A Topology Discovery Module based on a Hybrid Methodology

S. Avallone S. D'Antonio M. Esposito A. Pescapè S. P. Romano

University of Napoli "Federico II" Via Claudio 21 – 80125 Napoli, Italy

E-mail: {stavallo,saldanto,mesposit,pescape,spromano}@unina.it

Abstract

In this paper we investigate how network management and routing can be appropriately combined to provide effective support to advanced network infrastructures. The work belongs to the research field dealing with network architectures in which resource management is achieved through a combination of Quality of Service mechanisms, Traffic Engineering and Monitoring.

The contribution is represented by the definition and implementation of a topology discovery module that can be used to derive dynamic information about current network topology and status, both in the intra-domain and the inter-domain scenario. Such a module is based on the exploitation of routing information coming from either the OSPF or the BGP routing protocols. SNMP is used as a means for gathering data stored inside routers' databases.

1. Introduction

Modern network architectures are highly demanding in terms of requirements imposed to the management plane. When Service Level Specifications (SLS [1]) are the key driver behind resource configuration, in fact, the need arises for a centralized approach to intra-domain network management, on one side, and inter-domain (i.e. among the different providers along the SLS value chain) synchronization, on the other. The main objective inside a single provider's domain is the optimization of the SLS acceptance. This often entails the adoption of Traffic Engineering techniques based on QoS constraints. The exploitation of such paradigm, though, requires that network topology, together with up-to-date information about current level of utilization of network links, is available at one single point acting as a centralized management entity for the domain. This situation is depicted in Figure 1, which shows the various modules that are needed in order to allow SLS-based management of a domain. It comes out from the picture that the topology discovery module is the main source of information for the Traffic Engineering engine.

When the SLS scope spans over multiple domains, inter-provider synchronization becomes of paramount importance to orchestrate the distributed solution of the QoS partition problem (i.e. to appropriately split an end-to-end problem into a series of edge-to-edge sub-problems). In this scenario, a mechanism is needed to let a provider identify its peer successor (i.e. the *egress* router) along the path which connects SLS source and destination nodes.

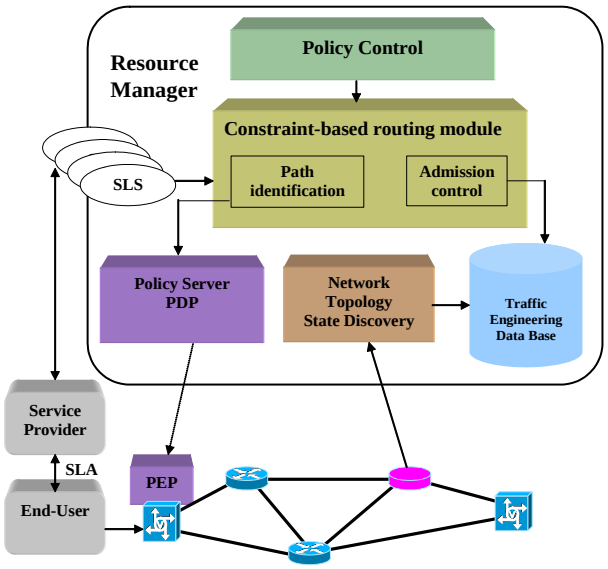


Figure 1. Overall Model.

In this paper we present a tool which has been conceived to solve the two issues mentioned above: i) topology discovery inside a single domain; ii) identification of the egress router towards the right inter-domain peer. Information retrieved by the topology discovery module is represented in XML format and then sent to the Resource Manager, which uses it to provide up-to-date information to the traffic engineer-

ing database.

The paper is organized in seven sections. Section 2 presents the main motivation of our work. Section 3 discusses related work and contrasts it with our approach. Section 4 describes the operation and structure of the Topology Discovery module, by focusing on its high level design and by highlighting its functionality. Section 5 and Section 6 illustrate, respectively, the actions performed in order to either get information about the egress point towards a specified destination or appropriately represent gathered data in XML format. Finally, section 7 summarizes results and discusses possible directions for future work.

2. Topology Discovery: motivations and discussion

By the term *topology discovery* we mean a process that identifies all the entities involved in the delivery of services to the users. There is an increasing number of Internet applications that attempt to optimize their network communication by considering the network topology. Hence, the need arises for algorithms and techniques that can identify network entities based on little or no information about their role and behavior and the way they interact. Typically, the output of a topology discovery process is represented by the list of available hosts, routers and subnets [2]. When using topology discovery for network management, the employed mechanisms must be as general as possible, so that they can be exploited for different types of networks and applications. Information gathered by a topology discovery module can prove useful in a number of situations, from 'faults isolation', to 'performance analysis', 'network planning' and 'services positioning'.

Many approaches to topology discovery have been proposed in the literature:

- *Passive Methodology*: relying on the use of SNMP (Simple Network Management Protocol) and DNS (Domain Name Server);
- *Active Methodology*: in this case there is a massive use of tools like 'ping' and 'traceroute';
- *Routing Based Methodology*: topology is derived by using the information of routing processes;
- *Hybrid Methodology*: the efficient combination of the previous methodologies.

The different approaches to topology discovery can be evaluated using four key parameters. In particular a topology discovery algorithm must be *efficient* (i.e. impose the least possible overhead on the network), *fast* (i.e. take the least possible time to complete the job), *complete* (i.e. discover the entire topology) and *accurate* (i.e. not make mistakes). The Passive Methodology is fast and reliable but it is not always usable. Indeed, SNMP is not universally deployed and also when present the community name is needed. At the opposite side, the Active Methodology is neither reliable nor fast, but it is always applicable (in all networks where ping and traceroute traffic is accepted). The Routing Based Methodology depends on the routing protocol but it is fast and reliable. Finally, a hybrid approach gives the opportunity to merge the benefits of each of the previous methodologies. In general none of the above non-integrated approaches (i.e. either passive, or active, or routing-based) is absolutely the best: indeed, it is important to choose the most suitable one depending on the managed network. This consideration is pushing research in the topology discovery field towards an adaptive hybrid methodology able to use an approach which is tailored to the current network infrastructure status and configuration.

In this paper we propose a hybrid approach, by appropriately combining a routing-based technique with a passive, SNMP-based, information retrieval methodology.

3. Related Work

This section compares and contrasts our framework with some other proposals of significance. We first briefly discuss the work of other alternative frameworks and then we present the motivation at the base of our work. In [3] Keshav et al., stepping from the assumption that traditional topology discovery algorithms are based only on SNMP (which is not universally deployed), describe several heuristics and algorithms to discover both intra-domain and Internet backbone topology while making as few assumptions as possible about the network. The proposed architecture combines SNMP routing information, traceroute, measurements, and heuristics to determine network topology. The projects developed starting from this preliminary work and called, respectively, *Octopus* [4] and *Argus* [5], implement three algorithms for automatic topology discovery using the Perl language. One of them uses SNMP to lookup the ARP tables of routers. The other uses DNS zone transfer and traceroute. The third tries to guess some addresses and then uses traceroute to find the topology. Another characteristic of this implementation is the use of modified versions of ping and traceoute, which, according to

the authors, speeds up the execution time of the algorithm by a large factor. As far as active techniques, the *Mercator* project [6] has explored tools such as traceroute to group IP addresses by network topology in order to produce an Internet map. Although many modern routers no longer respond to ICMP packets (used by the traceroute program), they have achieved good results on the modern Internet. Skitter [7] has been developed by CAIDA [8] to combine traceroute and benchmark-based analysis. This tool uses traceroute to find the paths connecting two nodes and also to collect performance information from them. At a higher level of abstraction, there have been interesting works [9, 10, 11, 12] which leverage traceroute measurements and BGP routes to help infer AS-Level connectivity. Finally, SNMP-based algorithms for automatically discovering network layer topology are featured in many common network management tools, such as HP's OpenView [13] and IBM's Tivoli [14]. These tools, however, assume that SNMP is universally deployed. The details of their discovery algorithm are proprietary and not available to the authors of this work.

In this paper, we develop a novel and practical solution to the problem of discovering network topology in IP networks based on the OSPF (Open Short Path First) routing protocol. The practicality of our proposal stems from the fact that it relies solely on standard routing information routinely collected in the router Management Information Bases (MIBs). In this work we will concentrate on the discovery of IP networks (all devices are IP addressable). We believe that our proposal is interesting when compared with other similar approaches. To our knowledge there are no other works about topology discovery based on a combination of SNMP and routing protocols. By using our approach several goals are achieved:

- by combining SNMP and routing protocol information we accelerate the process of manipulating data retrieved from the MIBs;
- by using SNMP in a single AS (without imposing that an SNMP agent is available on each and every router of the domain), we can easily access network devices;
- our solution effectively combines inter-domain and intra-domain routing information;
- our approach can be easily integrated with a more complete hybrid methodology.

4. The "Topology Discovery" module

In this section we will explain how we derive information about the topology of an Autonomous System in which OSPF has been chosen as the preferred Interior Gateway Protocol, also in a hierarchical configuration. By using the SNMP protocol, we can retrieve information from the Management Information Bases (MIBs) available at each router of the network, with special regard to OSPF Link State Advertisements (LSAs – Figure 2)).

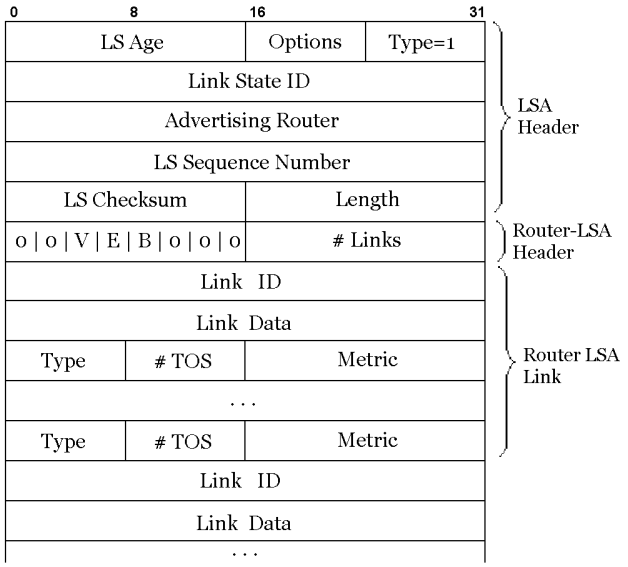


Figure 2. Format of a Link State Advertisement.

The topology discovery algorithm we designed can be synthesized in the following steps:

1. opening of an SNMP session with a randomly chosen router of the network;
2. in case of intra-domain hierarchical organization of routing, identification of a router belonging to the backbone area (i.e. area 0);
3. execution of the topology discovery algorithm in each and every area of the AS;
4. gathering of information about available network interfaces and associated link capacities.

In the following subsections we will investigate each of the above points in further detail.

4.1. Opening an SNMP session

Provided that all the routers in the AS support the OSPF protocol and that an SNMP agent is running on at least one of them (actually, SNMP must be active on at least one router per routing area), the first step is to open an SNMP session with one such router in order to gain access to the OSPF Management Information Base (O.I.D. 1.3.6.1.2.1.14, see Figure 3).

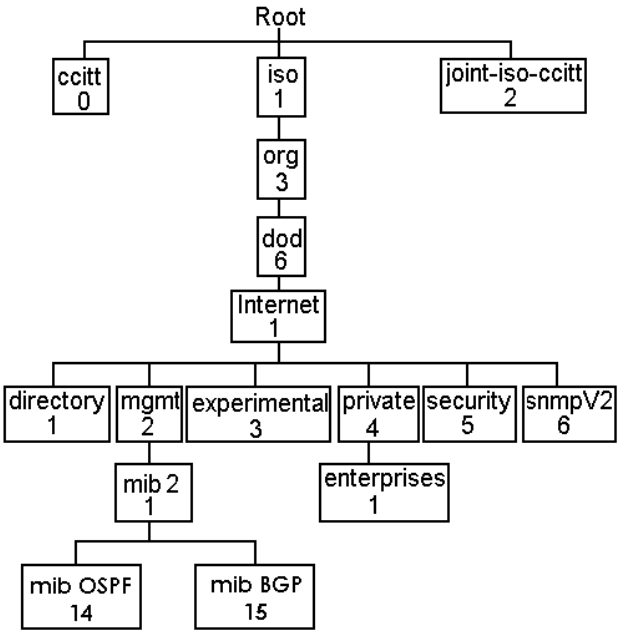


Figure 3. The OSPF and BGP MIBs.

Among the data contained in this part of the MIB, we can find detailed information about the local OSPF process, besides a copy of the router's LSAs. By analyzing this information, it is easy to determine the OSPF area which the router belongs to, and the identity of the designated Area Border Router (ABR).

4.2. Finding the Backbone Area

In case the first router we chose as a peer in the SNMP session belongs to a region other than the backbone area, we need to move towards this last area in order to obtain information about the overall organization of routers inside the AS. As illustrated in Figure 4, starting from a generic router belonging to area 1, we determine the identity of the local available border routers. Among such routers, we find one which is also attached to area 0: to this end, we iterate the search process, i.e. we send SNMP queries to the SNMP agents running on the ABRs in such a way as to determine whether or not they also belong to the backbone area. The advantage of gaining access to the backbone area resides in the possibility to derive information about the overall organization of the AS. In fact, by appropriately querying the MIBs of the backbone area's ABRs we can easily collect information about each area of the AS, thus reconstructing the general topology of the network.

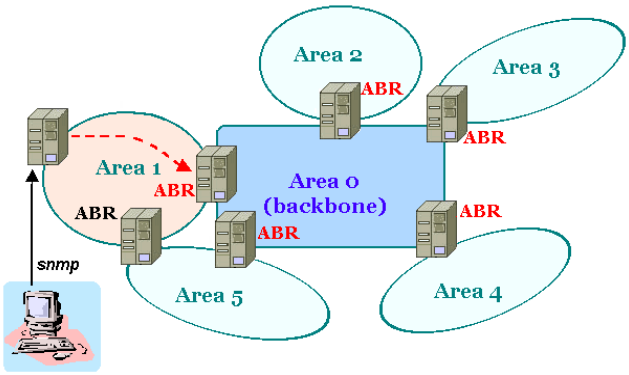


Figure 4. Finding the backbone area.

4.3. Computing Topology inside an Area

By sending requests to the SNMP agent of a router belonging to a specific area of the AS, we can gather information contained inside both router LSAs and network LSAs (as defined in the OSPF MIB). Data carried inside these structures, when appropriately combined, enable us to determine the topology of the area under investigation.

In the following we will dig into some of the algorithm details, which are summarized in Figure 5.

First, for each Network LSA, we find the associated Link State ID and Attached Routers IDs. Then, among the Router LSAs we look for the one containing the ID of the first attached router in the Advertising Router field. Finally, among the links contained inside the selected Router LSA, we identify the one whose Link State ID matches the corresponding ID extracted from the Network LSA. In this way, we can access the Link Data field, which contains the IP address of the router's interface belonging to that link. With the same approach, we also detect the IP address of the peering router's interface on the link. By iteratively applying the above procedure to each Network LSA, we eventually gain a comprehensive knowledge of the topology. In case of hierarchical routing, the algorithm presented is executed on the various area border routers identified. Each area topology is stored inside an ad hoc defined structure which concurs to

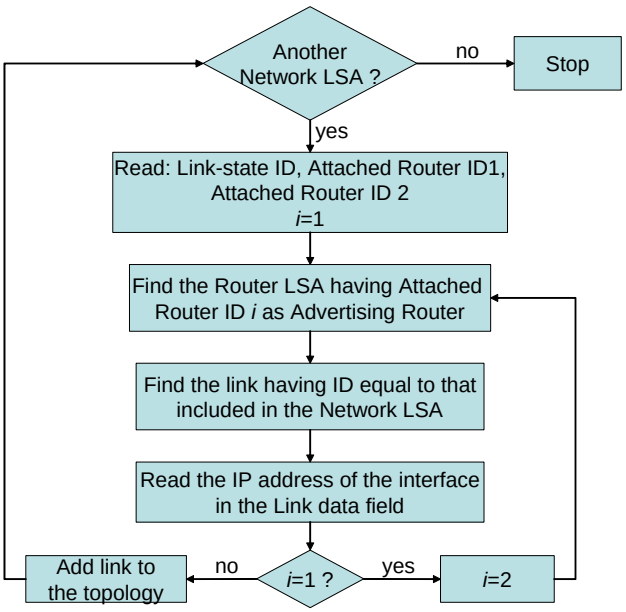


Figure 5. The topology discovery algorithm.

the definition of the overall image of the Autonomous System.

Figure 6 provides an example of the output produced by our algorithm when applied to a sample topology consisting of three areas (one backbone and two stub areas). As it comes out from the picture, the algorithm also determines information about the interfaces available at each router. In next subsection we explain how such information is computed.

4.4. Finding Routers Interfaces

As in the previous phase, the first step consists in reaching a router belonging to area 0. Afterwards, by sequentially contacting the SNMP agents running on the various area border routers identified, we can build the complete list of network devices belonging to the AS. For each such node, a new SNMP session is started in order to retrieve information contained in the node's MIB: IP addresses, interface identifiers and interface capacities.

5. The "Get-egress" module

This module is mainly built around the BGP-4 Management Information Base (Figure 3), with special regard to the sub-tree 15.6.1.6, which contains, for each possible destination, the so-called *Next-Hop* attribute.

This attribute actually indicates the IP address of the interface belonging to the first external router (i.e. the first router in a peering AS) towards a specified destination (see Figure 7). The main operations required in order to accomplish the above task are the following:

1. computation of the routes contained in the database of the BGP peer contacted (i.e. the AS ingress point);
2. search of the required destination among the available BGP routes;
3. computation of the AS egress point towards the destination.

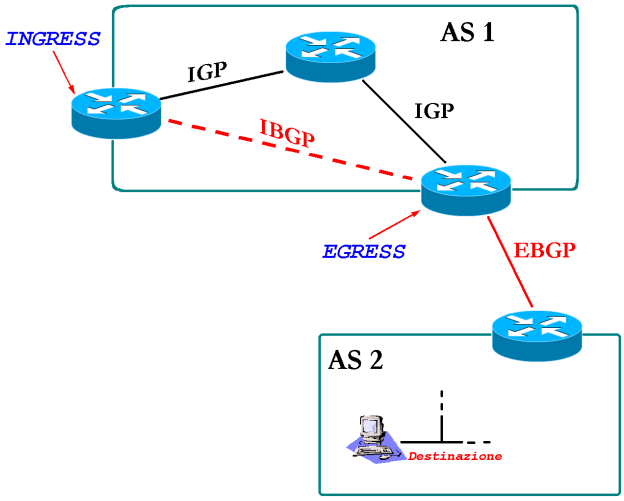
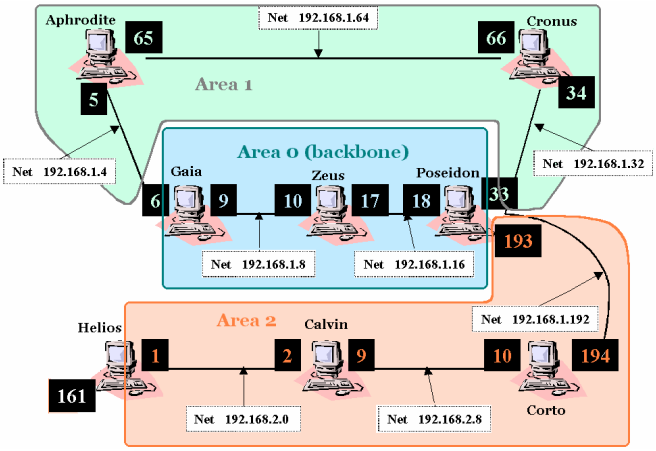


Figure 7. Getting information about the egress point.

6. The "Topology Sender" Module

This module is in charge of converting to XML format the network topology information acquired by means of the topology discovery module. Once done with this task, the XML data obtained are sent to the Resource Manager in order to provide it with up-to-date information about current network status and organization. Interaction with the manager can take place either via standard socket communication or by means of more advanced solutions like, for example, the Simple Object Access Protocol (SOAP).

Figure 8 illustrates the structure of the schema files we adopted for the description of topology information; Figure 9, on the other hand, describes how information reported in Figure 6 would be represented in XML format.




```
------------NODE'S LIST----------

 Nodo # 1
 Node-ID : 192.168.2.9
-------Interfaces' List--------

 Interface-ID : 4
 Interface-address : 192.168.2.9
 Interface-bandwidth : 10000000

 Interface-ID : 3
 Interface-address : 192.168.2.2
 Interface-bandwidth : 10000000

---------------------------
 Nodo # 2
   ......
   ......
```

```
---------- AS Topology ----------
 link # 1
SOURCE Router ID=192.168.1.34
SOURCE Interface ID=192.168.1.66
DESTINATION Router ID=192.168.1.65
DESTINATION Interface ID=192.168.1.65
 link # 2
SOURCE Router ID=192.168.1.34
SOURCE Interface ID=192.168.1.34
DESTINATION Router ID=192.168.1.33
DESTINATION Interface ID=192.168.1.33
 link # 3
SOURCE Router ID=192.168.1.9
SOURCE Interface ID=192.168.1.6
DESTINATION Router ID=192.168.1.65
DESTINATION Interface ID=192.168.1.5
 link # 4
   ......
   ......
```

Figure 6. A sample topology as computed by the algorithm.

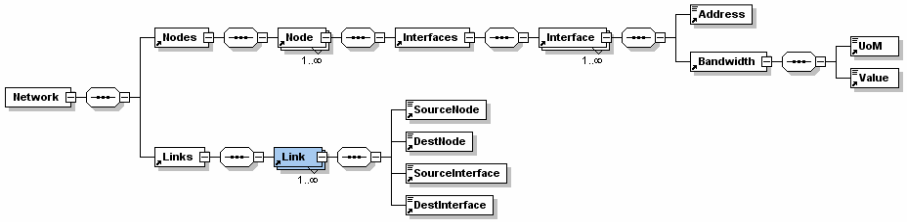


Figure 8. Structure of the XML files representing topology information.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Network id="1">
  - <Nodes>
    - <Node id="192.168.2.9">
      - <Interfaces>
        - <Interface id="4">
            <Address>192.168.2.9</Address>
          - <Bandwidth>
              <UoM>bps</UoM>
              <Value>10000000</Value>
            </Bandwidth>
          </Interface>
        + <Interface id="3">
        + <Interface id="1">
        + <Interface id="2">
        </Interfaces>
      </Node>
    + <Node id="143.225.229.161">
    + <Node id="192.168.1.65">
    </Nodes>
  - <Links>
    - <Link id="192.168.2.9">
        <SourceNode>192.168.2.9</SourceNode>
        <DestNode>143.225.229.161</DestNode>
        <SourceInterface>192.168.2.2</SourceInterface>
        <DestInterface>192.168.2.1</DestInterface>
      </Link>
    - <Link id="192.168.1.65">
        <SourceNode>192.168.1.65</SourceNode>
        <DestNode>143.225.229.161</DestNode>
        <SourceInterface>192.168.1.2</SourceInterface>
        <DestInterface>192.168.1.1</DestInterface>
      </Link>
    </Links>
</Network>
```

Figure 9. A sample XML file representing network topology.

7. Conclusions and Directions of Future Work

Automatic discovery of physical topology information plays a crucial role in enhancing the manageability of modern IP networks. Despite the importance of the problem, discovering network topology is an inherently difficult task. The network topology knowledge is significant for simulation and network management. It can also be used effectively for siting decisions, and as one of the inputs in a new class of topology-aware distributed systems. Since there are no standards, any algorithm developed to discover the topology can only use the basic IP primitives. In this paper, we have presented a novel, practical solution for discovering the topology of IP networks based on OSPF routing. We are currently in the process of optimizing our implementation and conducting more extensive experimental tests. Next step will be the inclusion of the proposed solution in a more general framework named NeToDi (Network Topology Discovery), which represents an adaptive hybrid solution to network topology discovery made by an efficient composition of active, passive and routing protocol based methodologies.

Acknowledgements

Research outlined in this work has been carried out in the framework of the European project INTERMON (Advanced architecture for INTER-domain quality of service MONitoring, modelling and visualisation), IST-2001-34123.

References

[1] D. Goderis et al., *A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks*, IEEE Communications Magazine, May 2001, Vol. 39, Issue 5, Page(s): 80–88.

[2] C. Gkantsidis, E. Zegura, *Experiment and learn to discover Network Topology*, October 1999 Georgia Tech.

[3] R. Siamwalla, R. Sharma, S. Keshav, *Discovering Internet Topology*, `http://www.cs.cornell.edu/skeshav/papers/-discovery.pdf`, 1999.

[4] Cornell Network Research Group, *Project Octopus*, `http://www.cs.cornell.edu/cnrg/topology aware/topology`

[5] Cornell Network Research Group, *Project Argus, Network topology discovery, monitoring, history, and visualization*, http://www.cs.cornell.edu/boom/projects/-Network Topology/topology.htm.

[6] R. Govindan and H. Tangmunarunkit, *Heuristics for internet map discovery*, IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000.

[7] CAIDA, *Skitter*, http://www.caida.org/TOOLS/-measurement/skitter/

[8] CAIDA home page, http://www.caida.org

[9] R. Govindan and A. Reddy, *An Analysis of Internet Inter-Domain Routing and Route Stability*, Proceedings of IEEE INFOCOM '97, April 1997.

[10] H. W. Braun and K. C. Claffy, *Global ISP interconnectivity by AS number*, http://moat.nlanr.net/AS/

[11] H. Chang, S. Jamin, and W. Willinger, *Inferring AS-level Internet Topology from Router-level Traceroutes*, Proceedings of SPIE ITCom '01, Scalability and Traffic Control in IP Networks, August 2001.

[12] H. Chang, S. Jamin, and W. Willinger, *On Inferring AS-Level Connectivity from BGP Routing Tables*, available at http://topology.eecs.umich.edu/archive/-inferbgp.ps.

[13] http://www.openview.hp.com

[14] http://www.tivoli.com