plotted in Fig. 6a and on execution time (measured in terms of number of messages) in Fig. 6b.

Fig. 6a, shows that, before the iterative algorithm was executed, 82% of hosts were located to their true region using only the node-local evidence. The iterative sum-product algorithm was run 10 times for each value of κ and the accuracy is measured as the mean proportion of nodes geolocated to their true location denoted as $p(correct)$. Variations between runs originate from randomly selected nodes for breaking deadlocks. We decide the most likely region of a host by the maximum value of the posterior probability denoted as p_{max}. For values of $\kappa > 0.5$ marginal functions are inaccurate and the value of $p(correct)$ drops from 0.92 to 0.89. Also there is significant variation in the values of $p(correct)$. But, as κ is decreased ($\kappa \leq 0.005$) the mean accuracy increases to 0.93 and the variation in $p(correct)$ is reduced signifying that the accuracy in each run of the experiment is closer to the average.

Fig. 6b, reveals that as $\kappa \to 0$ the total number of messages passed for achieving convergence increases. The number of messages increase exponentially (linear on a logarithmic scale) as $\kappa \to 0$. Also, a comparison between Fig. 6a and Fig. 6b, reveals that there is no appreciable change in $p(correct)$ (0.925 to 0.930) for $\kappa < 0.5$. This implies that the convergence can be achieved with fewer messages without adversely affecting the accuracy of the algorithm.

Also, Fig.6c illustrates the increase in the confidence with which each node in the IP topology is geolocated to their true region. Out of the 82% of nodes geolocated due to node-local evidence 51% of nodes are located to their true location with probability in the range $0.65 \leq p_{max} \leq 0.75$ whereas only 31% of nodes were geolocated with a probability in the range $0.95 \leq p_{max} < 1$. After incorporating the delay evidence the reliability increased such that 74% of nodes were geolocated within the range $0.95 \leq p_{max} < 1$ and 16% in the range of $0.65 \leq p_{max} < 0.95$ with the total percentage of nodes correctly geolocated at 92%.

B. Internet2 Data

In this section, we present results obtained by executing the iterative sum-product algorithm for the candidate topology but with the actual (measured) delay data. To validate the accuracy of our method on real data, it is important to have ground-truth information about each node in the topology. Although

Internet2 publishes its topology maps, it is very difficult to obtain ground-truth for each of the IP address especially for those nodes that are connected to Internet2 nodes. We infer ground-truth indirectly by choosing geographically distributed landmarks and conducting traceroutes to each of the IP address discovered. The true location of the IP node was based on the hostname of the router one hop away from the target IP. The ground-truth information thus obtained, divided the nodes into $M = 5$ regions (Atlanta, GA, Washington, DC, New York, NY, Chicago, IL and Cleveland, OH).

The result obtained after running the algorithm with real data is shown in Fig.7. Since the topology has been derived for Internet2 hosts which are mostly routers, as expected, the GeoIP database fails to provide evidence about any of the nodes. On the contrary the nslookup service provides names for 99% of IP hosts. Fig.7 shows that only 3% of nodes were located with high confidence ($p_{max} > 0.95$) given the node-local evidence whereas 81% of nodes had $p_{max} > 0.95$ after fusing the delay evidence. Our method accurately geolocates 97% of the nodes to the correct location ($p(correct) = 0.97$). Note that the total number of nodes correctly geolocated is slightly smaller when compared to only node-local evidence, $p(correct) = 0.99$. This appears to be an artifact due to the fact that regions Atlanta, New York City and Chicago are approximately equidistant. As a result, three nodes belonging to Chicago and having links to Atlanta are being placed in New York City. Moreover, such nodes do not have additional connections due to the limited topology size considered. We believe that in a larger topology such nodes would have additional links that would pin such nodes to their correct location.

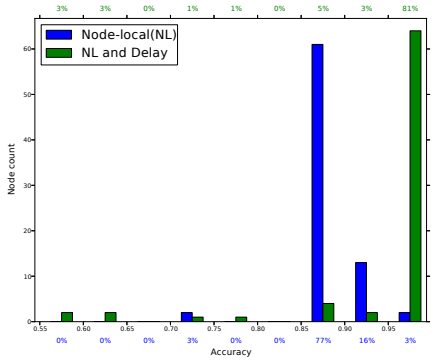


Fig. 7. Results for Internet2 IP topology

VI. Conclusion and Future work

A number of IP geolocation techniques are available that cater to applications that require precise location of each IP host. For applications that require the location of a large number of internet hosts, such techniques are unreliable and incomplete in the information they provide.

The main contribution of this paper is a new algorithm, that fuses multiple pieces of evidence from heterogeneous data sources. More importantly, the algorithm achieves simultaneous geolocation of a large number of hosts by automatically constructing factor graphs for any arbitrary IP topology. The algorithm is applied to a real network and is shown to accurately place nodes in the correct regions with high confidence.

For future work, we will address the scalability of our method and apply our algorithm to truly large-scale networks. Additionally, we will explore incorporating additional pieces of evidence to further strengthen the accuracy of the algorithm.

References

[1] M. Crovella and B. Krishnamurthy, *Internet Measurement: Infrastructure, Traffic and Applications*. New York, NY, USA: John Wiley & Sons, Inc., 2006.

[2] Geoip. Maxmind LLC. [Online]. Available: http://www.maxmind.com

[3] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," in *ACM SIGCOMM Computer Communication Review*, vol. 31. ACM, 2001, pp. 173–185.

[4] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, "Constraint-based geolocation of internet hosts," *Networking, IEEE/ACM Transactions on*, vol. 14, no. 6, pp. 1219–1232, 2006.

[5] P. Mátray, P. Hága, S. Laki, G. Vattay, and I. Csabai, "On the spatial properties of internet routes," *Computer Networks*, vol. 56, no. 9, pp. 2237–2248, 2012.

[6] M. Hilbert and P. Lopez, "The worlds technological capacity to store, communicate, and compute information," *Science Magazine*, vol. Vol 332, February 2011.

[7] Leveraging the science and technology of internet mapping for homeland security. CAIDA. [Online]. Available: http://www.caida.org/funding/cybersecurity/

[8] G. Kessler and S. Shepard, "A primer on internet and TCP/IP tools," *RFC 1739*, December 1994.

[9] DIMES. IP topology. [Online]. Available: http://www.netdimes.org

[10] T. Kernen, "traceroute.org." [Online]. Available: http://traceroute.org

[11] Metropolitan statistical areas. United States Census Bureau. [Online]. Available: http://www.census.gov/population/metro/data/

[12] B. Huffaker, M. Fomenkov, and K. Claffy, "Geocompare: a comparison of public and commercial geolocation databases," *Proc. NMMC*, 2011.

[13] Carna botnet, internet census 2012. [Online]. Available: http://internetcensus2012.bitbucket.org/paper.html

[14] T. Krenc, O. Hohlfeld, and A. Feldmann, "An internet census taken by an illegal botnet: a qualitative assessment of published measurements," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 103–111, 2014.

[15] M.J.Arif, S.Karunasekara, S.Kulkarni, A.Gunatilaka, and B.Ristic, "Internet host geolocation using maximum likelihood estimation technique," *IEEE International Conference on Advanced information Networking and Applications*, 2010.

[16] I. Youn, B. L. Mark, and D. Richards, "Statistical geolocation of internet hosts," in *18th Internatonal Conference on Computer Communications and Networks*. IEEE, 2009, pp. 1–6.

[17] A. Broido, Y. Hyun *et al.*, "Spectroscopy of traceroute delays," in *Passive and Active Network Measurement*. Springer, 2005, pp. 278–291.

[18] D. Reynolds, "Gaussian mixture models," *Encyclopedia of Biometrics*, pp. 659–663, 2009.

[19] H.A.Loeliger, "An introduction to factor graphs," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 28–41, 2004.

[20] G. D. Forney Jr., "Codes on graphs: Normal realizations," *IEEE Trans. Inform. Theory*, vol. Volume 47, pp. 520–548, 2001.

[21] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transaction on Information Theory*, vol. 47, pp. 498–519, 1998.

[22] H. A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," in *Proceedings of the IEEE*, vol. vol. 95, 2007, pp. 1295–1322.

[23] J. M. Mooji and H. J. Kappen, "Sufficient conditions for convergence of the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. Volume 53, 2007.

[24] Internet2 connectivity map. Internet2. [Online]. Available: http://noc.net.internet2.edu/i2network/research-and-education-network.html