# A Segment List Selection Algorithm Based on Delay

## Abstract

Segment routing (SR) sets segment routing labels at the entry node of the SR domain to guide the traffic in the network to match the forwarding path with message granularity. The resulting segment routing-based traffic engineering (SRTE) can efficiently and simply perform traffic analysis. Adjustment, because segment routing converts forwarding instructions into label stacks and encapsulates them in segment routing headers, so the choice of SR label stack will greatly affect the effect of SRTE. In this article, we propose a segment routing label stack generation algorithm for the delay parameters in traffic engineering, which can meet the delay requirements of request messages as much as possible. Our algorithm considers node centrality and load balancing, and minimizes the number of segment routing labels to reduce the proportion of message headers.

## Introduction

In the SDN network scenario, the controller, as the management and control plane of the network device, can obtain global network topology information, but all network traffic is regulated by the SDN mechanism, which often lacks some flexibility and timeliness. For example, all network traffic engineering uses SDN-based

As a current traffic engineering scheduling scheme commonly used in the industry, MPLS is originally intended to use labels to replace switch IP for fast forwarding. The operations of matching labels and assigning ports can be completed without sending the CPU, so it reduces The switch has a delay in processing and checking the table of messages, which improves the forwarding efficiency. However, the distribution and notification of labels requires internal gateway protocol (IGP) or open shortest path priority protocol (OSPF) for transmission. In addition, when MPLS is working, each next hop is determined, that is, a complete tunnel in the network needs to be reserved for traffic under the MPLS strategy, which needs to be used as a resource reservation protocol for traffic engineering (RSVP) The extended RSVP-TE reserves resources on the IP network. The application program running on the IP network terminal can indicate to other nodes the nature of the data packet flow (such as bandwidth, jitter, maximum burst, etc.) that the terminal will receive based on RSVP, so that the edge computing MPLS path device can perform the traffic forwarding path according to the demand distribution. However, the current MPLS deployment method has some drawbacks: First, the operating overhead of RSVP-TE as a resource reservation mechanism is usually very high; second, the IP network can originally use multi-path equal-cost routing (ECMP) to achieve native load balancing. However, MPLS, as an explicit source routing, encapsulates all the labels of the routing path in the message header, making it impossible to perform the native multi-path load balancing of the IP network between each hop.

Segment routing provides a new traffic scheduling idea. Unlike source routing, where the path of each hop is planned, segment routing only plans a destination label in the range of several hops, so that the label stack of the segment routing packet header is only used very little. The label can guide the network traffic forwarding. This can be seen as the SDN controller programming the network traffic processing mode. The SR label stack is equivalent to the programmed code, thereby achieving the network programming purpose of operating network data packets. .

However, which label needs to be allocated by the segment routing label stack needs to be determined by the control plane with a global perspective. It can be configured through the experience of the operator, or it can be calculated and allocated by the controller in the SDN scenario. At present, the commonly used label stack calculation methods are all based on bandwidth, and in a large environment where the business is more and more sensitive to delay, it is necessary to consider the use of delay as an element to guide label generation, that is, the network layer represented by SR has the effect of delay Provide a certain degree of protection. Therefore, this article will discuss the segment routing label stack generation algorithm that has the best effect of guaranteeing traffic delay based on the global perspective of the SDN controller.

Segment Routing (SR) is commonly used in traffic engineering recently, which use a Segment List placed in packet header to guide the traffic. In comparison with MPLS, SR do not source every route in path, it just point out a

It is critical to solving the problem that how to find out SR nodes in a network to optimize network performance by reducing latency. It is obverse that the selected nodes should be more centrality and the path through the node list should have a shorter latency. The problem becomes two tasks, one is to find out the most centrality nodes in the network, and the other one is to 保障流量尽可能按低时延的目的进行流量调度。

## Motivation

Several papers focus on selecting nodes SR list, but the weight of link they care about is bandwidth. [example here] Bandwidth is easy to obtain in the network, but it cannot reduce latency directly. Congestion control algorithm like Swift has verified that delay is simple and effective for congestion control in the datacenter. So, we are aimed at proposing a routing algorithm based on latency and used it in traffic engineering.

There are several sources of delay in the network, propagation delay, transmission delay, processing delay, and queuing delay. Propagation delay is definite for the time that light travels in the fiber are almost fixed. The farther the distance between the source and destination, the more time it will take to propagate. But the delay caused by the processing of the packet by the switch is almost unpredictable. The more intermediate routers along the way where the packet goes through, the higher the transmission, processing, and queuing delays for each packet. Finally, the higher a load of traffic along the path, the higher the likelihood of the packet being queued and delayed inside one or more switch buffers. In summary, the sum of the delays of processing packets in the switches at both ends of a link is an important basis for measuring the delay of a link. In particular, in-band network telemetry (INT) can measure per-hop sojourn times to provide a more accurate breakdown of delay.

Therefore, we use INT to measure the delay of every link in the network, but the INT is not the focus of this paper. We will use the delay provided by INT to propose a segment routing trans-point selection algorithm.

## Modeling

We abstract the network topology as an undirected graph of . Each edge connects two vertices, and each node has a certain degree.

## Algorithm

The algorithm for calculating segment routing trans-points is divided into two steps. The first step is the calculation of static core nodes. This step will be carried out regardless of the traffic, and only mathematical calculations are carried out for the topology of the network and the attributes of nodes. The second step is the selection of dynamic nodes. In this step, the network nodes will be weighed and judged based on the delay collected by INT, and finally, draw a conclusion of whether to add a device to the SR segment list.

### Computing nodes’ centrality

*Degree centrality* for a node is the fraction of nodes connected to it. For unipartite networks, the degree centrality values are normalized by dividing by the maximum possible degree (which is where is the number of nodes in). In the bipartite case, the maximum possible degree of a node in a bipartite node set is the number of nodes in the opposite node set 1. The degree centrality for a node in the bipartite sets with n nodes and with m nodes is

where is the degree of node [[1]](https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.bipartite.centrality.degree_centrality.html#id1).

*Betweenness centrality* is another kind of centrality. Betweenness centrality of a node is the sum of the fraction of all-pairs shortest paths that pass-through . Values of betweenness are normalized by the maximum possible value which for bipartite graphs is limited by the relative size of the two node sets 1. Let be the number of nodes in the node set and m be the number of nodes in the node set , then nodes in are normalized by dividing by

where

and nodes in are normalized by dividing by

where

In order to determine which centrality measurement method is more suitable for the scenario of selecting trans-points in this paper, the experiment will verify these two calculation methods respectively.

### Min-max hop algorithm

我们将问题抽象成如下表述，网络拓扑中有若干节点，他们的中心度可以根据拓扑属性获取，而我们想要获取的航点集合，希望在这些航点存在的情况下，选择三个大跳，实际上是segmentlist有3个，但是需要经过的交换机地址只有两个。使得在航点集合中任意选择2个航点时的最大跳数最小Min-max（但是实际上在树形拓扑中，端到端的跳数是固定的，除非是本来在一个集群下面却去别的集群绕圈这样的情况，所以这一步的计算需要在复杂的ISP网络中计算才有意义）

对于最小化最大值目标约束函数应该是经过两个航点的ecmp最长路径最短，由于ecmp是多路径等价路由，这里的等价指的是跳数等价，因此可以在这一步可以不考虑ecmp的负载均衡效果，只考虑起点->航点1->航点2->目的节点的最长路径最短。简单地说就是考虑如何降低最恶劣情况的影响。

为了求解这个目标函数，选择启发式算法进行计算，这里是用的是XXX算法，算法实现伪代码如下：

算法性能如下：

### Algorithm for node clustering based on link delay

On the basis of a set of nodes calculated by the above algorithm, consider the actual situation of the link delay associated with each node in the network. First, take the average of the link delays around each node in the candidate set, and use this value to reflect the current popularity of the node: the higher the average value, the larger the probability that the path to this node will encounter queuing, the lower the average value, the link resources around the node are relatively sufficient.

Since the average delay of the links around the node can be updated as the real-time attribute of the node, the time complexity to obtain this value is , and then the nodes are sorted according to the average experiment of the direct links of the node, the time complexity of the common quicksort is .

该问题在每一个需要重新选择SRv6标签栈的时间点，需要进行计算，将数学模型描述如下：网络中每个节点具有以时延为权重的无向边，（待讨论到底是无向图还是有向图）其中作为航点备选的节点是核心节点，将其余节点进行聚类，并最终用一个区域时延来表示一个更大范围的设备群的时延属性，这样就可以将网络中的所有设备矩阵进行降维，这实际上是一种网络节点分割算法，基于这样的分割结果，可以计算出每个区域的时延影响因子，进而选择对时延影响因子最小的一个或两个聚类核心作为标签栈。这里采用的是，算法表述如下：

算法性能分析

整个算法会涉及几个需要调整验证的参数：（1）标签栈深度D，以2（包含1个航点）和3（包含2个航点）；（2）静态算法中选择备选航点集合的数量与节点总数的比例alpha（0<alpha<1）；（3）

### Depth of Segment List

Depth of segment list in packet header is a matter of concern. A deep segment list can regulate traffic more accurately, but it will result in a lower effective load rate. Especially in SRv6, where IPv6 is used as a segment, the damage caused by the long segment list will be more serious.

Research has also shown that using 3- or 4SR did not appear to improve the results for ISP topology and was significantly harder regarding computation time and memory requirements. And the results for 3SR and 4SR are not better than those for 2SR, which coincides with the claim about 2SR being near-optimal[[2]](https://ieeexplore.ieee.org/document/8264828).

Due to the above reasons, our algorithm will only select 2SR nodes in network as the segment list in SRv6 header.

### Inhibition for link-delay feedback

The SR policy, which reflected in segment list, should not be updated frequently. SR policy in current industrial production is generally modified only when the link fails, which causes some links to have a high load rate, and the packet loss rate increases at the same time.

It is easy to understand that when a node is selected into the segment list due to the above algorithm, the traffic reaching the node will increase, and the delay of nearby links will grow. After the link delay information collected by the INT process, the algorithm will not recommend placing this node in the segment list. If the SR policy is going to adjust at this time, it will not be conducive to the transmission of the entire network traffic. Therefore, it is necessary to avoid adjusting the flow rate easily, which may cause confusion on a larger scale.

## Simulation

In the experiment, we chose bmv2 which can run p4 program as the software switch in the simulation verification. This is because in order to collect the delay of every direct link of the switch, the collection mode of INT needs to be customized, and the software switch of p4 can be better work with it. We have also implemented the forwarding capability in accordance with the SRv6 standard in bmv2. For the control plane, we wrote a python script with thrift-based API as an instance of the controller to complete the calculation and delivery of the segment list.

## Conclusion

## References