Operational Layer 3 Topology

Akshay Adhikari Lorraine Denby Jean Meloche
Balaji Rao
Avaya Labs Research, Basking Ridge, NJ
{akshay, ld, jmeloche, brao}@research.avayalabs.com

August 12, 2003

Abstract

Knowledge of the currently active network topology is useful for a number of tasks that include identifying sources of poor application service quality and proactive network fault detection. We present the concept of the *operational* topology of an Internet Protocol (IP) network and a technique for discovering it. This lightweight technique does not make use of any proprietary information and is thus highly suitable for use in a multi-vendor environment. We also discuss uses of the operational topology, and present some operational topology metrics and visualization.

1 Introduction

Knowledge of the currently active network topology is critical for various network management tasks. In the context of the IP layer, the network topology is a set of routers and the (directed) links or edges connecting them. A conventional topology is derived by active probing and/or information contained in Simple Network Management Protocol (SNMP) Management Information Bases(MIBs). It lacks two features: Firstly, it is a theoretical one. The only information it contains is a set of nodes and edges. As will be explained later, it is sometimes impossible to identify the existence of multiple paths between hosts, as may be the case if the network employs load balancing, policy routing or a dynamic routing protocol such as Open Shortest Path First(OSPF). Even if the topology indicates multiple *possible* paths, we do not know which of those paths are actually used in getting from a source to the destination. Secondly, it is static. It does not change over time to reflect path usage information - which paths are used, what the pattern of their usage is, and how often paths are used. Such information is extremely valuable for network management tasks, and so it is useful to define a different kind of topology that addresses these issues.

In this paper, we discuss a special kind of IP layer topology called the *operational* topology, and distinguish it from a conventional topology in two aspects. Firstly, by operational topology, we mean the nodes, edges *and paths* of a network that are actually used in communication between a set of endpoints (hosts) that use the network. It is dynamic, constantly evolving and discovering new nodes, edges and paths as they are exercised. Note that we know exactly which paths are used between a source-destination pair, not just the paths possible. Secondly, the operational topology includes more information than just nodes, edges and paths: it includes the usage pattern of paths between endpoints.

The term “operational topology” has been discussed in [5], in the context of Ethernet networks. The authors define it to be the currently active topology of a switched network (at the time of discovery), as determined by the state of the spanning tree protocol running on the switching elements. This resolves ambiguities resulting from redundant switching connections. Our contributions are twofold. Firstly, we define the concept of the operational topology for the network layer. Here, it is different from a conventional network layer (layer 3) topology in its ability to be dynamic and contain path information. Secondly, we describe a technique to generate it, that that does not use any vendor dependent information.

The rest of this paper is organized as follows. In Section 2, we describe some previous work on topology discovery and its limitations. Section 3 describes some additional features of the operational topology, and Section 4, some uses. In Section 5, we present a method to generate an operational topology, and in Section 6, its limitations. In Section 7, we present some operational topology metrics and visualization techniques. Finally, we conclude with some description of future work in Section 8. In the rest of the paper, we intend the word “topology” to mean operational topology, unless otherwise specified.

2 Previous work

As mentioned before, a topology consists of a set of nodes and directed edges connecting them. Numerous approaches have been described in the literature for generating network topology. Broadly, they are based on two methods:

(a) Simple network management protocol(SNMP): These methods rely on routing table information collected from routers from appropriate SNMP Management Information Bases (MIBs) [6][1]. These methods suffer from two drawbacks: Firstly, these MIBs do not contain information about multiple paths to a destination (as would exist if load balancing or policy routing is configured on a router). Such information exists only in proprietary MIBs, or is not accessible at all through SNMP. Thus, SNMP based methods may not be sufficient in a multi-vendor environment, and may not be capable of identifying multiple paths and their usage patterns. Secondly, SNMP access is turned off by many administrators for security reasons [6], and enabling access can be a very time consuming task, since it requires manual intervention.

(b) Active probing: Some methods use traceroute like probes in conjunction with ping. The most notable of these is the Skitter project at CAIDA [3]. They rely on the fact that if a computer is present at an address, it will generally respond to ping, and that routers do not forward packets whose Time To Live (TTL) field is 1. Instead, they send back an ICMP time exceed error message to the source of the probe, revealing their IP address. These probes are sent incrementally to a destination, each time with an increasing value of TTL, thus discovering all the routers along the path from the source to the destination. However, these methods also fail in the presence of load balancing. This is because successive probes may follow different paths, thus possibly resulting in edges that are not physically present.

Most tools such as HP's Openview, the Dartmouth Intermapper, and work described in [6] use several complementary sources of information including SNMP, active probing and Domain Name Service(DNS) zone transfers to get as much information about topology as possible. However, all the tools fail to account for multiple paths for the reasons mentioned above, and do not include path usage information.

3 Characteristics

We have mentioned two characteristics of an operational topology, namely that it contains paths in addition to nodes and edges, and also keeps usage information about paths. We wish to elaborate on some related aspects, that further distinguish an operational topology from a conventional one.

An operational topology shows us a view of the network that is used by a set of endpoints - only those nodes, edges and paths comprise the topology, that are used in communication between the endpoints. Therefore, our view of the network depends on where the endpoints are placed. On the other hand, the conventional topology is not concerned with usage of the network, and hence does not require a system of endpoints. All that is needed is to probe a set of IP addresses to find out if routers or hosts exist there, and then find their characteristics. Therefore, the scope of the network to be discovered can be arbitrarily large. A conventional topology using active probing and/or SNMP would require only a few sources of probes. To create an operational map of the Internet, on the other hand, would require a large number of endpoints placed at locations that would ensure adequate coverage of paths. Thus, both the number of endpoints required, as well as the need to control their placement are disadvantages of the operational topology. Therefore, it is more practicable to generate the operational topology of a small and controlled network like an enterprise network, rather than that of the public Internet.

Note one more unique aspect of the operational topology that is implicit in its definition. Conventional methods of discovery can only yield a snapshot of the network at the time of discovery. The results may be invalid a few minutes later as routing protocols update routing tables according to new information. On the other hand, the operational topology constantly updates information as paths between endpoints are exercised. From that point of view, neither the process of topology discovery, nor the topology itself are ever “complete”. We obtain the current state of the topology at the time we observe it. Thus, the speed of discovery cannot be defined. We will however, present some closely related metrics for an operational topology in a later section.

4 Uses

One of the most important uses of the topology is to aid in several network management and troubleshooting tasks, of which we list three here.

- Load Balancing: Knowing all the possible nodes, edges and paths that are actually used between endpoint pairs, and the pattern of their usage informs us about the efficacy of any load balancing mechanisms employed in the network. Also, for some applications like Voice over IP (VoIP), load balancing is not recommended, since it may lead to out of order delivery of packets. Hence we may want to observe if VoIP packets are routed through load balanced paths.

- Policy based routing: Sometimes routers are made to employ different routes for different kinds of traffic classified by source/destination addresses, Type of Service (TOS) settings, layer 4 protocol (TCP/UDP), etc. For example, in a converged network, delay sensitive traffic like VoIP may be sent via a high speed path, while traffic from less demanding applications may be routed differently. This mechanism is called policy based routing [2]. Thus, instead of a single conventional topology, there would exist multiple operational topologies, one for each class of traffic, and we could verify correct functioning of policy routing mechanisms.

- Routing problems: The operational topology can reveal unusual (and probably unwanted) routing patterns: for example, packets from the east coast destined to the east coast of the US being routed through a hop on the west coast.

Another very important use is in the closely related fields of fault detection and Quality of Service (QoS) monitoring/testing. In fact, this is the context in which we generate the topology. Information about how packets flow between endpoints in the network, combined with measurements of end-to-end QoS for these packets (such as one-way/round trip delay, loss and jitter) facilitate easy isolation of network problems. Measurements for the analysis of traffic flows such as most/least frequently used edges and edges that are always used in conjunction with each other are also useful for proactive fault management.

Of course, like a conventional topology, the operational topology can be used for other purposes like driving simulations and topology-aware algorithms.

5 Methodology

In this section, we describe our methodology to generate the topology collectively from a set of *endpoints*. Endpoints are network nodes that we are capable of controlling for the purpose of generating IP traffic to other endpoints or routers, and storing and reporting results obtained from this traffic. From that point of view, even routers may be endpoints; usually, however, endpoints will be end hosts on the network that run the topology generation algorithms, and from now on, we use the terms “endpoint” and “host” interchangeably. The nodes, edges and paths stored on the collective of endpoints comprises the topology, and can be fetched and analyzed by other entities.

5.1 Nodes, Edges and Paths: IP record route

As opposed to getting information from SNMP MIBs or from traceroute probes, our method relies on generating some kind of IP communication (such as User Datagram Protocol (UDP)) between pairs of endpoints. One endpoint A sends a message to the other called B. Each IP packet exchanged has the RECORD_ROUTE option enabled in its header. This tells each router along the path from A to B to write the address of its outgoing interface in the RECORD_ROUTE space of the IP header. Thus, at endpoint B, we get a sequence of IP addresses that defines the path the packet from A to B took, and vice versa. Exchanging such communications between several pairs of endpoints gives us a set of nodes, edges and paths between them.

The endpoints perpetually exchange messages, resulting in the addition of new nodes, edges and paths to the topology as they are exercised. Furthermore, the packets exchanged can have different characteristics such as special source/destination ports or TOS settings, enabling us to verify and troubleshoot the dynamic routing mechanisms as discussed in Section 3. Note that all that is needed is to exchange IP packets with the RECORD_ROUTE option enabled, and hence very small packet sizes are sufficient. Hence, the discovery traffic level can be kept minimal.

5.2 Node Merging: UDP probes

Every router, by definition, has at least two interfaces. When the record route option is processed, the router appends the address of its *outgoing* interface to the record route space. When multiple probes are sent to destinations that are reachable via different interfaces of a router, several IP address are obtained that belong to the same router. The process of merging identifies IP addresses that belong to the same router and groups them into an **equivalence class**. Thus, the result of merging nodes obtained from probes is a set of equivalence classes, one for each router in the topology.

Merging is accomplished following the method desribed in [3]. When a UDP packet is sent to an unused port on a router, an ICMP PORT UNREACHABLE error message is usually generated from the IP address of the router that is on the unicast route toward the sender of the UDP packet. If we probe one IP address and get back the error message from another IP address, we can conclude that both of them belong to the same router. These two addresses can be grouped into a class.

5.3 Cloud devices: TTL based identification

Often, during a topology determination using the RECORD_ROUTE option, it might not be possible to obtain all the router addresses along the path. There could be two reasons for this. First, IPv4 has a limitation of 9 hops because of a fixed header size. Any router that is in the path of the packet beyond the 9-hop limit cannot identify itself in the IP packet header for the RECORD_ROUTE option. Second, it is also possible that routers are configured not to add their address in the packet (this is sometimes done by network administrators for security purposes). In both cases, we refer to the routers as "Silent Routers" or "Cloud Devices".

It is possible to determine the existence and location of cloud devices by using the Time to Live (TTL) field of the IP header [4]. Whenever an IP packet is transmitted on a network, a TTL, which is a non-negative value ($<$ 255), is set on the packet. Every router must decrement the TTL by 1 as it forwards the packet. Also, routers discard packets that have a TTL of 0. This ensures that the packet has a finite lifetime on the network and does not bounce back and forth. Thus, the TTL value of the packet when it reaches the destination reveals the exact number of hops from the source to the destination. The difference between this value and the number of hops recorded in the IP header of the packet gives us the number of cloud devices. Note that this difference must be positive.

We now describe the method to determine the location of cloud devices with an example. Let us consider a simple network from Source to Destination along routers whose interfaces are A, B, C, D, E, F, G, H, I, J and L as shown in Figure 1.

Let us consider the following cases: Case 1: All routers between Source and Destination except J and L record their address. They are unable to do so since there is no more space in the IP header.

Case 2: The router "E" is configured not to record the route, and so does not record its address. L does not record its address since the 9 hop limit has been exceeded.

In both cases, we know that there are two cloud devices. To determine the position of the cloud devices, a series of ICMP probes with increasing TTLs are sent to every observed node along the recorded route in succession. If a TTL matched probe to any particular address does not result in a reply, or results in an ICMP TTL exceeded message, we can position a cloud at that particular TTL indicated hop.

In case 1, we send a probe to A with TTL=1, and we get a reply, then a probe to B with TTL=2, we get a reply. We continue to send probes with increasing TTLs until TTL=9. If all these resulted in a reply, we place the clouds at the end of the list. The exact number of clouds would be the difference in TTL indicated hop count and the list of IP addresses obtained in the IP header (2 in this case).

In case 2, we get a recorded address list of A, B, C, D, F, G, H, I, J. We use the above list as a signature and start out sending out probes. Probes to A, B, C and D with a TTL of 1,2,3,4 respectively yield a reply. Now, probe to F with TTL=5 will result in an ICMP TTL exceeded from E (It could also result in no reply at all). This happens as the TTL needed to reach E is actually 6. This indicates the location of the first cloud after D. Continuing along the path, a probe to the same address F with TTL=6 yields a reply, probe to G with TTL=7 yields a reply. Probes to H with TTL=8, to I with TTL=9 and to J with TTL=10 yield replies. Since we have determined the first 9 hops, we position any remaining cloud devices at the end of the path after J. Here, there would be one cloud device after J, representing L.

The same algorithm can be used in a network that employs load balancing. The additional step that need to be performed in a load balanced network would be to use the original recorded route as a signature, and to record the route of the probes. If the probe's route is a substring of the signature (indicating that the probe followed the same path segment as the original packet), we can use the paths to identify the cloud device's location.

6 Limitations

In this section, we describe some limitations of our approach, and also some workarounds.

1. 9 hop limit: In the IPv4 header, there is space for recording at most 9 IP addresses (this includes the endpoint addresses as well). Consequently, paths longer than 9 hops remain incomplete - we can only label the rest of the path as a series of cloud devices. By suitable modification of the endpoint operating system, we have increased this limit by 2, by preventing the endpoints from entering their own address into the record route space. However, now any paths longer than 11 hops remain incomplete. This problem can be solved by deploying endpoints at intermediate points in the network. Also, with IPv6, the headers are extensible and we should be able to record larger paths.

2. Non-cooperating/Non-compliant routers: Routers can be configured to ignore the record route option, for which we have already discussed a work around. In our experiments, we have also found some equipment that does not follow the standard described in [4] for responding to UDP messages to unused ports. This causes the merging algorithm to fail. Thus, other methods/heuristics of node merging have to be

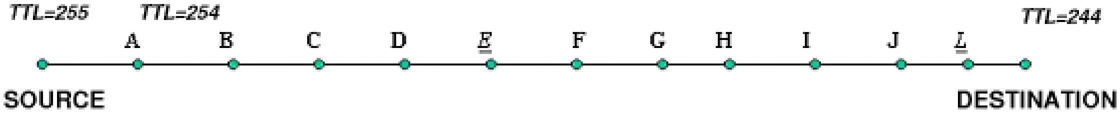

Figure 1: TTL based Cloud extraction.

investigated. These include using SNMP to find all the IP addresses of a router, using DNS zone transfers [6]. Another method that we are in the process of investigating is called “time based merging”. The idea is that if several IP addresses belong to the same router, then ping response times from them should all be the same. All known addresses can be probed from several endpoints and a test of statistical significance conducted to merge nodes.

7 Measurements

In this section, we use the example of a topology discovery we conducted recently on a production network to highlight some interesting metrics. Our experiment involved a set of 11 endpoints. The rate of traffic injected into the network was approximately 10 UDP packets per second. We now discuss some metrics:

1. Number of nodes, edges and paths: At the time we inspected it, the operational topology between the 11 endpoints contained 27 nodes and 141 edges (See figure 2). Note that even though the number of nodes and edges in a topology may be small, the number of paths may be combinatorially large. In a system of n endpoints, the number of distinct paths is at least $2 * nC2$. The number of combinations can be larger depending on node degree. In this example, there were 709 paths.

2. Rate of growth of the topology: We have discussed before that the speed of discovery cannot be defined for an operational topology. However, it is useful to know how much time it takes to reach a state where all possible nodes, edges and paths have been discovered; this is a measure of how dynamic the network is.

   If a network does not employ load balancing, policy routing or dynamic routing, the topology is static, and all the nodes, edges and paths can be exercised and hence discovered quickly. However, if any of these features are present, it is possible that rarely used nodes, edges or paths may be discovered only after the system has run for a very long period of time.

   As an example, consider figure 3. This is a plot of the number of edges and paths known to the system of endpoints, plotted against the time at which they were first used (discovered). Almost all edges had been discovered in the first few hours of operation. However, the last few *paths* were discovered after the system had been left running overnight, another indication that a large number of paths can be derived from a small set of edges.

   Of course, the time required to learn paths depends on the rate of exchange of traffic between the endpoints. The faster the communication, the faster is the rate of discovery of new nodes/edges. Therefore, figure 3 is by no means representative, however, it does give us an insight into the dynamics of the system.

3. Path usage: Other interesting metrics relate to node/edge/path usage. For example, we may be interested in knowing the most/least frequently used edges/paths and the last time a path was used. These metrics, like the rate of growth of the topology, give us an insight into the dynamics of the network. Such metrics can also be applied to a subset of the traffic flowing in the network. Various filters may be applied to obtain the subset, such as traffic of a particular Type of Service(TOS) setting, traffic for which the end-to-end delay is smaller than or larger than a certain threshold, etc. These measurements are easy to visualize by different colors or thickness for drawing nodes and edges. As an example, Figure 4 shows the topology of our network, with edges color coded linearly from white to red according to usage count.

8 Conclusions and Future Work

In this paper, we have described a novel concept called an operational layer 3 topology, that includes, in addition to nodes and edges of a conventional topology, the actual paths used in communication between endpoints using the network, and their usage pattern. We described a method to generate the operational layer 3 topology for an IP network, that relies on the IP RECORD _ROUTE option, and some features of the IP/ICMP protocol. We also described some uses, metrics and visualization for the operational topology.

Future work will be directed along several fronts. Like other methods of topology discovery, ours is vulnerable to router configurations that prevent gathering of relevant information (the record route option in our case). In our experience, the record route option is less likely to be disabled than SNMP access. However, more experience is required to determine if this is actually the case. Like other conventional topology discovery mechanisms, we would like to exploit several complementary sources of information for node merging, like DNS, SNMP and ping round trip times (while avoiding,as far as possible, reliance on proprietary information). Metrics for operational topologies is a separate area of research by itself. Finally, we would also like to investigate techniques for generating layer 2 operational topologies.

References

[1] BREITBART, Y., GAROFALAKIS, M. N., MARTIN, C., RASTOGI, R., SESHADRI, S., AND SILBERSCHATZ, A. Topology discovery in heterogeneous IP networks. In *INFOCOM (1)* (2000), pp. 265–274.

[2] CISCO. Quality of Service (QoS) Fact Sheet.

[3] HU, B., A DANIEL, AND DAVID, P. Topology discovery by active probing, 2002.

[4] INFORMATION SCIENCES INSTITUTE, U. O. S. C. Internet Protocol, DARPA Internet Program Protocol Specification. Request for Comments 791, Internet Engineering Task Force, Sept. 1981.

[5] MEISS, M. R., AND WALLACE, S. S. Standards-based Discovery of Switched Ethernet Topology.

[6] SIAMWALLA, R., SHARMA, R., AND KESHAV, S. Discovering Internet Topology. In *INFOCOM* (1999).

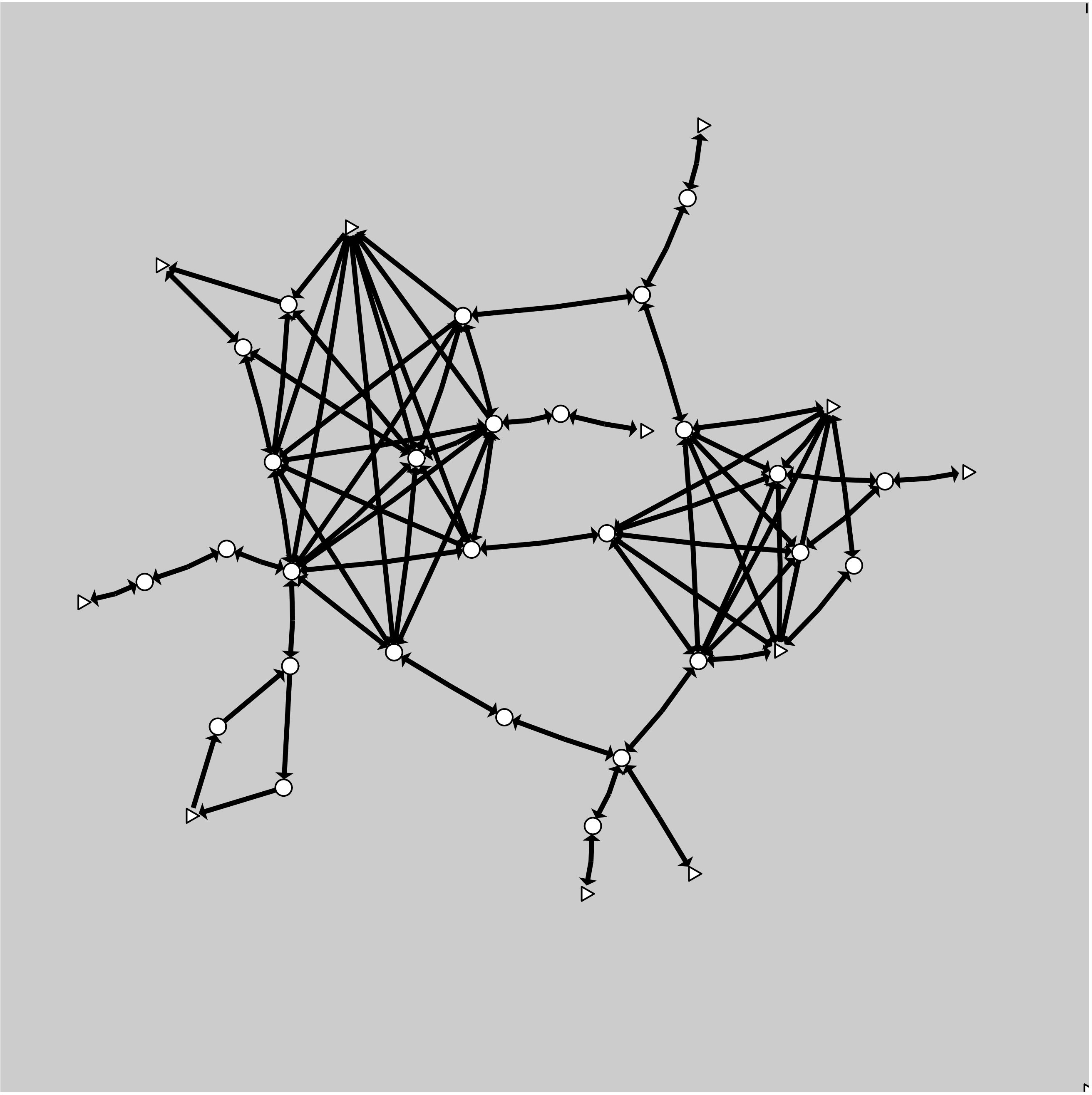

Figure 2: A snapshot of the operational topology

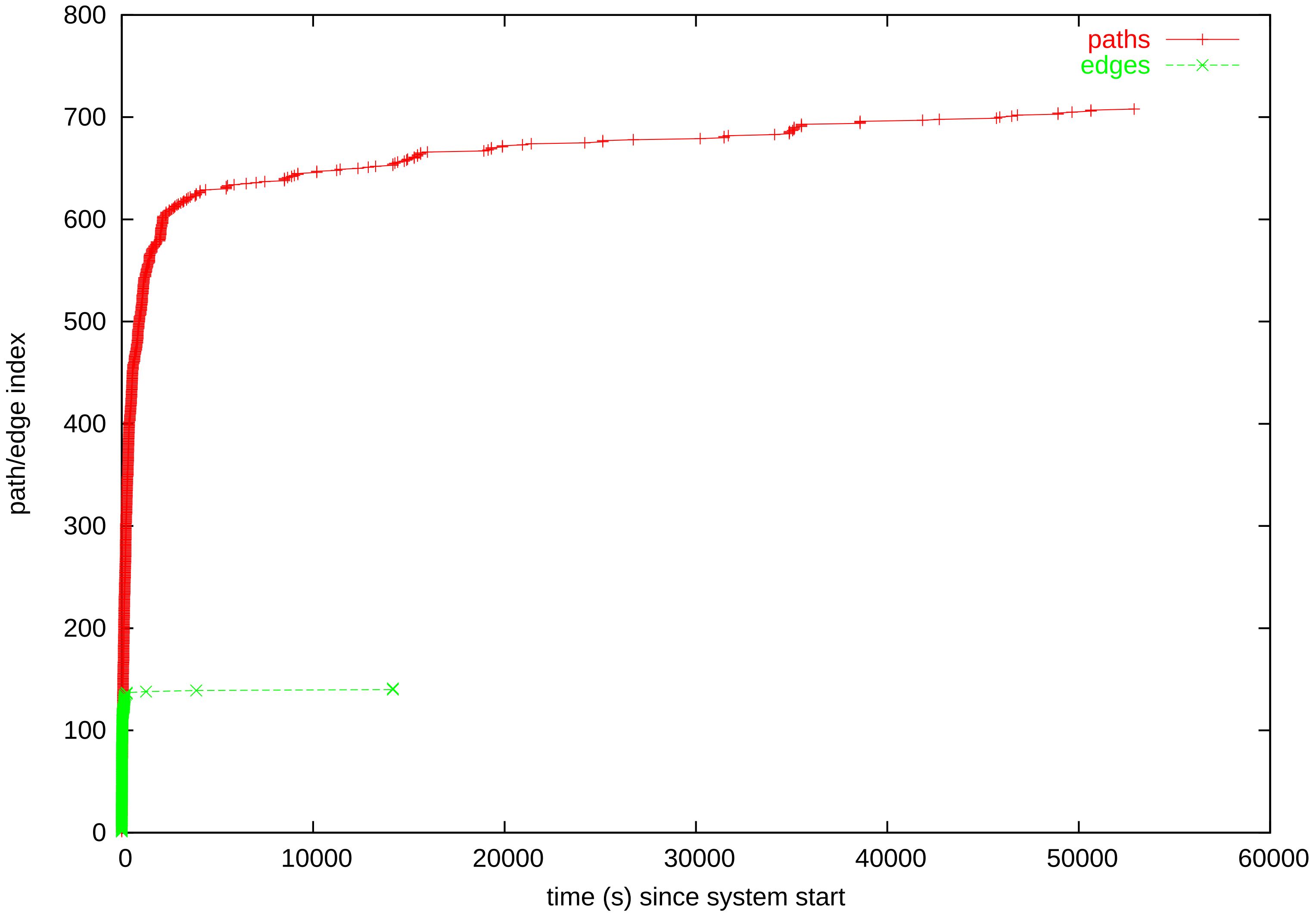

Figure 3: Growth of the operational topology

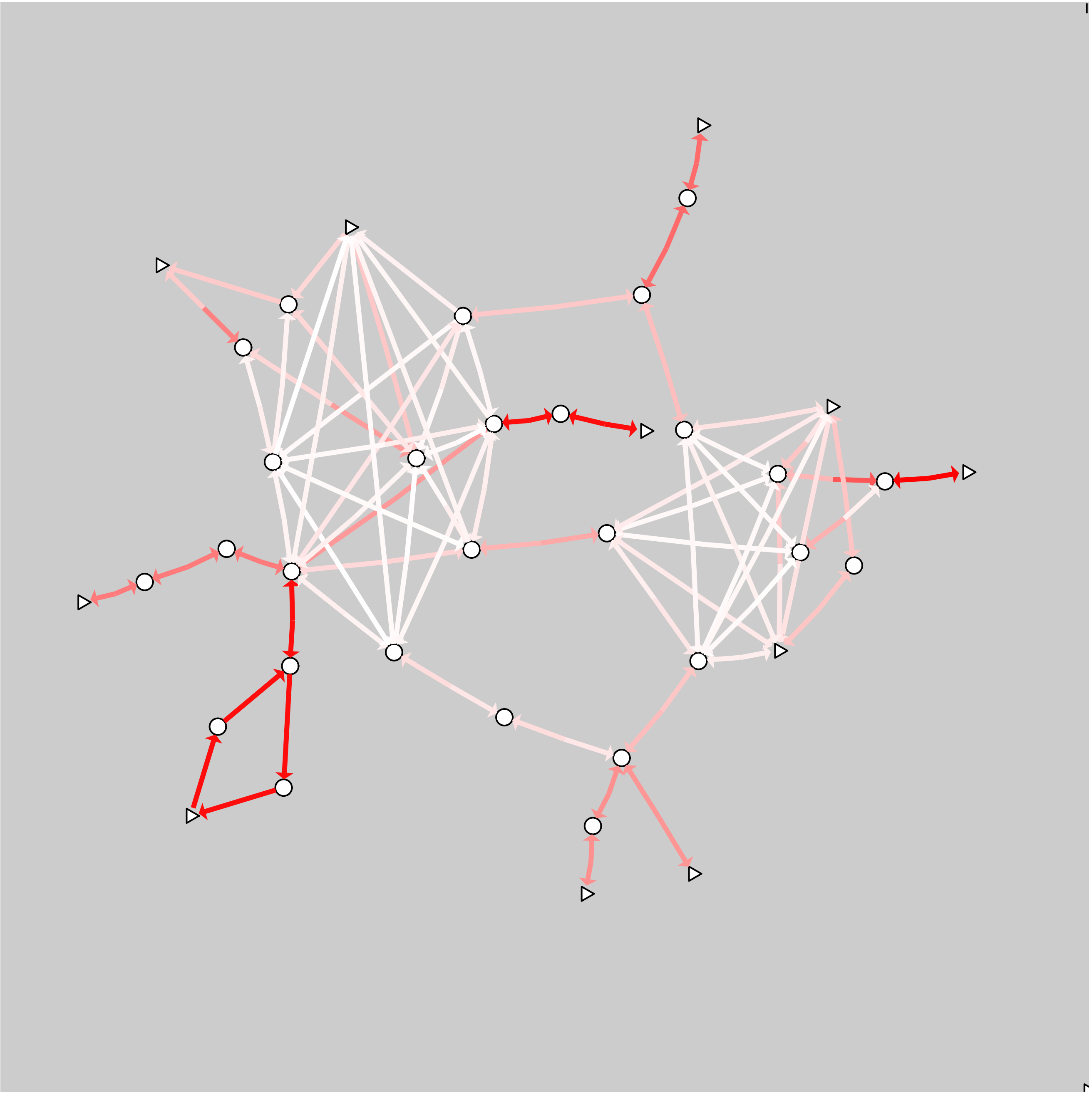

Figure 4: Edges color-coded to show usage count