

# Linking Services to the Network through Service Path Analysis.

Roger Castillo CTO AlterPoint, Inc., Stephen Bayer Sr. Researcher AlterPoint Inc.  
rcastillo@alterpoint.com sbayer@alterpoint.com

## ABSTRACT

Service management has become a key focus area for IT management as IT organization have sought higher levels of abstraction to manage ever more complicated infrastructures. Existing management approaches are still focused on network device level management and new management abstractions are required deal with this service management complexity.

No complete picture has existed detailing ‘how’ specifically applications and users are connected via the network. Present management solutions have attempted to approximate this picture through user input, application mapping/discovery and application traffic analysis with very limited success. By computing path analysis per application class and identifying attachment points for various hosts participating in those applications, a service path analysis view of the network is possible. This path capture is accomplished through static analysis of the network configurations (routers/switches) as well as by analyzing packet filters that suggest whether a particular application is able to communicate to/from the network at a particular point. Through this analysis, an explicit linkage between network services and networking infrastructure is made possible.

This paper outlines:

- Algorithms used to perform the service path capture
- Discussion of potential applications for Service Path Analysis for Enterprise network management
- Potential implications for network and service management at large

By leveraging Service Path Analysis several new key management applications become possible including, service oriented security, diagnostics and change impact analysis. As an example for diagnostics, given a service path for an application, the specific infrastructure and related configuration can be precisely determined in order assist in finding root cause for outages.

Through Service Path Analysis a service context can be provided to network management allowing network management to become service aware.

## Keywords

Network management, service, reachability, service and control

## 1. Introduction

IP’s evolution over the past two decades has resulted in enterprise IT infrastructures that are fundamentally dependent on IP for delivery of all applications and IT services. The proliferation of applications and services running on a single IP platform can be directly linked to the increase in enterprise IT organizations’ reliance on IP technology to deliver services to their customers and increase the reach of the business while at the same time lowering costs.

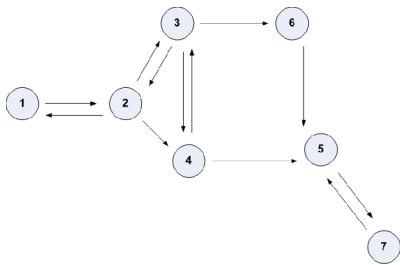
The ubiquity of mobile and wireless networks combined with the increasing convergence of what were once separate dedicated networks—data, audio, and video—has accelerated industry consolidation of many networks onto a single, converged “IP platform” or multi-service network. The current reactive, device level approach provides little leverage for dealing with the management complexity. Today’s networks are managed in a best efforts manner requiring significant overspend in manpower and hardware.

New management abstractions like service management and service oriented architecture for IT have been promoted as a way to better manage this growing complexity. Unfortunately, Service management approaches have been limited in IT in large part due to the complexity of understanding the way Applications are delivered to customers by the IT infrastructure. Existing approaches have ignored network configuration as means to develop this understanding.

Given these recent advancements in network configuration analysis and vendor independent configuration capture, a Service Path approach has been made possible the linking of services to networking infrastructure.

### Network Path Analysis

The network graph is represented as a directed graph or digraph to reflect that packets may be transmitted across links in either direction, the configuration of packet filters and routing policies may result in links along which packets may be transmitted in one direction, but not the other. To compute the deliverability of packets across the network we will thus need to use a directed graph, or “digraph”, in which each edge connecting nodes has an origin and destination node, and where the deliverability of packets in both directions between node  $u$  and node  $v$  will be represented by two separate edges, one from  $u$  to  $v$ , another from  $v$  to  $u$ . In figure 1 below we see a digraph with the direction of each edge explicitly annotated. The graph represents an IP network, with nodes representing devices and arrows indicating packet delivery direction. Note that packets are deliverable from device 6 to device 5 but are not deliverable from 5 to 6.



**Figure1: A digraph showing directed edges.**

The classic algorithm in graph theory for determining reachability is Warshall’s algorithm for computing the “transitive closure” of a digraph [1]. Given a graph,  $G$ , with vertices,  $V$ , and edges,  $E$ , the transitive closure of  $G$  is defined to be the graph  $G'$  with the same set,  $V$ , of vertices, but whose set of edges,  $E'$ , defined as follows:

*There is an edge from vertex  $u$  to vertex  $v$  if and only if there is a path of edges in  $G$  from vertex  $u$  to vertex  $v$ .*

If we create a Layer 3 graph of an IP network by specifying each IP device as a vertex and creating two edges, one in each direction, for each pair of devices that are IP-adjacent (packet can move from between devices), we can then apply Warshall to compute transitive closure of our network graph. Networks generally provide interconnectivity between all devices within physical network partitions. In this base case, the transitive closure of our graph would show every node to be reachable from every other node. This first step provides a representation of simple IP Layer 3 connectivity.

This simple Layer 3 connectivity does not reflect the configuration of packet filters (e.g., Access Control Lists and Firewall Rules) to prevent delivery of certain packets for security reasons, nor of routing policies aimed at limiting the paths packets may traverse, for traffic engineering purposes. The configured reachability of the network is thus significantly different from that implied by the simple Layer 3 topology. In order to fully capture reachability we will need to extend Warshall’s algorithm to take these impacts into account.

Warshall represents a digraph as an adjacency matrix. For a digraph with  $N$  vertices, the adjacency matrix,  $A$ , is an  $N \times N$  matrix of 1’s and 0’s whose entry at  $a_{ij}$  is equal to:

*1 if there is an edge from  $i$  to  $j$ ,  
0 otherwise.*

Given two adjacency matrices,  $A$  and  $B$ , we can compute their Boolean product, in a manner analogous to regular matrix multiplication in linear algebra, taking the dot product of each row of  $A$  with each column of  $B$ , but replacing the arithmetic operations of multiplication and addition in the dot product with the Boolean AND and OR operations. Thus, in the product  $C = A * B$ , we define:

$$c_{ij} = \bigvee_{k=1}^N (a_{ik} \wedge b_{kj})$$

If we perform a Boolean matrix multiplication of an adjacency matrix with itself then the product,  $C = A^2$  has the property that for every pair  $(i, j)$ ,  $c_{ij} = 1$  if and only if there is a path of 2 or fewer “hops” from vertex  $i$  to vertex  $j$ . Generally, for  $C = A^m$ ,  $c_{ij} = 1$  if and only if there is a path of  $m$  or fewer “hops” from vertex  $i$  to vertex  $j$ . Thus,

with N vertices, we can compute the transitive closure of our adjacency matrix by computing AN.

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$A^N = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

### Graph Analysis with Packet Filters

Geoffrey Xie and researchers from Carnegie Mellon University and AT&T Labs have proposed an extension to the Warshall algorithm to compute the reachability of network elements in the presence of packet filters [2]. In order to accomplish this we will need the following software components:

A Packet Set Data Structure

The IP Header space consists of the following fields from the IP Packet Header:

1. Source Address - 32 bits: 000.000.000.000 thru 255.255.255.255
2. Destination Address - 32 bits: 000.000.000.000 thru 255.255.255.255
3. Protocol - 8 bits: 0 thru 256
4. DSCP (or TOS) - 8 bits: 0 thru 256
5. Destination Port - 16 bits: 0 thru 65535

The IP Header space component supports the Boolean set operations: union, intersection and difference. These operations will be performed repeatedly in all of the algorithms we need, so the implementation will be performance-critical.

### The Packet Filter Classifier

The Packet Filter Classifier will take as input a set of Access Control Lists from routers and rule sets from firewalls and produce a set of Packet Sets that may be delivered across the applicable interface.

The Packet Filter Classifier will iterate over the rules. Each rule specifies a Packet Set and a predicate, either to permit or to deny. The Classifier will maintain 3 classes of Packet Sets, permitted, denied and unclassified, with the entire IP packet space initially unclassified. As each rule is encountered, the intersection of its subject Packet Set with the unclassified class is unioned with the target class specified by the predicate. By default, when all rules have been processed, the remaining unclassified sets are unioned to the “deny” sets. Those packet sets found to be permitted may be persisted with the as an additional processing step with modeling ACL’s and Firewall rules in a network configuration repository.

### Layer3 Network Topology

The first step in building the reachability model is capturing Layer 3 topology. Layer 3 topology consists of the following:

- Router interface-to-interface relationship
- Association of packet filter on interfaces (ACLs)

Utilizing this Layer 3 Topology model we define a set of vertices for each Layer 3 device, each endpoint subnet in the network, and a vertex to represent the public Internet.

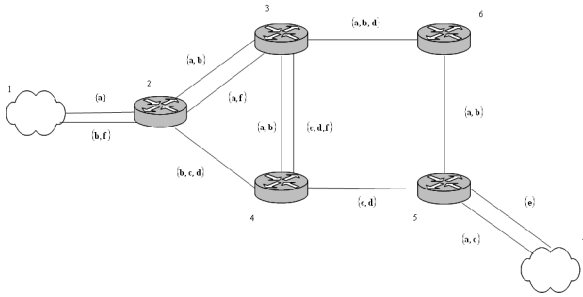
### The Service Path Algorithm

From the retrieved Layer 3 topology we create a digraph comprising nodes of the following types:

1. A node for each router and for each additional network device that performs packet filtering (e.g. firewalls).
2. A node for each subnet that contains host computers that are either providers or consumers of the services that whose deliverability we wish to analyze. Typically these will be exterior to the network; i.e. they will be linked to router interfaces that are linked to no other router. However, there may be hosts in the interior of the network – these may be identifiable either by

the topology discovery mechanisms or by querying the user.

The edges of the graph comprise the Layer 3 links between the routers and the host subnets. Two edges, one in each direction, connect each pair of nodes. At each edge we apply the packet filter classifier to the ACLs or firewall rules applied to the outbound interface of the device at the tail of the edge and to the inbound interface of the device at the head. We compute the intersection of the two resultant packet sets.



We will populate an adjacency matrix, not with 1's and 0's, but with references to these packet sets, wherever vertices are adjacent, and nulls where they are not. The diagonal will be initialized with the sets comprising the entire packet space. By doing so, we make a trade-off between processing and memory.

$$A = \begin{pmatrix} \{a,b,c,d,e,f\} & \{a\} & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \{b,f\} & \{a,b,c,d,e,f\} & \{a,b\} & \{b,c,d\} & \emptyset & \emptyset & \emptyset \\ \emptyset & \{a,f\} & \{a,b,c,d,e,f\} & \{a,b\} & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \{c,d,f\} & \{a,b,c,d,e,f\} & \{c,d\} & \{a,b,d\} & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \{a,b,c,d,e,f\} & \emptyset & \{a,c\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \{a,b\} & \{a,b,c,d,e,f\} & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \{e\} & \emptyset & \{a,b,c,d,e,f\} \end{pmatrix}$$

Now, define a set-based matrix multiplication for our adjacency matrix by specifying the dot product in terms of unions and intersections of the packet sets. For the product  $C = A * B$ , we then have:

$$c_{ij} = \bigcup_{k=1}^N (a_{ik} \cap b_{kj})$$

The product of the adjacency matrix with itself represents the sets of packets that are deliverable between nodes two hops away.

$$A^2 = \begin{pmatrix} \{a,b,c,d,e,f\} & \{a\} & \{a\} & \emptyset & \emptyset & \emptyset & \emptyset \\ \{b,f\} & \{a,b,c,d,e,f\} & \{a,b,c,d\} & \{c,d\} & \{a,b\} & \emptyset & \emptyset \\ \{f\} & \{a,f\} & \{a,b,c,d,e,f\} & \{a,b\} & \{a,b,d\} & \emptyset & \emptyset \\ \emptyset & \emptyset & \{f\} & \{c,d,f\} & \{c,d\} & \{d\} & \{e\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \{a,b,c,d,e,f\} & \emptyset & \{a,c\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \{a,b\} & \{a,b,c,d,e,f\} & \{a\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \{e\} & \emptyset & \{a,b,c,d,e,f\} \end{pmatrix}$$

**Important Note:** The IP header includes a Time-To-Live field, limiting the number of hops and by implication the number of matrix multiplications.

### Impacts of Routing Policy

Xie, et al. propose a method by which routing architecture and policy can be incorporated into the algorithm by modeling their impacts as packet filters [2].

First we create a routing instance graph, a graph whose vertices are routing processes on network devices and whose edges represent the adjacency of routing process: routing processes are adjacent if they are configured on IP-adjacent interfaces and employ the same routing algorithm. A Routing Information Base is created for each process and is initialized with a route for each interface to its configured subnet, as well as any manually configured and static routes. The routes are then distributed from each process to its neighbor subject to configured route filters and distribution policies. The effect of routing protocols represented as “permitted” packet sets and all others as “denied”. These virtual packet filters may then be intersected with those on the outbound edge of the device. Subsequent application of the matrix multiplications will then reflect deliverability of packets only over paths permitted by the routing configuration.

### Service Path Capture

The reachability algorithm will inform us of the deliverability of packets across a network, it provides no information as to the paths that deliverable packets may traverse. The algorithm, however, may be enhanced to easily capture this information.

To accomplish this, we will associate with each primary packet set contained in an entry,  $a_{ij}$ , a list of edges. Each list will be initialized with the edge of the adjacency of the entry in which it appears. Thus, in our example above, the entry for  $a_{21}$  contains packet sets b and f, both of which

will have an edge list that is initialized with the edge, <2,1>. The packet sets in the entries on the diagonal will have empty lists.

$$A = \begin{pmatrix} \{a,b,c,d,e,f\} & \{\emptyset\} & \emptyset & \emptyset & \emptyset & \emptyset \\ \{\emptyset, \{\emptyset\}\} & \{a,b,c,d,e,f\} & \{\emptyset, \{\emptyset\}\} & \{\emptyset, \{\emptyset, \{\emptyset\}\}\} & \emptyset & \emptyset \\ \emptyset & \{\emptyset, \{\emptyset\}\} & \{a,b,c,d,e,f\} & \{\emptyset, \{\emptyset\}\} & \emptyset & \emptyset \\ \emptyset & \emptyset & \{\emptyset, \{\emptyset, \{\emptyset\}\}\} & \{a,b,c,d,e,f\} & \{\emptyset, \{\emptyset\}\} & \emptyset \\ \emptyset & \emptyset & \emptyset & \{\emptyset, \{\emptyset\}\} & \{a,b,c,d,e,f\} & \{\emptyset, \{\emptyset\}\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \{\emptyset, \{\emptyset\}\} & \{a,b,c,d,e,f\} \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \{\emptyset, \{\emptyset\}\} \end{pmatrix}$$

In performing the matrix multiplications, we will perform an additional operation when computing the dot product to create the entry for  $c_{ij}$ : when the intersection of the packet sets in  $a_{ik}$  and  $b_{kj}$  is found to be non-empty, we will have one or more primary sets in both  $a_{ik}$  and  $b_{kj}$  that will be copied into the entry for  $c_{ij}$ . For each such set, we union the edge list associated with the set in  $a_{ik}$  with that associated with the set in  $b_{kj}$  and store the result as the list associated with the set in  $c_{ij}$ . Thus, in our example, when we compute the product of row 2 with column 5, we obtain the intersection of the set, {b,c,d} from  $a_{2,4}$  and {c,d} from  $a_{4,5}$  resulting in the set {c,d}. The edges then associated at  $a_{2,5}$  will combine those found for each set in  $a_{2,4}$  and  $a_{4,5}$ , so that c and d will each have edges <2,4> and <4,5> it its list stored with element  $a_{2,5}$ .

Through repeated multiplications, we thus accumulate for each packet flow in  $c_{ij}$ , the set of edges that may be included in all paths from node i to node j. Our final  $A^N$  matrix will contain all of the path information for all deliverable packets. The first row of this matrix will then look like this:

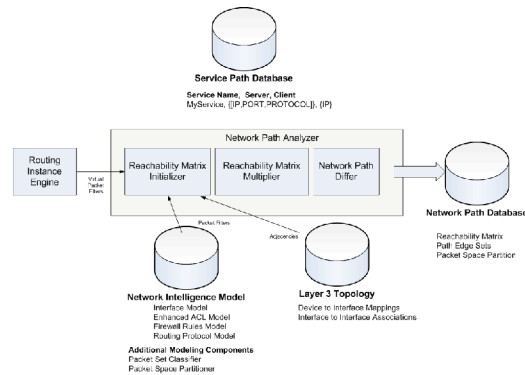
$$\begin{pmatrix} \{a,b,c,d,e,f\} & \{a\} & \{a\} & \{a\} & \{a\} & \{a\} & \{a\} \\ & \langle 1,2 \rangle & \langle 1,2 \rangle & \langle 1,2 \rangle & \langle 1,2 \rangle & \langle 1,2 \rangle & \langle 1,2 \rangle \\ & & \langle 2,3 \rangle & \langle 2,3 \rangle & \langle 2,3 \rangle & \langle 2,3 \rangle & \langle 2,3 \rangle \\ & & & \langle 2,4 \rangle & \langle 2,4 \rangle & \langle 2,4 \rangle & \langle 2,4 \rangle \\ & & & & \langle 4,3 \rangle & \langle 4,3 \rangle & \langle 4,3 \rangle \\ & & & & & \langle 3,6 \rangle & \langle 3,6 \rangle \\ & & & & & & \langle 4,5 \rangle \\ & & & & & & & \langle 6,5 \rangle \\ & & & & & & & & \langle 5,7 \rangle \end{pmatrix}$$

From this matrix we may populate the Network Path Database, capturing the set of packets deliverable between any pair of IP addresses in the network.

### Service Path Analysis

By representing the deliverability and the available paths in terms of the IP header fields, we have extended the Layer 3 topology to incorporate Layer 4 (protocol UDP or TCP) and Layer 7 (port number). By mapping named services to IP addresses and port numbers we can bind network path information to the application or service. This set of explicit paths defines the logical boundary of a network service and explicitly partitions the network into the sub-network, which delivers the Application to a given set of customers.

### Service Path Analysis Engine



### Service Management Use Cases

Once the network relationships are captured and analyzed. A broad set of service oriented management applications become possible.

Given only the Network Path DB the following use general questions can be answered:

- Given a set of source [IP address, Port, Protocol] triples what set of IP addresses are reachable?
- Given a set of source [IP address, Port, Protocol] triples and a set of destination IP addresses, what is the set of network paths?

By establishing a Service Path DB: an database of service source information and user destination information and leveraging the captured Network Path DB there are general set of service questions, which are answered:

- Who can get this service?
- Who can't get this service?
- From a given client, which services can I get?

- For a given service what devices (and configurations) provide this service?
- What service is impacted with this change?

### ***Management Applications***

*Service Diagnostics and Trouble-shooting:* A user is unexpectedly unable to access a certain service. What is the source of the problem? Is the problem in the configuration of the devices or are devices or cabling disabled? We can test the packet header IP address of the host and the IP address and port of the server to determine if the network is, in fact, configured to deliver packets between them. This solves for the classic question: “Is the outage caused by the application or the network?”

*Service Change Report:* The difference between two reachability matrices identify what packets are deliverable under one configuration but not under the other. Network configuration management is a key focus, due the large #'s of outages are created by unknown configuration impact. A service change report could be run a priori a change to predict the impact of configuration change on services delivered by the network, enabling true change impact prediction and the impact to correlated against the service.

*Network Service Compliance:* Rules for network service traffic can be specified in terms of network-wide policy. These network policies can be expressed through service path rules. This represents a significant improvement over current device-specific compliance and allows for network compliance rules sets to be evaluated for consistency a priori any network change.

*Network Service Analytics:* Combining the Service Path DB and Network Path DB provides a service dimension to analyzing network management data. By adding network and service path context, the following questions can be answered:

- What is the total amount of hardware used to deliver a service?
- What are the hardware maintenance costs of a service?
- Is this service over or under provisioned?

### **Open Issues**

The following are outstanding issues requiring further research.

1. Handling of stateful firewall rules?
2. Accommodating MPLS traffic into the reachability model?
3. Computing implied packet filters as a function of routing policy requires further research

### **Conclusion**

The potential for Service Path Analysis is far reaching for network management and IT management in general. The breakthrough work accomplished by Xie et al has open the possibilities for managing networks from a perspective that allows the network to be better designed and utilized. This approach has the potential of changing the economics of networking with respect to planning application rollout, determining optimum use and diagnosing outages. Given this approach, the SPA model can be the hub for all network operational data (FCAPS). Currently, service management approaches through Configuration Management Database (CMDB) strategies assume the network just another data source. With SPA the network configuration provides the relationship context, which unifies the entire picture of services on the network.

### **References**

- [1] Robert Sedgewick Algorithms in Java, Third Edition, Part 5: Graph Algorithms. Addison Wesley, 2003.
- [2] Geoffrey Xie, Jibin Zhan, David A. Maltz, Hui Zhang, Albert Greenberg, Gisli Hjalmytsson, and Jennifer Rexford. On static reachability analysis of IP networks. Technical Report CMU-CS-04-146, Carnegie Mellon University, 2004.