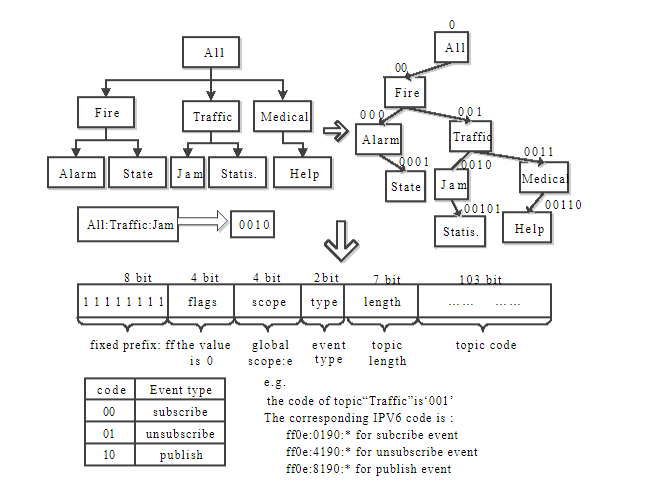
**主题树的编码**

1. **肖丹**

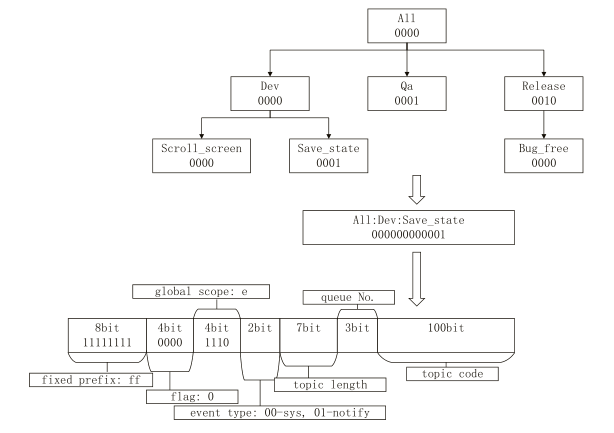


方法：将主题分级，每层编码时遵循先后顺序，第一个节点编码为0，第二个为01，第三个为011，以此类推，All 🡪 0， Fire 🡪 0，Traffic 🡪 01，Medical 🡪 011，Alarm 🡪 0，State 🡪 01，Jam 🡪 0，Statis 🡪 01，Help 🡪 0；在查询主题对应的ipv6编码时，将分级主题依次拼接：All 🡪 0，Fire 🡪 00，Traffic 🡪 001，Medical 🡪 0011，Alarm 🡪 000，State 🡪 0001，Jam 🡪 0010，Statis 🡪 00101，Help 🡪 00110

优点：将主题树编码为二叉树的形式，可快速区分不同主题

缺点：同层主题编码所需位数较大，若某一主题下子主题较多，则无法表示

1. **刘昌威**



方法：将主题树分级，每层采用固定4bit编码，0000 ~ 1111

优点：充分利用每一bit，将主题树编码为多叉树，可应对不同订阅情况

缺点：每一级只能包含16种主题情况

**采取方案**：综合考虑两者，将多个主题树连接至虚拟根节点上，考虑每一层主题数量M，为该层分配log M 比特，这样能够尽最大程度利用ipv6中的每一位

**多级流表**

概念：n个端口拥有n+1张流表，除每个端口自身的匹配流表外，共有一份系统消息匹配流表

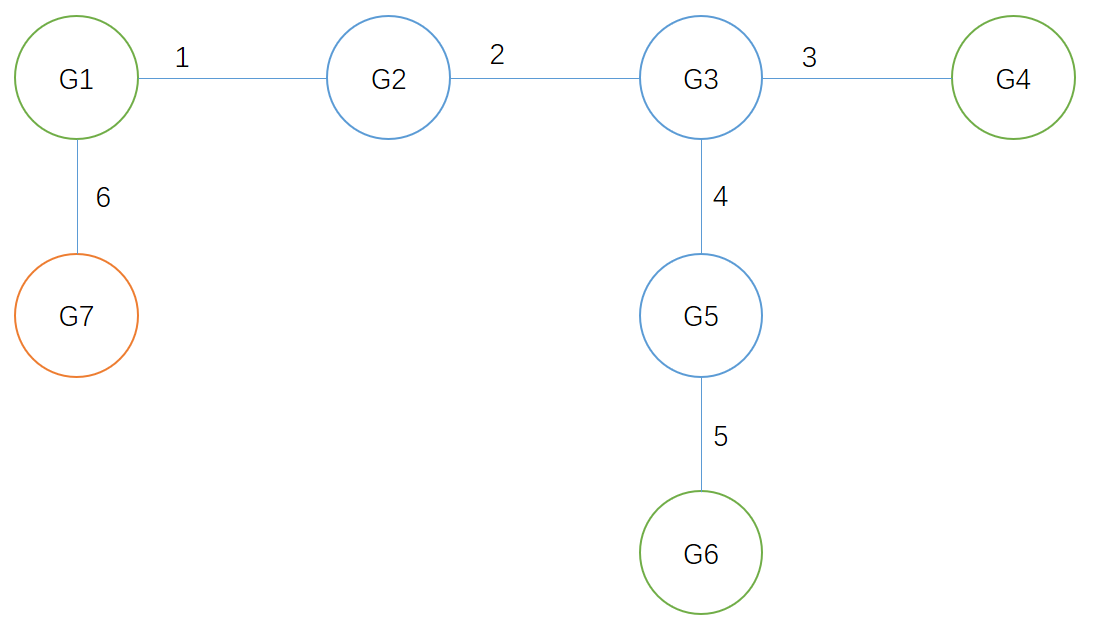
好处：职责明确，消息到来时先匹配系统消息流表，不符合则转向指定端口的流表，提升了匹配、存储效率

**路由算法**

思路：借鉴stenier思想；采用分布式计算的思想，以集群为单位，只需计算集群间的路径即可。集群控制器通过定时、事件驱动的方式获取LSA和发布、订阅信息，即根据拓扑和订阅情况计算路由，具体计算时，先计算出所有订阅、发布该主题的节点间距离，选取距离最短且集群控制器ip最小的那条路径，记录边的信息并将两个节点及该路径经过的所有过路加入结果集中，然后，依次选取发布订阅节点到结果集中节点的最短路径，添加节点和路径信息，直至所有发布订阅节点都包含在结果集中

若新增发布订阅节点，则在当前路由基础上，先判断新增节点是否在结果集中，是则不处理；否则计算到达结果集的最短路径，将所有路径下发新的流表

采取最短路径和最小ip的方式，能够确保在分布式计算的情况下，拥有全局一致性



如图所示，发布订阅节点为G1、G4、G6，经过计算，G1 – G4距离为3，G1 – G6距离为4，G4 – G6距离为3，但是G1集群ip较小，因此将G1、G4和过路节点G2、G3添加至结果集中，保存路径1、2、3，再取节点订阅G6，计算G6到结果集中的距离，最短为G6—G3，将G6、G5添加至结果集中，保存路径4、5,；路由计算结束后，若新增订阅节点G7，计算最短路径为G7 – G1，将G7添加至结果集中，保存路径6

计算结束后，各个集群判断自己是否在路径集合中，若存在则下发对应的流表

**新系统的整体流程**

1. **新建集群加入**
2. 新增集群G1，集群控制器启动，读取配置文件，得知集群内连接主机、交换机、对外端口情况；对本地主机下发订阅消息流表；下发对外端口的Hello、LSA、LDAP双向流表，并向外发送Hello消息
3. 集群G2启动流程同上，通过对外端口到达集群控制器的Hello流表通路，接收到邻居G1的Hello消息，将G1添加至本地邻居表，并封装对外端口等信息，将Hello消息向所有的对外端口发送
4. 集群G1接收到返回的Hello消息，得知自己已被添加至G2的邻居表中，将相关信息提取出来，并添加G2至本地邻居表
5. 集群G1、G2相互之间定期发送LSA维持心跳，若超时，则主动广播集群丢失情况
6. 管理员与集群控制器一样，只有在接收到邻居Hello消息后，才能感知集群，这里管理员可以通过访问数据库的方式得知LDAP主题树相关信息，并定期向网络中广播LDAP消息
7. 集群控制器在接收到LSA后会提取邻居信息、订阅信息，接收到LDAP后，会采用相关编码方式将主题树转换为对应的ipv6地址；具体编码方式如上所示
8. 新增发布—订阅节点
9. 新增订阅节点，主机host1本地保存订阅信息，加入对应主题的多播组，同时通过wsn层向集群控制器发送订阅请求，集群控制器收到订阅请求后，下发本地接收流表，发送LSA广播，通知网络中其他集群该订阅信息，同时计算新的路由，算法如上，若在路径中，则下发相应流表
10. 新增发布节点，流程同上，消息直接发送，事先下发了转发的流表项，因此消息可以送达