Toward Fast and Efficient IP-level Network Topology Capture

Thomas Bourgeau, Timur Friedman
UPMC Sorbonne Universités, LIP6 and CNRS

ABSTRACT
This paper presents a new network topology capture algorithm, NTC, which aims to better capture network dynamics through accelerated probing, reducing the probing load while maintaining good coverage. There are two novel aspects to our approach: it focuses on obtaining the network graph rather than a full set of individual traces, and it uses past probing results in an adaptive way to guide future probing. We evaluate our algorithm on real traces and demonstrate outstanding performance compared to existing work.

Categories and Subject Descriptors
C.4 [PERFORMANCE OF SYSTEMS]: Measurement techniques

Keywords
Measurement Systems, Complex Networks, Traceroute

1. INTRODUCTION
Large-scale distributed traceroute-based measurement systems are used to obtain the topology of the Internet at the IP-level which serves as the basis to monitor and understand the behavior of the network. However, existing approaches to measuring the public IPv4 network space have focused on conducting full end-to-end route traces and as a result take a considerable time, in the order of days, to obtain a full graph of the network which is too slow to capture much of the network’s dynamics [2]. The reason of such low performances is due to the high discovery budget which is the number of measurement probes needed to discover the resulting IP-level network topology.

There has been earlier work on improving the efficiency of measurement systems so as to minimize the discovery budget. For example, Donnet et al.’s Doubletree lowers the discovery budget by introducing stopping rules for nodes already discovered [3] and Shavitt et al. have suggested to validate the presence of previously measured links in a minimum time of approximately O(log n) [4]. Our approach differs from existing network topology discovery methods, in that we take advantage of previously detected topological features to guide the sampling of the network topology and we look at the impact of capturing network dynamism with partial traceroutes sampling with adaptive probing strategy to track dynamism events along the paths where topological changes have been diagnosed.

This paper describes our network topology capture algorithm (NTC) and evaluates it based upon distributed traceroute measurements collected on PlanetLab [1]. Emulating NTC on this dataset, we find that our approach is extremely efficient in reducing the discovery budget which translates directly into more efficient capture of network topology.

2. NOTATION AND FORMALISM
A distributed IP-level measurement system consists of a set of distributed sources $S$ that discover the paths $\vec{v}_p$ towards a set of destinations $D$ using a probing tool, such as traceroute. A round $r$ corresponds to the time needed by the system to discover the topology represented as a directed graph $G_r = (\mathcal{V}_r, \mathcal{E}_r)$, where the edge set $\mathcal{E}_r$ and the vertex set $\mathcal{V}_r$ are built from the aggregation of all discovered paths found. We denote as traceroute query $q_j \in Q_r$ the probing parameters used by a trace function $T_r$ to discover a given edge $e_k \in \mathcal{E}_r$ such that $T_r(q_j) = e_k$ and $Q_r$ represents the query set for a given round $r$. As a given edge can be seen by several traceroute queries, we define as redundancy set $C_{e_k}^r = \{q_j(T_r(q_j) = e_k)\forall q_j \in Q_r\}$ the set of all queries that have seen the same edge at a given round $r$. Finally, we denote as dynamism events the appearance and disappearance of graph elements between consecutive rounds.

3. NETWORK TOPOLOGY CAPTURE
We present in Fig. 1 the four phases workflow of our pro-

Figure 1: NTC algorithm overview
posed Network Topology Capture (NTC) algorithm that exploits probing redundancy to lower the discovery budget while tracking dynamism events when changes are observed.

Phase 1: Our algorithm starts by performing a complete discovery of the network topology by issuing full traceroute to collect all possible paths between the entire set of sources $S$ toward all the destinations $D$. This action is performed by the FullProbe functionality that returns a complete graph $G$, and an exhaustive redundancy set $C_r$.

Phase 2: Our algorithm exploits the probing redundancy learned in previous rounds to reduce discovery budget through the Dispatcher functionality, that selects randomly $\alpha$ queries for each known edge of the redundancy set learned in previous rounds. The resulting sub-set $C_{r_k} \subseteq C_{r_{k-1}}$ is then used to validate each edge with limited queries.

Phase 3: Based on the built redundancy sub-set, each probing agents validate the presence of the edge associated to the selected query. If a change is detected ($T_r(\phi) \neq T_{r+1}(\phi)$), the agent explore the possible additional changes that might occur along the detected changed path until $\beta$ ips are discovered in the backward and forward direction from the detected change. This is done by the Prober functionality that returns the new sub-path and queries discovered when changes are detected.

Phase 4: For each change detected, we update the redundancy set $C_r$ and redundancy sub-set $C_{r_k}$ by removing the changed query and by adding new edges and queries discovered. This is done by the Updater function, that returns the updated redundancy set and the discovered graph built from the redundancy sub-set found at that round.

Note that our algorithm can be tuned by two parameters; the $\alpha$ parameter provides a threshold on the probing reduction to reach and the $\beta$ parameter limits the exploration of dynamism events on changed paths.

4. NTC PERFORMANCE EVALUATION

In this section, we propose to evaluate the performance of our NTC algorithm through a trace-driven simulation based on our own measurement dataset [2] collected during 2 months over PlanetLab overlay [1].

Discovery budget: The discovery budget saved by our algorithm, is given by $B(\alpha, \beta)$, which is the ratio of the number of probes generated by our NTC algorithm to the discovery budget of our measurement dataset. We see clearly in Fig. 2(a) that our algorithm allows to save at most 92% of measurement probes when $\alpha = 1$ and $\beta \leq 30$; which would accelerate probing rounds 17 times faster. Even in the worst case, our algorithm outperforms state of the art approaches such as Doubltree that allows to save at most 75% of discovery budget.

Network topology coverage: We analyze the effect of our NTC algorithm on the topology coverage by observing $\Phi(\alpha, \beta)$ which is the ratio of the edges captured by our NTC algorithm to the edges captured by our measurement dataset. We observe in Fig.2(b) that we obtain high edge coverage that reach 95% which is much better than Doubltree that needs higher discovery budget to reach at most 89% of edge coverage.

Dynamism events captured: To quantify the efficiency of our algorithm in capturing network topology dynamics, we introduce $\Omega(\alpha, \beta)$ which is the ratio of the number of dynamism events captured by our NTC algorithm to the number of dynamism events captured by our measurement dataset. We notice in Fig.2(c) that our algorithm can reach high edge dynamism capture (75% for $\alpha = 10$ and $\beta = 30$). Thus, our algorithm is the first approach that obtains reasonable network topology dynamism capture with low discovery budget.

5. CONCLUSION

This paper describes a new network topology algorithm (NTC) that demonstrates how large-scale measurement systems can be operated with a reduced discovery budget, so as to better capture network dynamics while at the same time maintaining broad coverage of the network topology. Its novel approach is based on knowledge of previously learned graph elements, combined with adaptive probing. As our algorithm outperforms existing approaches, it opens the way to new large-scale measurement systems deployment with fast and efficient capture of network topology dynamics.

6. REFERENCES


1The research leading to these results has received funding from the EU FP7 OpenLab project (Grant No. 287581)