**ospf**

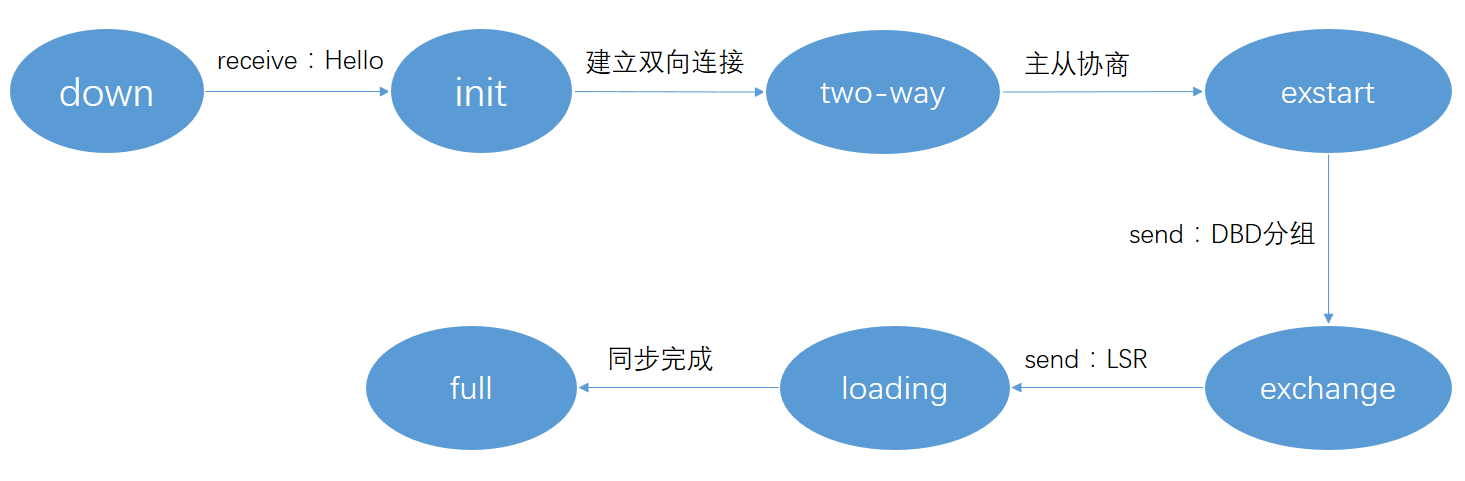
1. **概念**

开放最短路径优先协议，是由Internet工程任务组开发的路由选择协议，作用为网络间路由器互相发现、交换信息并最终生成路由表。它是一个内部网关协议(Interior Gateway Protocol，简称IGP），用于在单一自治系统（autonomous system,AS）内决策路由。

1. **原理**
2. 每台路由器通过使用Hello报文与它的邻居之间建立邻接关系
3. 每台路由器向每个邻居发送链路状态通告(LSA),有时叫链路状态报文(LSP). 每个邻居在收到LSP之后要依次向它的邻居转发这些LSP(泛洪)
4. 每台路由器要在数据库中保存一份它所收到的LSA的备份，所有路由器的数据库应该相同
5. 依照拓扑数据库每台路由器使用Dijkstra算法（SPF算法）计算出到每个网络的最短路径，并将结果输出到路由选择表中

OSPF的简化原理：发Hello报文——建立邻接关系——形成链路状态数据库——SPF算法——形成路由表。

1. **流程**



Down：新建状态；此时路由器刚接入网络，需要发送Hello消息，向网络中的其他路由器介绍自己，当收到Hello消息后进入init状态

Init：初始化状态； 新建路由收到网络中的Hello消息，此前发送的Hello消息被接受，邻居路由将新建路由器添加至本地邻接表中，邻居路由器发送回应hello信息，包含自己的router Id和所有邻居列表，此时为单向连接，新路由器收到回复消息，然后进入two-way状态

Two-way：双向连接状态；新建路由器收到回复Hello消息，发现自己的router Id在邻居表中，建立双向连接，新建路由器通过主从协商的方式进入exstart状态

Exstart：信息交换初始状态；在这个状态下，相邻路由器之间建立Master/Slave关系，然后双方交换DBD分组，进入exchange状态

Exchange：信息交换状态；本地路由器和邻居交换一个或多个DBD分组（也叫DDP），DBD包含有关LSDB中LSA条目的摘要信息，然后交换机通过LSR请求，进入loading状态

Loading：信息加载状态；收到DBD后，将收到的信息同LSDB中的信息进行比较。如果DBD中有更新的链路状态条目，则向对方发送一个LSR，用于请求新的LSA，然后更新完成后进入full状态

Full：完全邻接状态；邻接的链路状态数据库同步完成

1. **区域划分及备份**

**区域划分**：因为OSPF路由器之间会将所有的链路状态（LSA）相互交换，毫不保留，当网络规模达到一定程度时，LSA将形成一个庞大的数据库，势必会给OSPF计算带来巨大的压力；为了能够降低OSPF计算的复杂程度，缓解计算压力，OSPF采用分区域计算，将网络中所有OSPF路由器划分成不同的区域，每个区域负责各自区域精确的LSA传递与路由计算，然后再将一个区域的LSA 简化和汇总之后转发到另外一个区域，这样一来，在区域内部，拥有网络精确的LSA，而在不同区域，则传递简化的LSA。区域的划分为了能够尽量设计成无环网络，所以采用了Hub-Spoke的拓朴架构，也就是采用核心与分支的拓朴。

**DR/BDR：**当多台OSPF路由器连到同一个多路访问网段时，如果每两台路由器之间都相互交换LSA，那么该网段将充满着众多LSA条目，为了能够尽量减少LSA的传播数量，通过在多路访问网段中选择出一个核心路由器，称为 DR（Designated Router），网段中所有的OSPF路由器都和DR互换LSA，这样一来，DR就会拥有所有的LSA，并且将所有的LSA转发给每一台路由器；DR就像 是该网段的LSA中转站，所有的路由器都与该中转站互换LSA，如果DR失效后，那么就会造成LSA的丢失与不完整，所以在多路访问网络中除了选举出DR 之外，还会选举出一台路由器作为DR的备份，称为BDR（Backup Designated Router），BDR在DR不可用时，代替DR的工作，而既不是DR，也不是BDR的路由器称为Drother，事实上，Dother除了和DR互换 LSA之外，同时还会和BDR互换LSA。其实不难看出，DR与BDR并没有任何本质与功能的区别，只有在多路访问 的网络环境，才需要DR和BDR，DR与BDR的选举是在一个二层网段内选举的，即在多个路由器互连的接口范围内，与OSPF区域没有任何关系，一个区域可能有多个多路访问网段，那么就会存在多个DR和BDR，但一个多路访问网段，只能有一个DR和BDR。

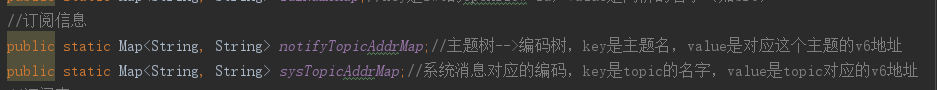
1. **优点**
2. OSPF是真正的LOOP- FREE（无路由自环）路由协议。源自其算法本身的优点。（链路状态及最短路径树算法）
3. OSPF收敛速度快：能够在最短的时间内将路由变化传递到整个自治系统。
4. 提出区域（area）划分的概念，将自治系统划分为不同区域后，通过区域之间的对路由信息的摘要，大大减少了需传递的路由信息数量。也使得路由信息不会随网络规模的扩大而急剧膨胀。
5. **小结**

ospf协议的关键有两点：1.将LSA（链路状态组播信息）传递给某一区域内所有路器，

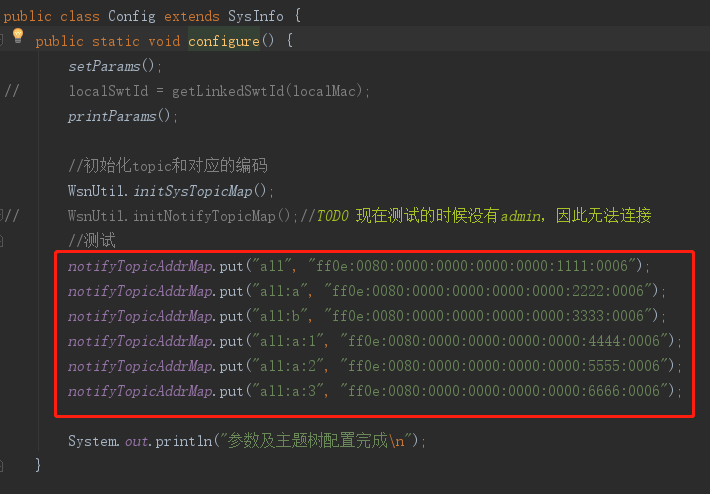
与之相对应的是RIP协议，它需要将路由表传递给与其相邻的路由器；2.路由器建立LSDB，通过spf算法计算两点间最短路径，而dijkstra算法计算单源最短路径时需要的正是LSDB中存储的链路邻接信息。

1. **学长代码**
2. **消息的传递**

首先是主题树和系统消息的编码实现



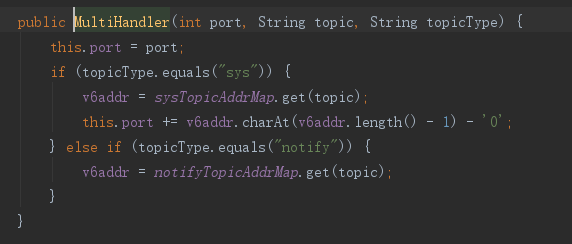
主题树 –> 编码树，notifyTopicAddrMap保存主题树和对应编码树



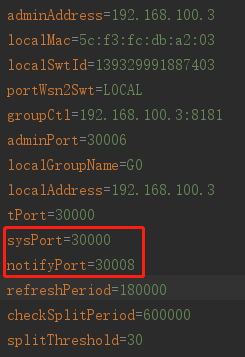
系统消息 --> 编码，sysTopicAddrMap保存对应的系统消息编码



网络中使用v6的方式传递“sys”、“notify”消息

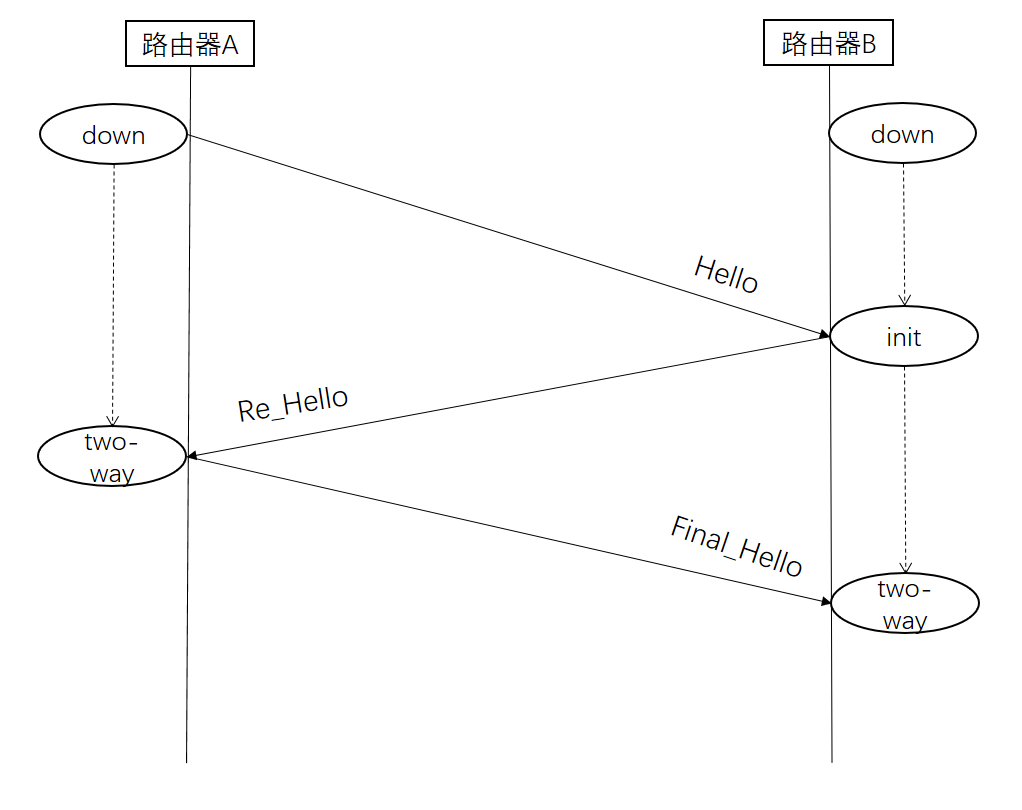


监听对应的端口



1. **集群间邻居探测**

用三次握手的方法确保建立双向连接

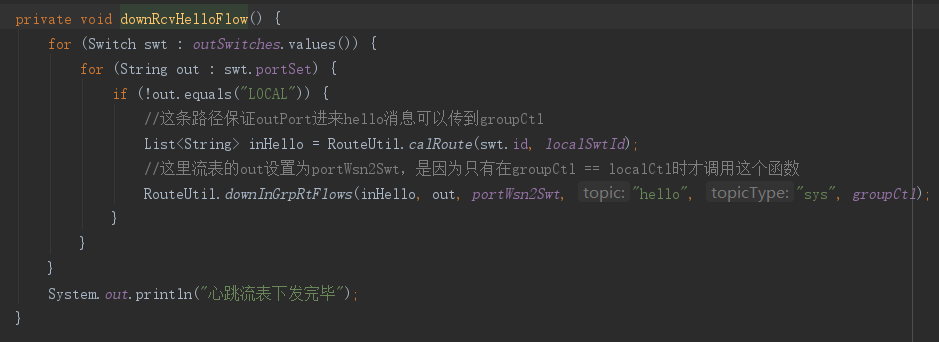


具体表现为：由集群控制器维持心跳监测，定期查询全网路由器拓扑情况；Hello、Re\_Hello、Final\_Hello消息都为“sys”系统消息，监听30000端口，消息的格式都为Hello类

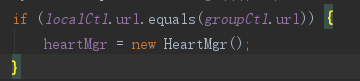
Hello类，通过内部信息是否为null区分Hello、Re\_Hello消息



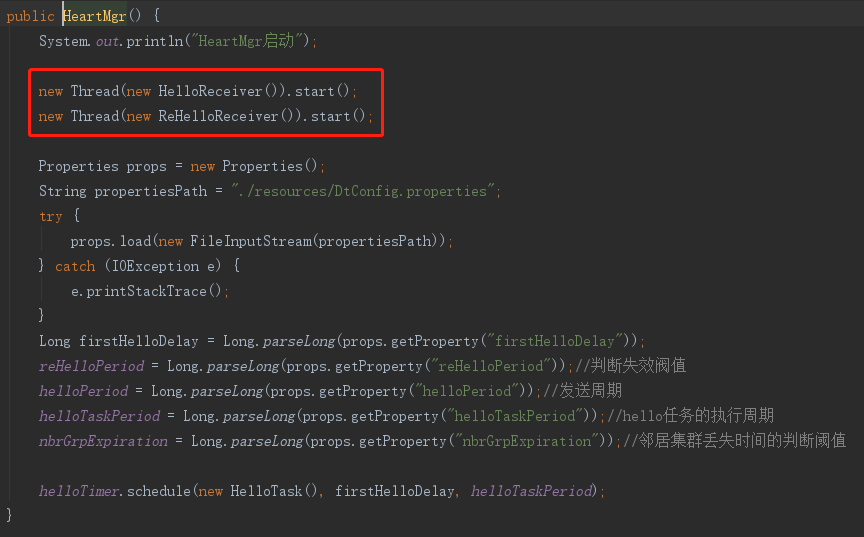
读取集群配置信息时，预先下发对外端口路径上的Hello流表



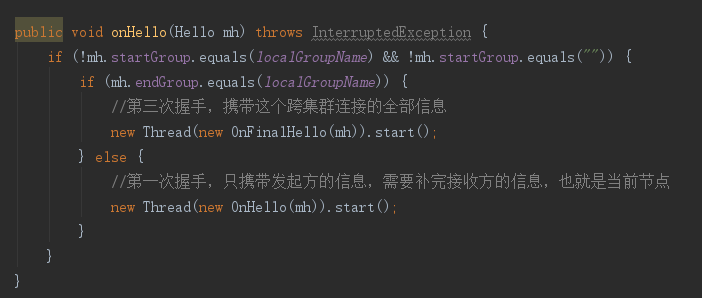
集群控制器起心跳监测



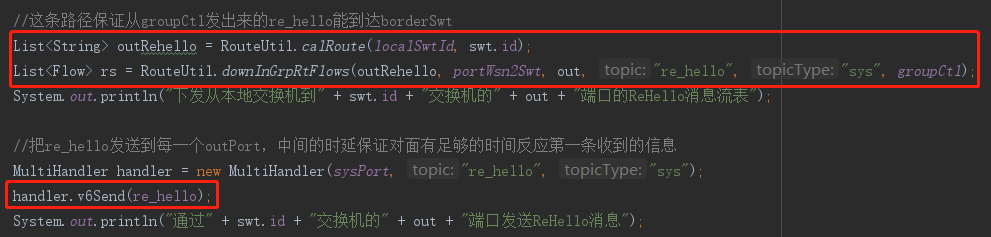
监测代码中起Hello、Re\_Hello接收消息监测



Hello消息收到时分析处理



收到Hello后，在返回Re\_Hello之前需要通过集群控制器下发流表；图中标红处为通过dijkstra算法计算出的路由，以及通过v6地址发送Re\_Hello消息



若返回的是Final\_Hello



1. LSA广播

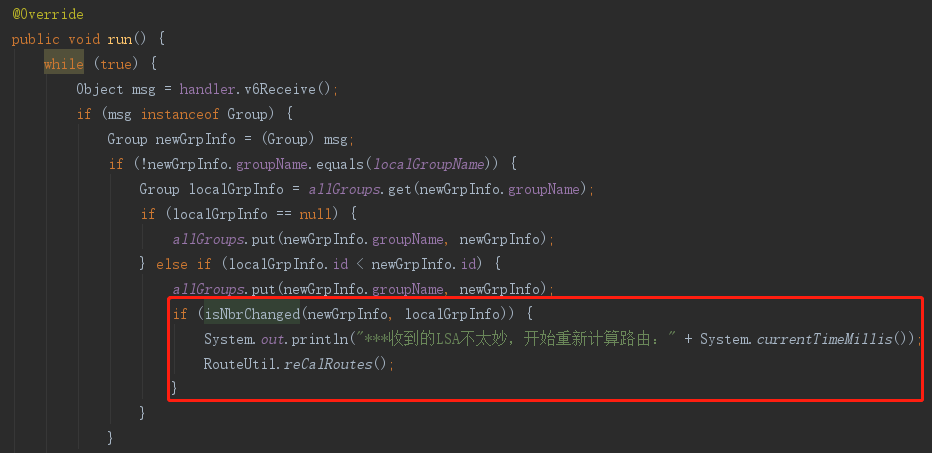
监听“sys”系统消息中的lsa主题



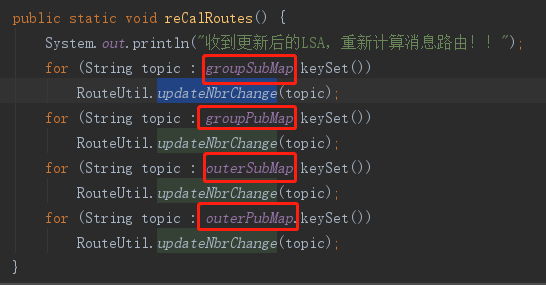
广播信息类，红色标注为重要的内容，包括该条LSA序号、集群名称、邻接关系



当收到LSA广播消息后，通过id判断是否为新的信息，并比较两者邻接关系，不同则重新计算路由

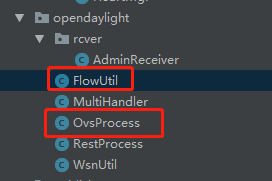


此时计算集群内外发布订阅表中的路由

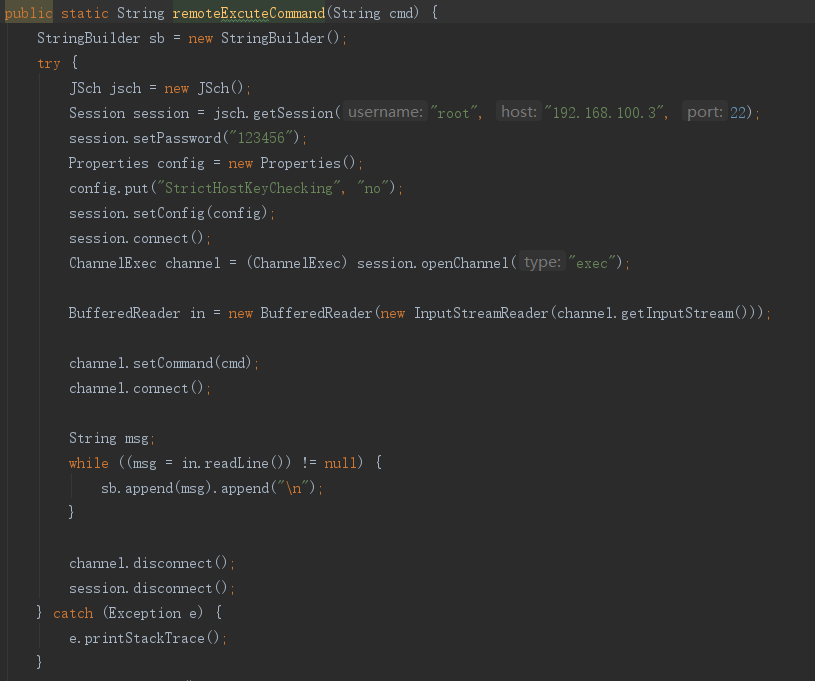


1. 下发流表

主要由两部分组成：FlowUtil负责生成待下发流表语句；OvsProcess通过集群控制器负责下发流表



远程连接集群控制器并下发流表



1. **总结**

刘昌威学长的代码总体上完成了ospf协议的功能，定义了相应的Hello消息类、监听对应的主题信息，通过三次握手的概念完成邻居的探测，定时进行心跳来维护拓扑。这一部分代码基本可以复用。

区别在于具体下发流表时需要通过jsch连接集群控制器来下发流表，这与我的设计有所不同，需要将wsn层的路由计算功能划分至节点控制器端，集群控制器维护全局拓扑；因此在实际编码过程中需要进行相关的修改。