Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 21, 2016

H. Seidel, Ed.
BENOCS GmbH

October 19, 2015

ALTO map calculation from live network data
draft-seidel-alto-map-calculation-00

Abstract

This document describes a process to generate ALTO compliant information from live network data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 21, 2016.

Table of Contents

1. Overview

The ALTO protocol is designed to export network information to applications that need to select suitable endpoints among a wider set of available ones. However, it does provide details about the network information retrieval and processing.

This document describes a process to generate ALTO network and cost maps from live network data and provides experience details about that process in a large network.

The ALTO map generation process comprises three steps. The first step is to gather information which is described in Section 2. Subsequently the gathered data is processed which is described in Section 3. The last section defines methods to generate ALTO network and cost maps from the processed data.

In general it is not possible to gather detailed information about the whole Internet since it is segmented in many networks and in most

cases it is not possible to collect information across network borders. Hence, information sources are limited to the own network.

2. Network Data Collection

The first step in the process of generating ALTO network and cost maps from live network data is to gather the required information from the network. This comprises at least topology and routing information which contain details about present endpoints and their interconnection and form the basic dataset. With this information it is possible to compute paths between all known endpoints. The basic dataset can be extended by many other information obtainable from the network.

2.1. Topology Information

Topology information comprises details about routers and their interconnection, also called links, within a network. Such information are provided by various sources. The most prevalent sources are interior gateway protocols (IGPs) which can be divided in link-state (e.g. IS-IS, OSPF) and distance-vector protocols (RIP). Most suitable are link-state protocols since every router propagates its information throughout the whole network. Hence, it is possible to obtain information about all routers and their neighbors from one single router in the network. In contrast, distance-vector protocols are less suitable since routing information is only shared among neighbors. To obtain the whole topology with distance-vector routing protocols it is necessary to retrieve routing information from every router in the network.

Since IGPs lack of the possibility to easily steer traffic within the network many network operators utilize MPLS to enable custom path configuration. MPLS uses labels to identify configured paths. These labelled paths create an overlay network on top of the actual network forming its own virtual topology. Part of the MPLS architecture is the Label Distribution Protocol (LDP) that is used to configure the paths and therefore can be used to obtain MPLS topology information.

With the rise of software-defined networking (SDN) and its abstraction of network management achieved by the decoupling of network data and control plane network management became easier, since the hardware does not require manual configuration anymore. This is done by SDN controllers that relay routing information to the switches and routers. So instead of gathering topology information from the hardware within the network it can be fetched from SDN controller.

The data sources mentioned so far are only a subset of potential topology sources and depending on the network type, (e.g. mobile, satellite network) different hardware and protocols are in operation to form and maintain the network.

2.2. Routing Information

Routing information comprises details about known endpoints and paths in a network. In general there are two types of protocols, that disseminate routing information on the Internet, interior gateway (IGP) and exterior gateway protocol (EGP). While IGPs provide details about endpoints and links within the own network, EGPs are used to provide details about links to endpoints in foreign networks outside of the operation scope of the own network. A path is described by two endpoints and the traversed links. Routing protocols assign metric values to links called link weights which represents the cost to send data across a link. With the knowledge about the link weights routing algorithms (e.g. Bellman-Ford) calculate the path through the network for each source-destination endpoint pair in the network.

The most widely-used routing protocols on the Internet are IS-IS, OSPF and BGP. IS-IS and OSPF are IGPs and have already been introduced in Section 2.1. BGP is an EGP based on the distance-vector algorithm. As characteristic for distance-vector protocols, it only shares routing information among neighbors. If no BGP route reflector is present that collects routing information from all BGP routers it is necessary to pick up that information directly from each BGP router in the network. However, BGP is not only used as EGP but also alongside IGPs (iBGP) to distribute known endpoints and the corresponding metrics within a network.

In large real life network deployments such as ISP networks IGPs are mainly used to disseminate topology information and link metrics. Endpoint information such as subnets and attachment points are mostly distributed by (i)BGP.

The previously mentioned SDN controller of a SDN is also a suitable source for routing information. In general, as with topology details the available routing information sources mainly depend on the network type. However, our work focuses on networks using IS-IS or OSPF as IGP and BGP as EGP.

2.3. Extended Information

Besides topology and routing information which are fundamental to know how data between endpoints are exchanged, networks have a multitude of other attributes about its state, condition and

operation. That comprises but is not limited to attributes like link utilization, bandwidth and delay, ingress/egress points of data flows from/towards endpoints outside of the network up to the location of nodes and endpoints. In general, extended information comprises all information that a network provides which does not belong to topology or routing. Typical sources are SNMP, Netflow or an operations support system (OSS).

2.4. Example Network

Figure 1 depicts a network which is used to explain the steps carried out in the course of this document. The network consists of nine routers (R1 to R9) whereat two of them are border routers (R1 + R8) connected to neighbored networks (AS 2 to AS 4). Furthermore, AS 4 is not directly connected to the local network but has AS 3 as transit network. The links between the routers are point-to-point connections, hence a /30 subnet is sufficient for each. These connections also form the core network which we assigned the 100.1.1.0/24 subnet. This subnet is large enough to provide /30 subnets for all router interconnections. In addition to the core network the local network also has five client networks attached to five different routers (R2, R5, R6, R7 and R9). Each client network is a /24 subnet with 100.1.10x.0 (x = [1..5]) as network address.

The example network utilizes two different routing protocols, one for IGP and another for EGP routing. The used IGP is a link-state protocol (IS-IS). The applied link weights are shown in Figure 2. To obtain the topology and routing information from the network the ALTO server must be connected directly to one of the routers (R1...R9), Furthermore, the server must be enabled to communicate with the router and vice versa.

The applied EGP in the network is the border gateway protocol (BGP), which is used to route between autonomous systems (AS). So, BGP is running on the two border routers R1 and R8. Furthermore, internal BGP is used to propagate external as well as internal prefixes within the network boundaries. Hence it is running on every router with an attached client network (R2, R5, R6, R7 and R9). If no route reflector is present it is necessary to fetch routes from each BGP router separately. Otherwise, only one connection to route reflector is sufficient to obtain all routes.

For monitoring purposes, SNMP is enabled on all routers within the network. Thus, using SNMP an ALTO server is capable to obtain several additional information about the state of the network. In this example, utilization, latency and bandwidth information are retrieved periodically via SNMP from the network components to get and keep an up-to-date view on the network situation.

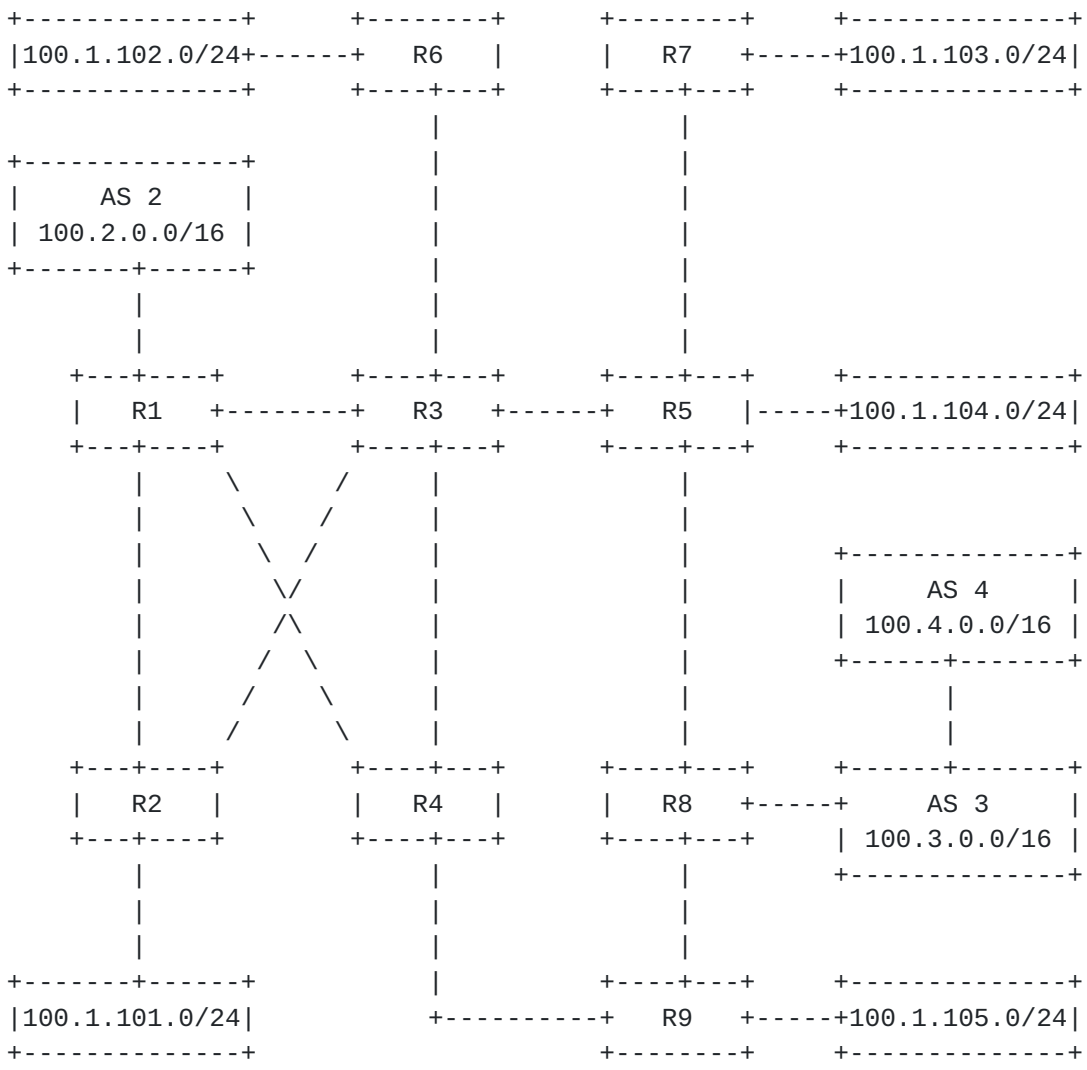

Figure 1: Example Network

	R1	R2	R3	R4	R5	R6	R7	R8	R9
R1	0	15	15	20	-	-	-	-	-
R2	15	0	20	-	-	-	-	-	-
R3	15	20	0	5	5	10	-	-	-
R4	20	-	5	0	5	-	-	-	20
R5	-	-	5	5	0	-	10	10	-
R6	-	-	10	-	-	0	-	-	-
R7	-	-	-	-	10	-	0	-	-
R8	-	-	-	-	10	-	-	0	10
R9	-	-	-	20	-	-	-	10	0

Figure 2: Example Network Link Weights

In summary, the following information are collected from the network:

IS-IS: topology, Link weights

BGP: prefixes, AS numbers, AS distances, metrics

SNMP: latency, utilization, bandwidth

2.5. Experiences

To be able to retrieve the information presented in this chapter we implemented many of the mentioned protocols. While connecting to the different data source we faced behavior that was different than we anticipated from our interpretation of the corresponding protocol. This includes behavior caused by older versions of the protocol specification, a lax interpretation on the remote side or simply incompatibility with the corresponding standard.

3. Network Data Processing

Due to the variety of data source available in today's network it is necessary to aggregate the information and define a suitable data model that can hold it efficiently and easily accessible. The most suitable model is an annotated directed graph, since it perfectly fits to represent topology and the attributes can be annotated at the corresponding positions in the graph itself. More details about the topology graph are provided in the following section.

3.1. Topology Graph

The topology graph is the data model that represents the surveyed network. The node of the graph represents the routers in the network while the edges stand for the links that connect the routers. Both routers and links have a set of attributes that holds information gathered from the network.

3.1.1. Router

The routers connect the different segments of the network. Each router holds a basic set of information which are described in the upcoming list:

ID: Unique ID within the network to identify the router.

Neighbor IDs: List of directly connected routers.

Endpoints: List of connected endpoints. The endpoints can also have further attributes themselves depending on the network and address type. Such potential attributes are costs for reaching the endpoint from the router, AS numbers or AS distances. Be aware

that endpoints can belong to more than one router, for example when they are assigned to link interfaces.

In addition to the basic set many more attributes can be assigned to router nodes. This mainly depends on the utilized data sources. Such additional attributes can be geographic location, host name and/or interface types, just to name a few.

3.1.2. Links

A link is a unidirectional connection between two routers. The basic information set hold by a link is described in the following list:

Source ID: ID of the source router of the link.

Destination ID: ID of the destination router of the link.

Weight: The cost to cross the link defined by the used IGP.

Additional attributes that provide technical details and state information can be assigned to links as well. The availability of such additional attributes depends on the utilized data sources. Such attributes can be characteristics like maximum bandwidth, utilization or latency on the link as well as the link type. Even the link cable color is a possible attribute, even though its sense is doubtful.

3.2. Example

Picking up the example from Section 2.4 the example network shown in Figure 1 already represents the layout of our internal network graph where the routers R1 to R9 represent the nodes and the connections between them are the links.

3.2.1. Example Router R2

ID: 2

Neighbor IDs: 1,3 (R1, R3)

Endpoints:

Endpoint: 100.1.101.0/24

Metric: 10 (default client subnet metric)

ASNumber: 1 (our own AS)

ASDistance: 0

Host Name: R2

R2 has one directly attached IPv4 endpoint that belongs to its own AS.

3.2.2. Example Router R8

ID: 8

Neighbor IDs: 5,9 (R5, R9)

Endpoints:

Endpoint: 100.3.0.0/16

Metric: 100

ASNumber: 3

ASDistance: 1

Endpoint: 100.4.0.0/16

Metric: 200

ASNumber: 4

ASDistance: 2

Host Name: R8

R8 has two attached IPv4 endpoints. The first one belongs to a directly neighbored AS with AS number 3. So the AS distance from our network to AS3 is 1. The second endpoint belongs to an AS (AS4) that is no direct neighbor but directly connected to AS3. To reach endpoints in AS4 it is necessary to cross AS3, which increases the AS distance by one.

3.2.3. Example Link

The collectable link attributes are equal for all links and only their values differ. The attributes utilization, bandwidth and latency collected via SNMP where added to the existing example from Section 2.4. Taking the links between R1 and R2 as example, we get the following link attributes:

R1->R2:

Source ID: 1

Destination ID: 2

Weight: 15

Bandwidth: 10Gbit/s

Utilization: 0.1

Latency: 2ms

R2->R1:

Source ID: 2

Destination ID: 1

Weight: 15

Bandwidth: 10Gbit/s

Utilization: 0.55

Latency: 5ms

Since the values for utilization and latency are very volatile the presented values are only exemplary.

3.3. Experiences

Processing network information is very complex with a high demand in resources. Gathering information from an autonomous system connected to Internet means the software and machine must be able to store and process hundreds of thousands of prefixes, several hundreds of megabytes of Netflow information per minute, information from hundreds of routers and attributes of thousands of links. Hence, a lot of disk memory, RAM and CPU cycles as well as efficient algorithms are required to process the information as they come in.

4. ALTO Map Calculation

The goal of the ALTO map calculation process is to get from the graph presentation of the network as described in Section 3 to a coarse-grained matrix presentation. The first step is to generate the

network map. Only after the network map has been generated it is possible to compute the cost map since it relies on the network map.

4.1. Network Map Calculation

To generate an ALTO network map a grouping function is required. A grouping function processes information from the network graph to group endpoints into PIDs. The way of grouping is manifold and algorithm can utilize all information provided by the network graph to perform the grouping. The functions may omit certain endpoints to simplify the map or to hide details about the network that are not intended to be published with the resulting ALTO network map.

For IP endpoints which is either an IP (version 4 or version 6) address or prefix, [RFC7285] requires the use of longest-prefix matching algorithm (Section 5.2.4.3 [RFC1812]) to map IPs to PIDs. This requirement yields the constraints that every IP must be mapped to a PID and that the same prefix or address is not mapped to more than one PID. To meet the first constraint every calculated map must provide a default PID that contains the prefixes 0.0.0.0/0 for IPv4 and ::/0 for IPv6. Both prefixes cover their entire address space and if no other PID matches an IP endpoint the default PID will. The second constraint must be met by the grouping function since it assigns endpoints to PIDs, so in case of collision the grouping function must decide which PID get the endpoint. Be aware that these or even other constraints may apply to other endpoint types depending on the used matching algorithm.

4.1.1. Network Map Example

A simple example on how such grouping can work is to group per host name from our network graph. Each router host name is the name for a PID and and the attached endpoints are the member endpoints of the corresponding router PID. Additionally backbone prefixes should not appear in the map so they are filtered out. The following table shows the resulting ALTO network map based on the example network from Section 2.4:

PID	Endpoints
R1	100.2.0.0/16
R2	100.1.101.0/24
R5	100.1.104.0/24
R6	100.1.102.0/24
R7	100.1.103.0/24
R8	100.3.0.0/16, 100.4.0.0/16
R9	100.1.105.0/24
default	0.0.0.0/0, ::/0

Figure 3: Example ALTO Network Map

Since router R3 and R4 have no endpoints assigned they are not represented in the network map. Furthermore, as previously mentioned the "default" PID was added to represent all endpoints that are not part of the example network.

4.1.2. Experiences

Large IP based networks consist of hundreds of thousands of prefixes which are mapped to PIDs in the process of network map calculation. As a result, network maps get very large (up to tens of megabytes). However, depending on the design of the network and the chosen grouping function the calculated network maps contains redundancy that can be removed. There are at least two ways to reduce the size by removing redundancy.

First, Adjacent IP prefixes can be merged. When a PID has two adjacent prefix entries it can merge them together to one larger prefix. It is mandatory that both prefixes are in the same PID. However, it cannot be ruled out that the large prefix is assigned to another PID. This MUST BE checked and it is up to the grouping function whether it merges the prefixes and removes the larger prefix from the other PID or not. A simple example, when a PID comprises the prefixes 192.168.0.0/24 and 192.168.1.0/24 it can easily merge them to 192.168.0.0/23.

Second, a prefix and its next-longer-prefix match are in the same PID. In this case, the smaller prefix can simply be removed since it is redundant for obvious reasons. A simple example, a PID comprises the prefixes 192.168.0.0/22 and 192.168.1.0/24 and the /22 is the next-longer prefix match of the /24, the /24 prefix can simply be removed. In contrast, if another PID contains the 192.168.0.0/23 prefix 192.168.1.0/24 cannot be removed since the next-longer prefix is not in the same PID anymore.

4.2. Cost Map Calculation

After successfully creating the network map, the next step is to calculate the costs between the PIDs, which forms the cost map. Those costs are calculated by cost functions which use the information provided by the network graph described in Section 3. Cost function calculate values unidirectional which means that it is necessary to compute the costs from every PID to every PID. In general, it is possible to use all available information in the network graph to compute the costs. In case a PID contains more than one IP address or subnet the cost function calculates a set of cost values for each source/destination IP pair. In that case a tie-breaker function is required which decides the resulting cost value. Such tie-breaker can be simple functions such as minimum, maximum or average value.

Be aware, no matter what metric the cost function is using, the path from source to destination is defined by the minimum path weight. The path weight is the sum of link weights of all traversed links. Usually, the path is determined with the Bellman-Ford or Dijkstra algorithm. Hence, the cost function must first determine the path before it can determine any other metric value. As a result, the metric value can differ from the expectation where the path with the shortest path weight was not considered. But it is also possible that more than one path with the same minimum path weight exist, which means it is not entirely clear which path is going to be selected by the network. Hence, a tie-breaker similar to the one used to resolve costs for PIDs with multiple endpoints is necessary.

An important note is that [RFC7285] does not require cost maps to provide costs for every PID pair, so if no path cost can be calculated for a certain pair the corresponding field in the cost map is left out. Administrators MAY also not want to provide cost values for other PID pairs for arbitrary reasons. Such pairs MAY BE defined before the cost calculation is performed and should be respected in the calculation process even though cost values are computable.

There is an active internet draft [I-D.yang-alto-path-vector] that proposes vector based cost maps rather then the current point-to-point matrix. A vector representation can reduce the required computation to generate cost maps from a directed graph as proposed in Section 3.

4.2.1. Cost Map Example

Based on the network map generated in Section 4.1.1 it is possible to calculate the cost maps. In this example the chosen metric for the cost map is the minimum number of hops necessary to get from source

to destination PID. Our chosen tie-breaker selects the minimum hops when more than one value is returned by the cost function.

PID	default	R1	R2	R5	R6	R7	R8	R9
default	x	x	x	x	x	x	x	x
R1	x	0	2	3	3	4	4	3
R2	x	2	0	3	3	4	4	4
R5	x	3	3	0	3	2	2	3
R6	x	3	3	3	0	4	4	4
R7	x	4	4	2	4	0	3	4
R8	x	4	4	2	4	3	0	2
R9	x	3	4	3	4	4	2	0

Figure 4: Example ALTO Hopcount Cost Map

Interesting to mention is, that R1<->R9 has several paths with equal path weights. The paths R1->R3->R5->R8->R9, R1->R3->R4->R9 and R1->R4->R9 all have a path weight of 40. Due to our minimum value tie-breaker 3 hops for the path R1->R4->R9 is chosen as value. Furthermore, since the "default" PID is sort of virtual PID with no endpoints that are part of the example network no cost values are calculated for other PIDs from or towards it.

4.2.2. Experiences

The major challenge we encountered with cost map calculations was the vast amount of CPU cycles that where required to calculate the costs in large networks. The issue was that the costs were calculated between the endpoints of each source-destination PID pair. However, very often several to many endpoints of a PID are attached to the same node, so the same path cost is calculated several times. This is clearly inefficient. So instead, we looked up the routers the endpoints of each PID are connected to in our network graph and calculated cost map based on the costs between the routers.

5. Informative References

[I-D.yang-alto-path-vector]
Bernstein, G., Lee, Y., Roome, W., Scharf, M., and Y. Yang, "ALTO Extension: Abstract Path Vector as a Cost Mode", draft-yang-alto-path-vector-01, July 2015.

[RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "A Simple Network Management Protocol (SNMP)", RFC 1157, May 1990.

[RFC1812] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

[RFC2328] Moy, J., "OSPF Version 2", RFC 2328, April 1998.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC7159] Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, March 2014.

[RFC7285] Almi, R., Penno, R., Yang, Y., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, September 2014.

[RFC7607] Kumari, W., Bush, R., Schiller, H., and K. Patel, "Codification of AS 0 Processing", RFC 7607, August 2015.

Author's Address

Hans Seidel (editor)
BENOCS GmbH

Email: hseidel@benocs.com