

# shortest\_path\_scheduler 类

## 1 类概述

shortest\_path\_scheduler 类，继承 dynamic\_scheduler 类。通过最短路径算法对 SFC 进行部署。

## 2 类属性

名称	描述	类型	默认值
__records	部署记录	{}	{}
log	是否输出信息	Bool	True

## 3 类方法

名称	描述
__init__(log=True)	类内部构造函数
deploy_sfc_node_equal_nf(sfc, network, shortest_path, path_delay, vnf_types)	sfc 的 nf 个数与路径节点相同时的部署方法
deploy_sfc_node_more_nf(sfc, network, shortest_path, path_delay, vnf_types)	sfc 的 nf 个数小于路径节点的部署方法
deploy_sfc_node_less_nf(sfc, network, shortest_path, path_delay, vnf_types)	sfc 的 nf 个数大于路径节点相同时的部署方法
deploy_sfc(network, sfc, p, delay_list, vnf_types)	部署单条 sfc
deploy_sfcs(network, vnf_types, sfcs, sort=True)	部署所有 sfc
deploy_sfcs_with_draw(network, vnf_types, sfcs, path='', sort=True, period=0.5)	部署 sfc 并实时画图
bubbleSort(arr)	按照流量大小对 sfcs 逆序排序

## 4 最短路径算法流程

### 4.1 最短路径算法原理

最短路径部署算法原理如图所示，找到输入输出节点的一条最短路径，并在此最短路径上部署 sfc。根据 sfc 需要 nf 数量 N 与最短路径链路节点数

量  $M+2$  (2 为输入输出节点) 数量比较, 采用了一种基于负载均衡的方案, 即延迟流量最小的虚拟链路与合并占用资源最大虚拟链路两端 NF, 分为以下三种情况:

- 1)  $M+2=N$ : 虚拟节点按照顺序部署
- 2)  $M+2>N$ : 将虚拟节点的流量最小的路径跨节点延长, 其他路径正常部署
- 3)  $M+2<N$ : 循环将 sfc 流量最大的虚拟链路连接 nf 部署在同一个节点上(首位除外), 每循环一次  $N$  减 1, 直到相等后按照 (1) 部署

