1. 系统功能测试
2. 管理员ui测试
3. 管理员信息展示

|  |  |
| --- | --- |
| 测试项目 | 管理员信息展示 |
| 测试目的 | 验证当管理员启动后，能够通过ui界面展示管理员信息，包括主题树信息、本集群内部信息等 |
| 测试步骤 | 1. 管理员启动并完成初始化操作  2. 启动wsn模块  3. 启动用户，进行发布、订阅的注册 |
| 预期结果 | 1. 管理员ui界面上展示管理员集群、主题树信息  2. 点击集群可以进一步查询集群内部信息，包括集群内交换机名称、下属主机及其详细信息 |
| 实际结果 | 与预期相同，具体见下图：  1. 图1：管理员信息展示  2. 图2：主题树信息展示  3. 图3：集群详细信息展示 |
| 结果分析 | 管理员ui界面系统运行正确，并能通过  1. 定时，每隔一定时间（刷新时间）监测集群信息，并更新  2. 事件驱动，当收到wsn上传的用户信息时，主动更新  相关方式更新本集群信息 |



图1 管理员信息展示

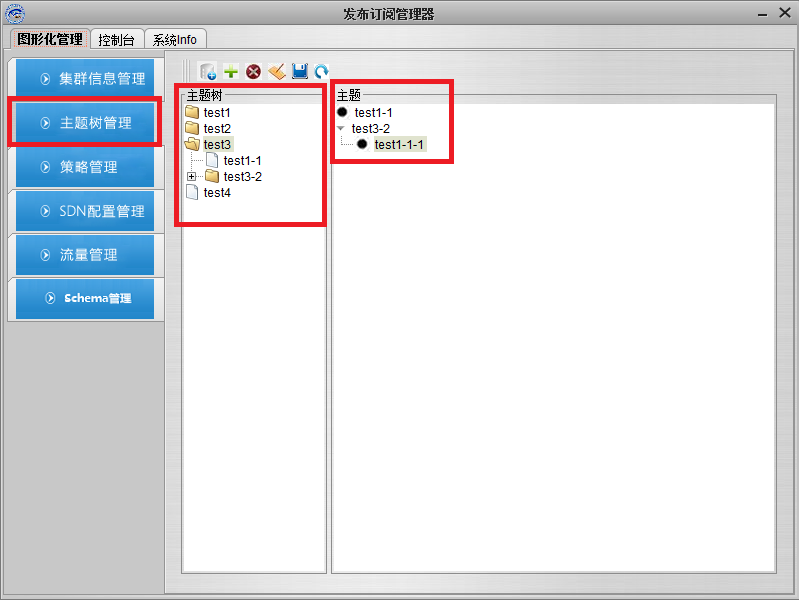


图2 主题树信息展示

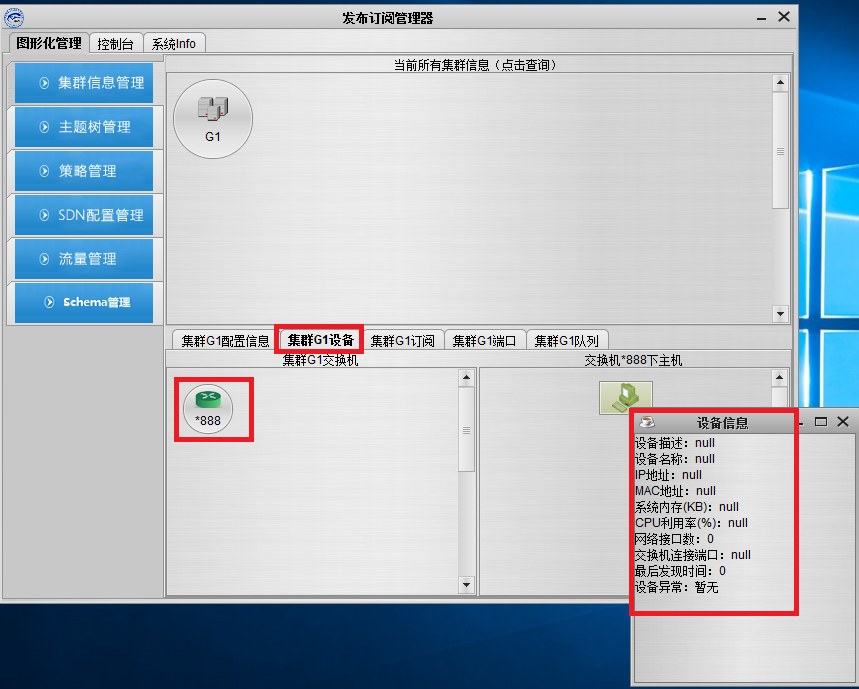


图3 集群详细信息展示

1. 新增邻居集群展示

|  |  |
| --- | --- |
| 测试项目 | 新增邻居集群展示 |
| 测试目的 | 验证当有新的集群连入系统后，能够在ui界面上展示新集群信息 |
| 测试步骤 | 1. 完成1.1 测试模块  2. 启动控制器，等待拓扑探测过程 |
| 预期结果 | 1. 管理员ui界面上显示新增集群名  2. 点击集群可以进一步查询集群内部详细信息，包括交换机名称、下属主机及其详细信息 |
| 实际结果 | 与预期相同，具体见下图：  4. 图4：新增集群展示  5. 图5：新增集群详细信息 |
| 结果分析 | 管理员ui界面系统运行正确，并能通过控制器定时上报的集群信息，更新界面并展示 |



图4 新增集群展示



图5 新增集群详细信息

1. 队列测试
2. 队列信息收集

|  |  |
| --- | --- |
| 测试项目 | 队列信息收集 |
| 测试目的 | 验证通过下发相应查询指令，可以获取交换机对外端口的队列信息 |
| 测试步骤 | 1. 通过ssh连接交换机系统  2. 遍历对外端口，下发查询队列语句  3. 查看获取的队列信息 |
| 预期结果 | 解析返回的查询结果，获得队列的进出字节数、速率等信息 |
| 实际结果 | 与预期相同，具体见下图：  6. 图6：队列信息查询结果 |
| 结果分析 | 能够获取到交换机的队列信息 |

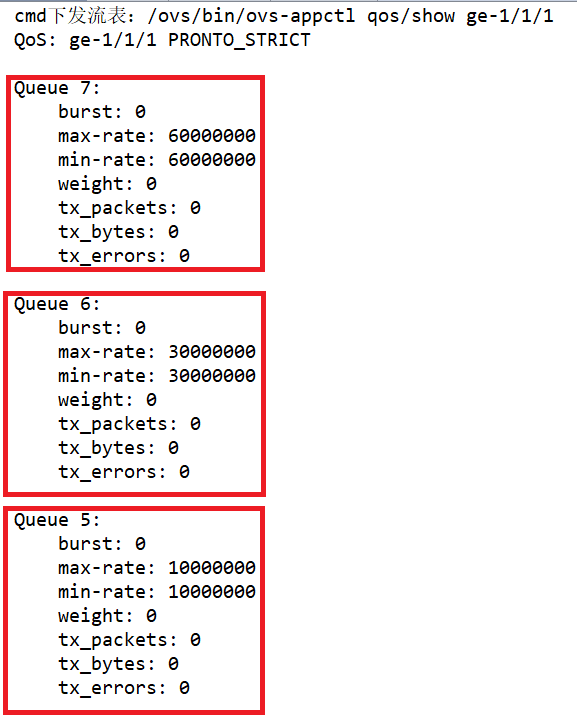


图6 队列信息查询结果

1. 队列信息处理

|  |  |
| --- | --- |
| 测试项目 | 队列信息处理 |
| 测试目的 | 验证当得到队列信息后，可以通过相关算法计算队列运行情况，并将运行结果上报给管理员 |
| 测试步骤 | 1. 完成1.2.1 测试模块  2. 查看集群上报信息 |
| 预期结果 | 经过计算、分析，控制器将队列信息封装传递给管理员 |
| 实际结果 | 与预期相同，具体见下图：  7. 图7：队列上报信息 |
| 结果分析 | 能够将队列分析的结果传递给管理员 |

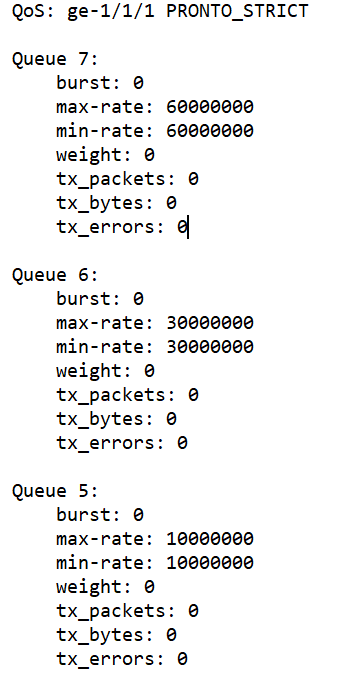


图7 队列上报信息

1. 路由测试
2. 管理路径生成

|  |  |
| --- | --- |
| 测试项目 | 管理路径生成 |
| 测试目的 | 验证当有新集群加入时，通过判断是否为管理员身份，进而计算管理路径 |
| 测试步骤 | 1. 启动G2 集群的控制器模块  2. 启动G3 集群的控制器模块  3. 启动G1 集群的管理员模块 |
| 预期结果 | 1. G2、G3通过拓扑发现后并没有计算管理路径  2. 当管理员G1 计入后，三个集群分别计算管理路径 |
| 实际结果 | 与预期相同，具体见下图：  8. 图8：管理路径计算结果 |
| 结果分析 | 管理路径是以管理员节点为根节点的树形结构，在计算时必须确保管理员角色在全网拓扑中，若不存在则无需计算，因此测试的整体过程为：  1. 当G2、G3通过拓扑发现感知到邻居节点的存在后，由于两者都为控制器角色，因此无需计算管理路径  2. 当G1加入拓扑后，由于存在了管理员身份，因此需要计算管理路径 |

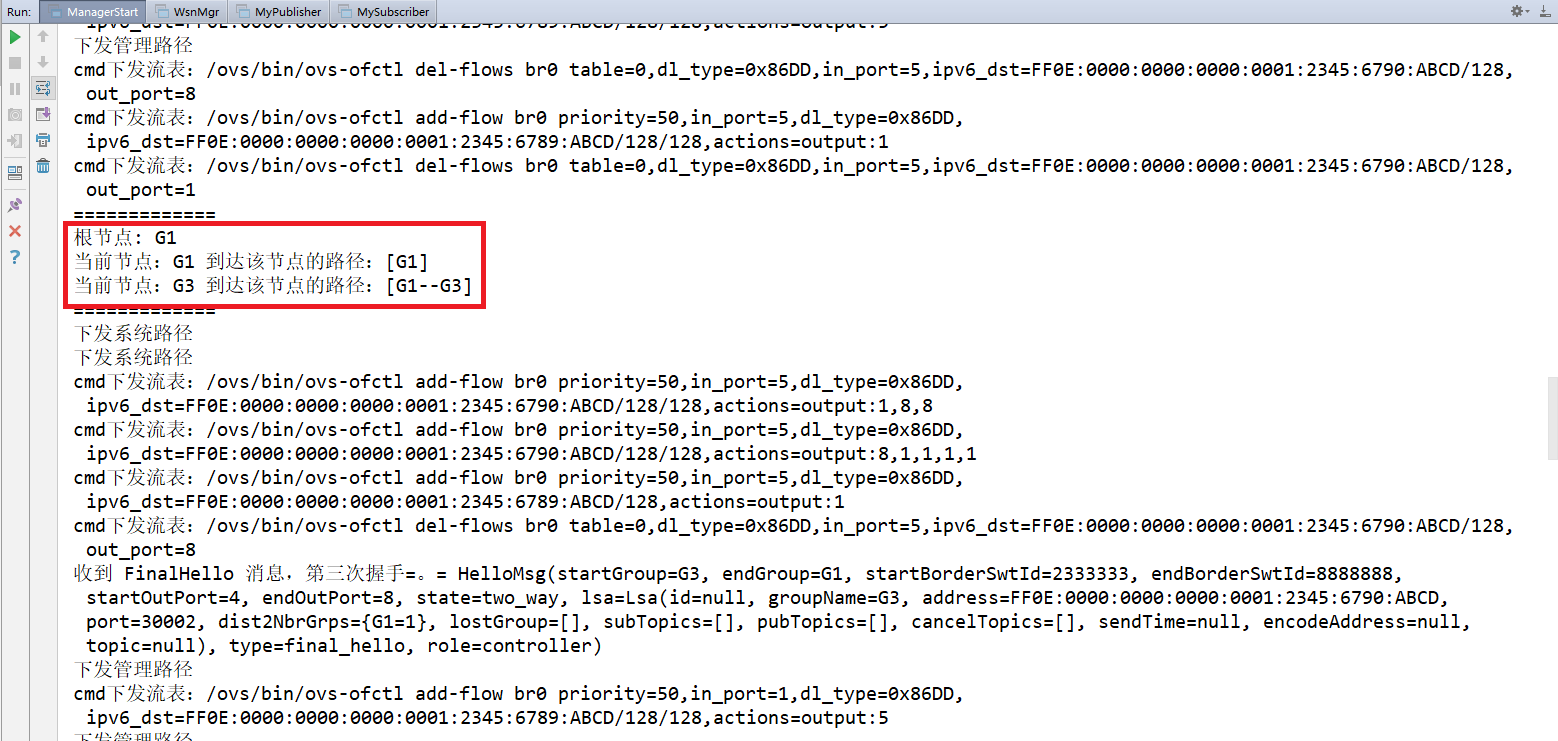


图8 管理路径计算结果

1. 主题路径生成

|  |  |
| --- | --- |
| 测试项目 | 主题路径生成 |
| 测试目的 | 验证当全网中存在多个发布订阅节点时，会计算相应的主题路径 |
| 测试步骤 | 1. 分别启动G1、G2、G3集群  2. 在集群G1 上开启针对主题“test1”的发布  3. 在集群G2 上开启针对主题“test1”的订阅  4. 在集群G3 上开启针对主题“test1”的订阅 |
| 预期结果 | 1. 在获知集群G1、G2针对相同主题存在发布订阅行为后，计算主题“test1”的主题路径  2. 集群G3 由于订阅了相同主题，计算新的主题路径 |
| 实际结果 | 与预期相同，具体见下图：  9. 图9：G1-G2 主题路径计算结果  10. 图10：G1-G2-G3 主题路径计算结果 |
| 结果分析 | 主题路径是根据相同主题计算的树形结构，采用的是steiner算法，每当有新节点加入或更改时都需要重新计算，因此测试的整体过程为：  1. 当集群G1、G2中存在相同主题的发布订阅行为后，计算主题路径  2. 新节点G3对相同主题有订阅行为，计算新的主题路径 |

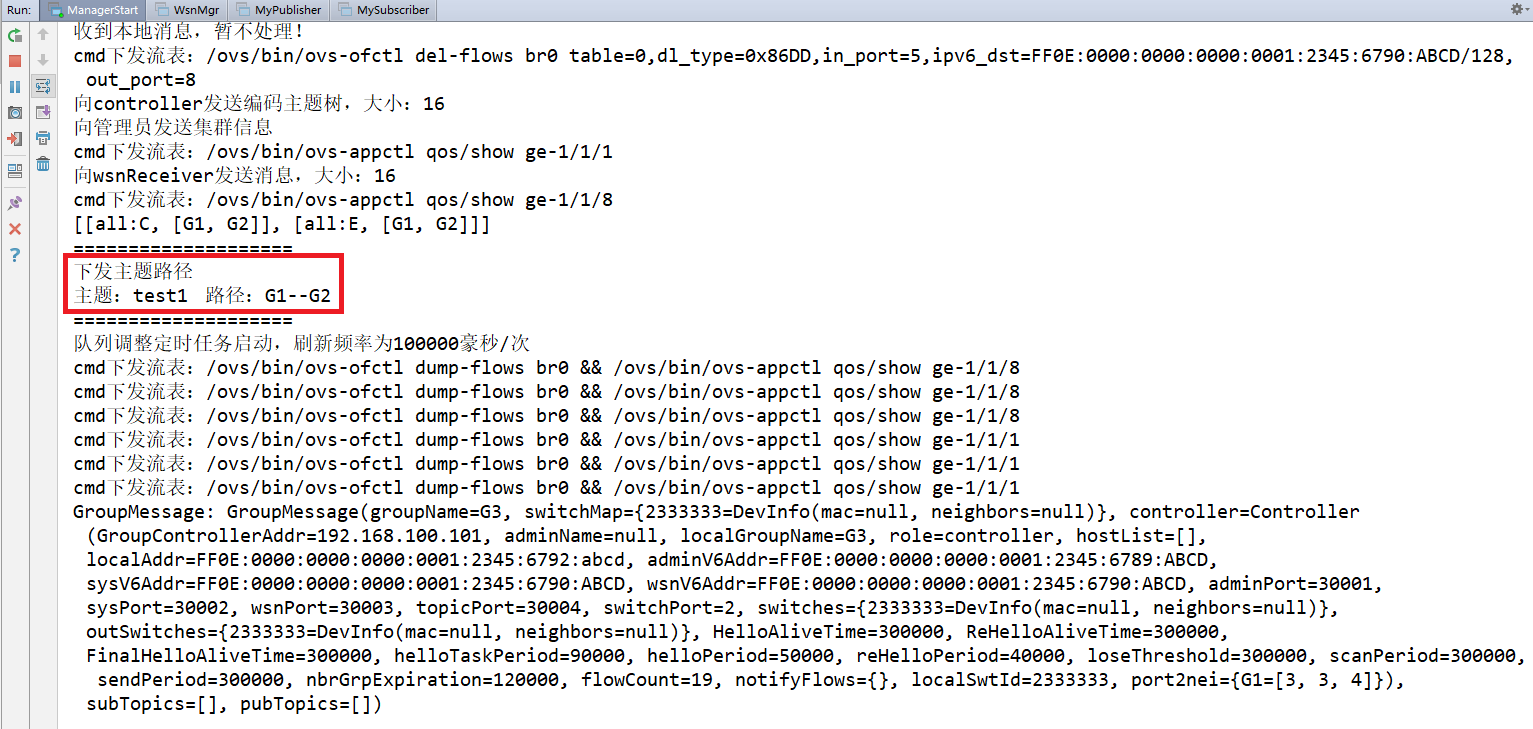


图9 G1-G2 主题路径计算结果

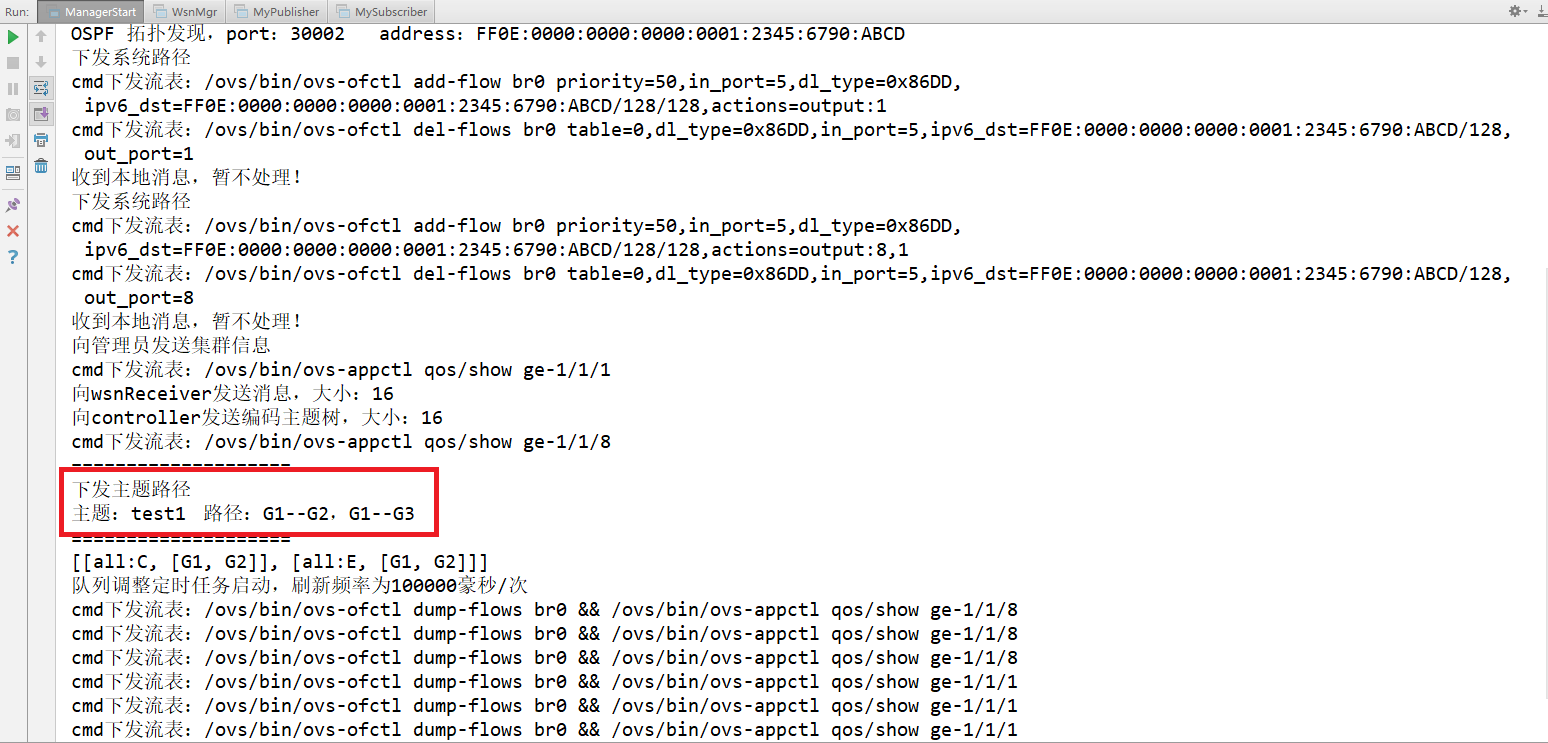


图10 G1-G2-G3 主题路径计算结果

1. 交换机测试
2. 流表生成

|  |  |
| --- | --- |
| 测试项目 | 流表生成 |
| 测试目的 | 验证计算出的（管理、主题）路径，可以转换为SDN交换机识别的流表形式 |
| 测试步骤 | 1. 完成1.3.2 测试模块  2. 查看流表打印信息 |
| 预期结果 | 计算出的主题路径是在集群G1、G3间针对主题“test1”的通路，具体表现为：  1. ipv6地址为“FF0E:0100:0000:0000:0000:0000:0000:0000”的消息可以从G1集群交换机的5、8端口进出；  2. ipv6地址为“FF0E:0100:0000:0000:0000:0000:0000:0000”的消息可以从G3集群交换机的2、3端口进出 |
| 实际结果 | 与预期相同，具体见下图：  11. 图11：G1流表生成结果  12：图12：G3流表生成结果 |
| 结果分析 | 计算出相应路径后，根据保存的拓扑信息，可以得到具体端口进出情况，从而封装ipv6地址、交换机进出端口、优先级，转换为SDN交换机识别的流表项 |

1. 流表下发

|  |  |
| --- | --- |
| 测试项目 | 流表下发 |
| 测试目的 | 验证转换后的流表项，可以下发各指定交换机 |
| 测试步骤 | 1. 使用ssh方式远程连接交换机  2. 完成1.4.1 测试模块  3. 查看交换机流表信息 |
| 预期结果 | Ssh远程连接上了指定的交换机，生成的流表项在交换机的流表中查找成功 |
| 实际结果 | 与预期相同，具体见下图：  13. 图13：G3交换机中流表内容 |
| 结果分析 | 通过指定交换机ip、端口、用户名、密码，可以远程操作交换机的cmd窗口，进一步能够下发流表项，从而达到流表下发的目的 |



图13 G3交换机中流表内容

1. 流表维护

|  |  |
| --- | --- |
| 测试项目 | 流表维护 |
| 测试目的 | 验证本集群内对下发的流表进行了维护，当有新的流表项加入时会自动保存 |
| 测试步骤 | 1. 完成1.4.2 测试模块  2. 继续生成流表并下发  3. 查看集群内的流表信息 |
| 预期结果 | 集群内对流表项进行了统一管理，整理为流表信息 |
| 实际结果 | 与预期相同，具体见下图：  14. 图14：内存中流表内容 |
| 结果分析 | 由于交换机只识别单一流表项，因此需要对具有相同ipv6转发地址、进端口、优先级的流表项进行合并整合，这就需要在程序中维护一份和交换机相同的流表信息 |

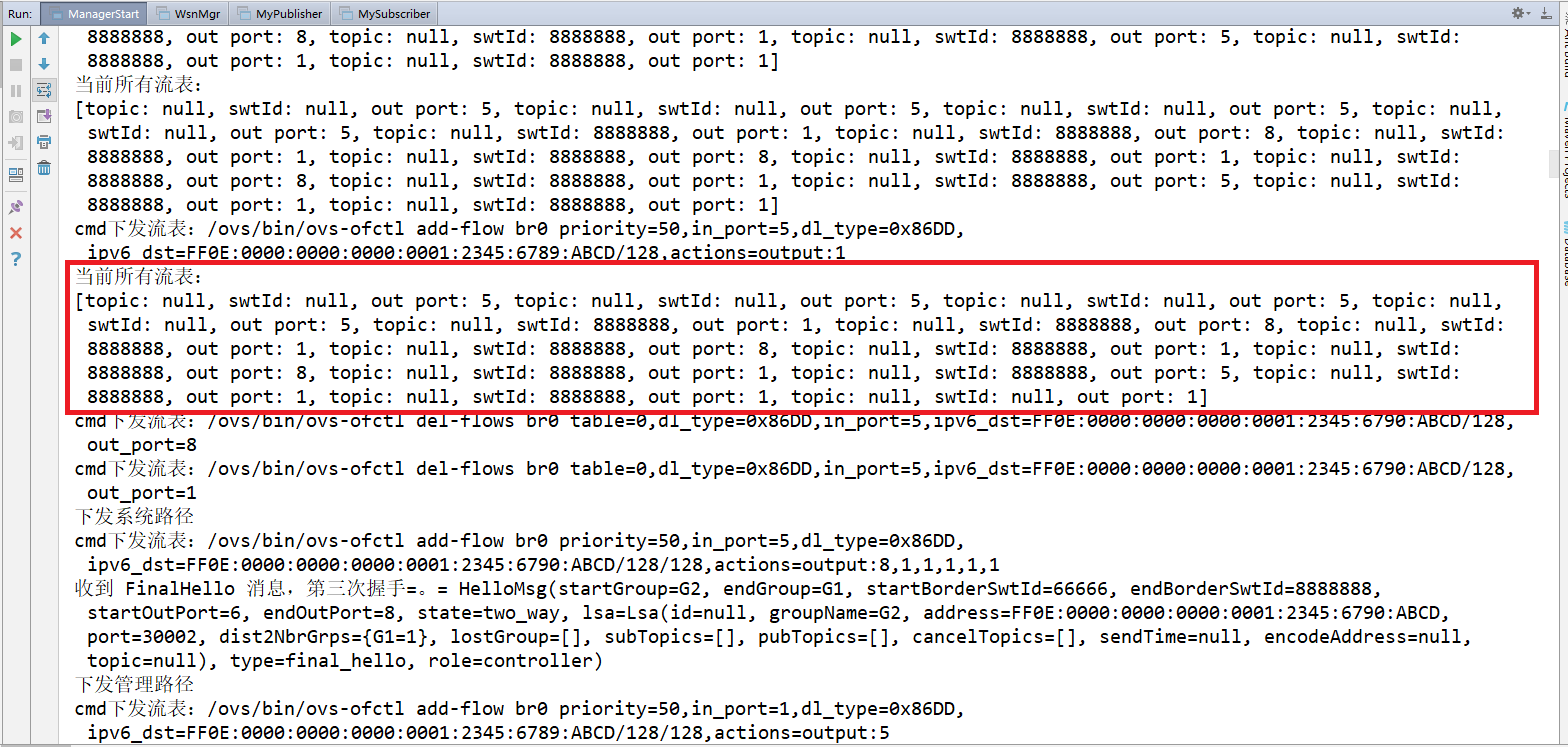


图14 内存中流表内容

1. 主题树测试
2. 主题树获取

|  |  |
| --- | --- |
| 测试项目 | 主题树获取 |
| 测试目的 | 验证能够通过xml文件获取到主题树信息 |
| 测试步骤 | 1. 根据文件名访问主题树文件  2. 获取主题树信息 |
| 预期结果 | 打印出完整的主题树信息 |
| 实际结果 | 与预期相同，具体见下图：  15. 图15：主题树获取结果 |
| 结果分析 | 构建出统一的虚拟根节点，通过dom4j方式读取xml内容，得到主题树句柄 |

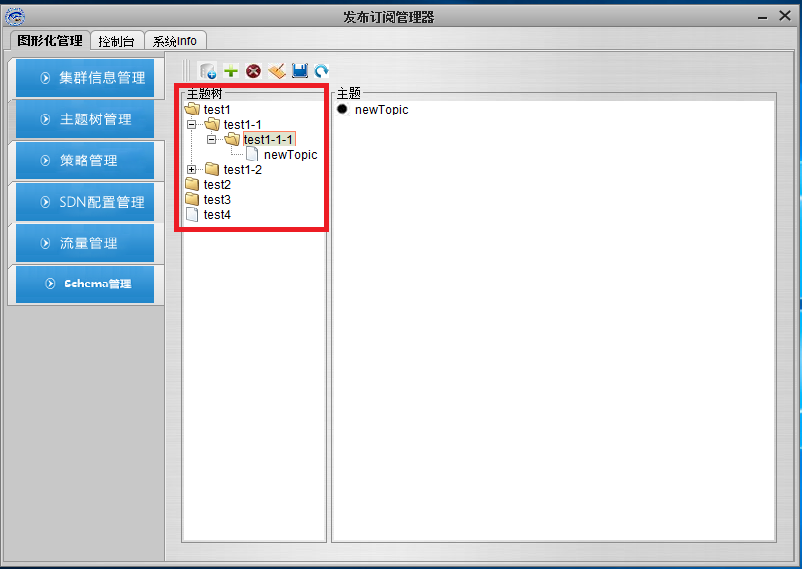


图15 主题树获取结果

1. 主题树修改

|  |  |
| --- | --- |
| 测试项目 | 主题树修改 |
| 测试目的 | 验证针对主题的修改能够生效，并写回到配置文件中 |
| 测试步骤 | 1. 完成1.5.1 测试模块  2. 添加“新建节点”主题  3. 重新读取主题树，并查看输出内容 |
| 预期结果 | 主题“test1”的内容被修改，重新打印的主题信息为修改后的主题树 |
| 实际结果 | 与预期相同，具体见下图：  16. 图16：主题树修改结果 |
| 结果分析 | 主题树保存在xml文件中，每次读取时采用dom4j方式，得到主题树句柄；若产生修改，则动态写回文件中，这样保证数据的准确性 |

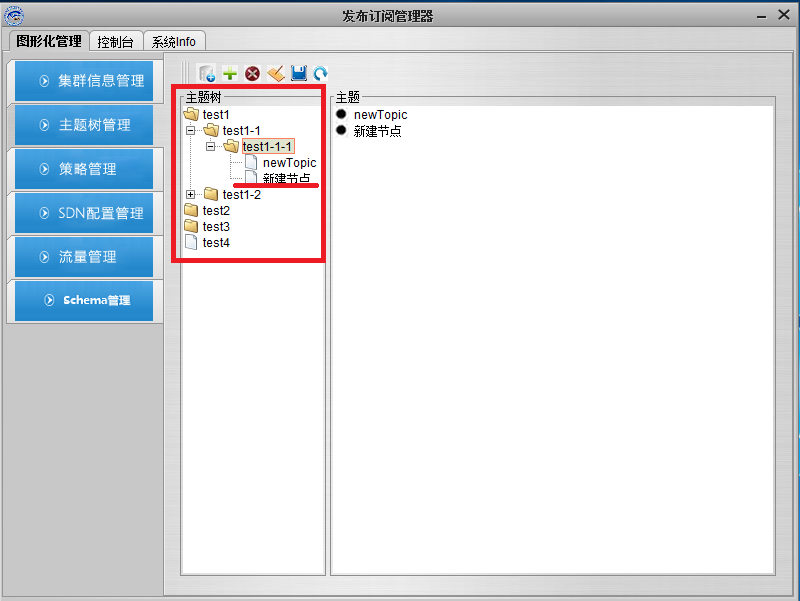


图16 主题树修改结果

1. 拓扑测试
2. 拓扑构建

|  |  |
| --- | --- |
| 测试项目 | 拓扑构建 |
| 测试目的 | 验证当新增邻居集群时，会根据ospf协议进行三次握手，从而建立邻居关系 |
| 测试步骤 | 1. 启动集群G1  2. 启动集群G2 |
| 预期结果 | 集群间进行了三次握手，最终建立邻居关系 |
| 实际结果 | 与预期相同，具体见下图：  17. 图17：G1三次握手过程  18. 图18：G2三次握手过程 |
| 结果分析 | 拓扑模块采用的是ospf协议，控制器会定时向对外端口发送Hello消息，收到Hello消息的集群建立单向连接关系，并反向发送ReHello消息，初始集群收到后建立双向连接关系，至此，拓扑构建过程结束 |

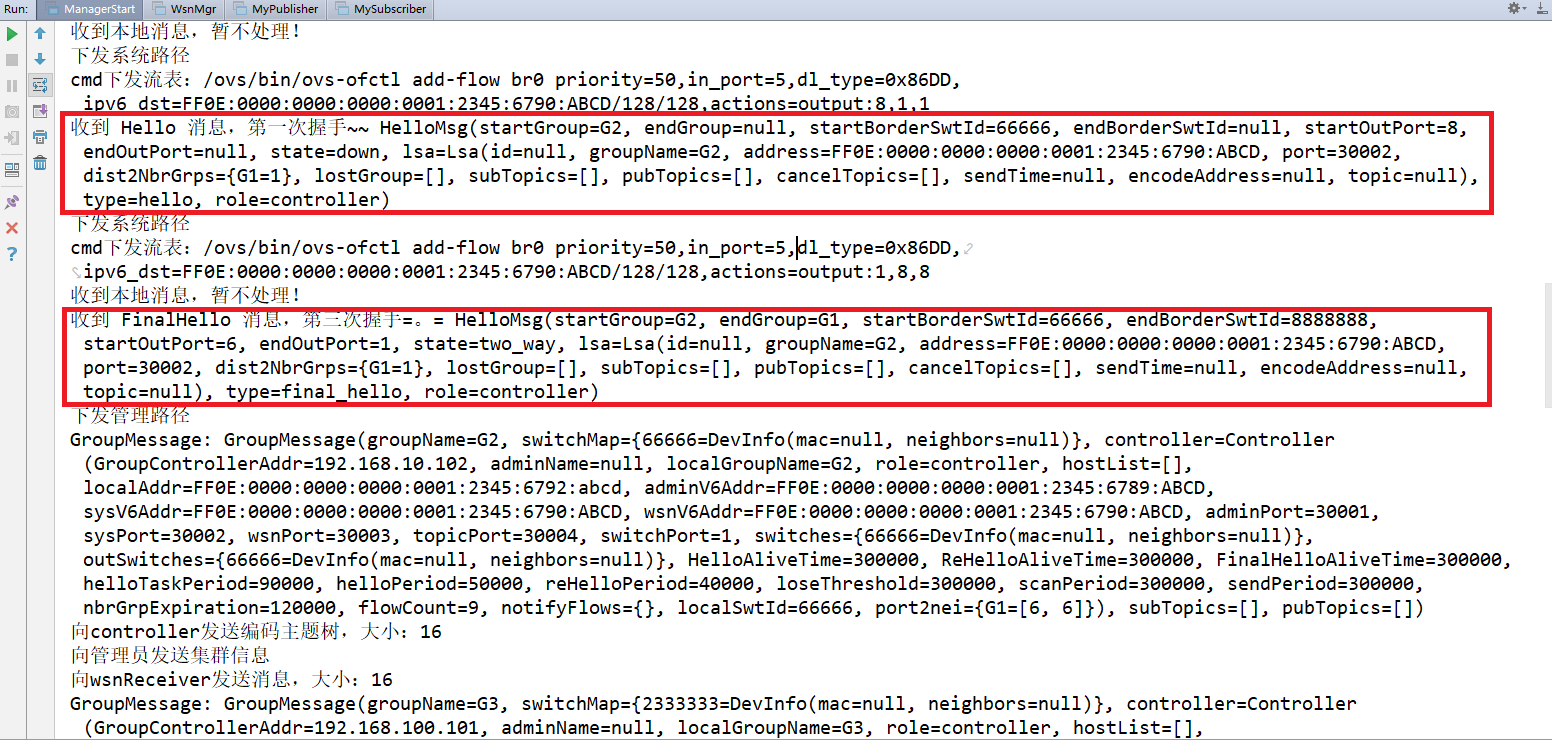


图17 G1三次握手过程

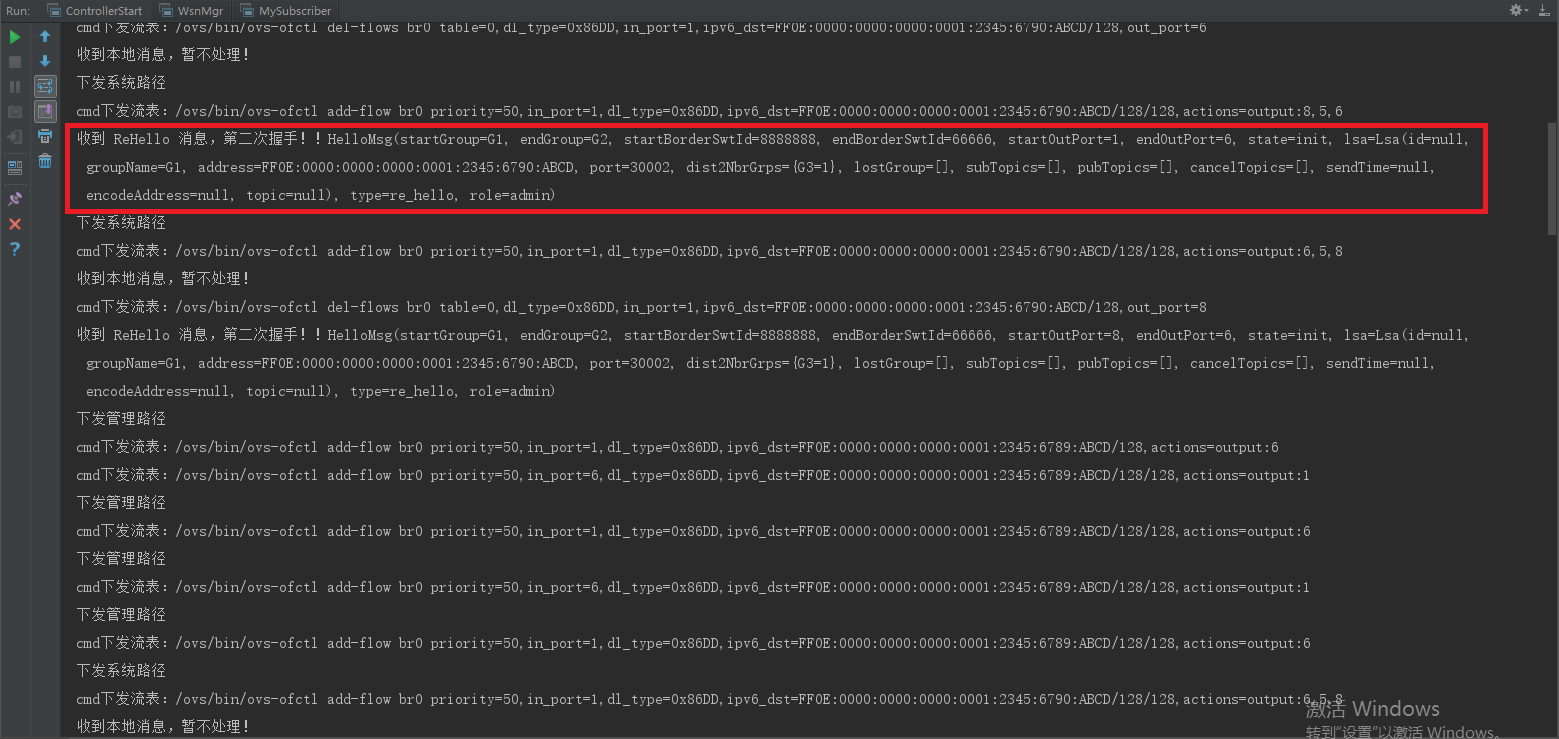


图18 G2三次握手过程

1. 拓扑维护

|  |  |
| --- | --- |
| 测试项目 | 拓扑维护 |
| 测试目的 | 验证邻居集群可以通过lsa消息共享拓扑信息，从而完善全局拓扑 |
| 测试步骤 | 1. 启动集群G1  2. 启动集群G2  3. 启动集群G3 |
| 预期结果 | 1. 集群G1、G2间由于直接相邻，经过三次握手后建立双向连接关系  2. 集群G1、G3间由于直接相邻，经过三次握手后建立双向连接关系  3. 集群G1、G2间交换lsa消息，G2获知了G3集群的存在；  同样，集群G1、G3间交换lsa消息，G3获知了G1集群的存在 |
| 实际结果 | 与预期相同，具体见下图：  19. 图19：G1广播lsa内容 |
| 结果分析 | 拓扑模块采用的是ospf协议，邻居集群间定期交换lsa信息，lsa构建出的lsdb保存了全局统一的拓扑，从而各个集群才能计算出相同的全网拓扑信息 |

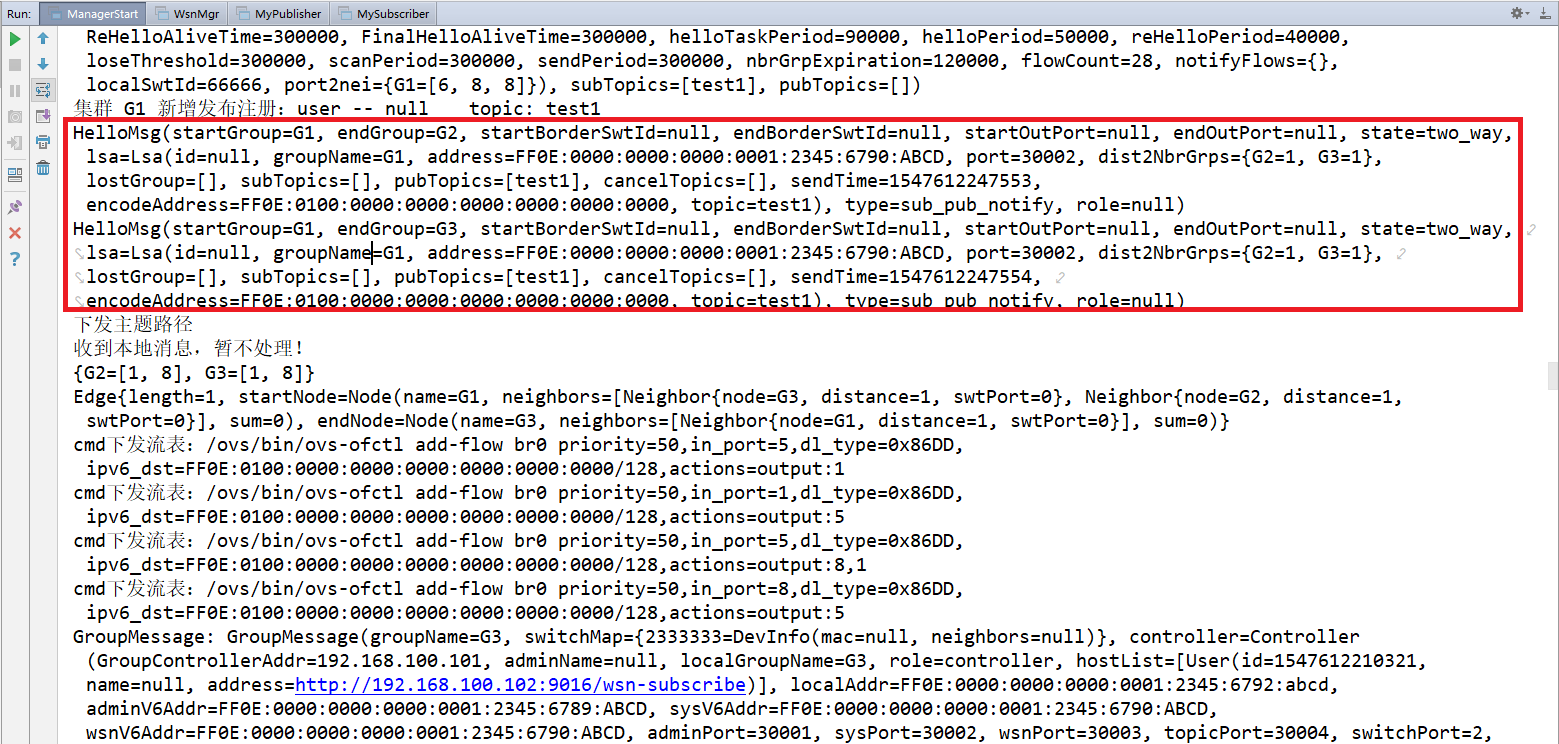


图19 G1广播lsa内容

1. Wsn模块测试
2. 接收主题树

|  |  |
| --- | --- |
| 测试项目 | 接收主题树 |
| 测试目的 | 验证当启动wsn模块后，可以接收并保存来自控制器的主题树信息 |
| 测试步骤 | 1. 启动管理员模块  2. 启动控制器模块  3. 启动控制器集群内的wsn模块 |
| 预期结果 | 1. 在启动控制器模块后，收到管理员下发的主题树信息  2. 启动wsn模块后，收到控制器下发的wsn信息 |
| 实际结果 | 与预期相同，具体见下图：  20. 图20：wsn收到的主题树信息 |
| 结果分析 | Wsn模块与控制器之间采取的是ipv6组播的方式，由于监听了相同的wsn组播地址，因此可以接收到控制器下发的主题树信息 |

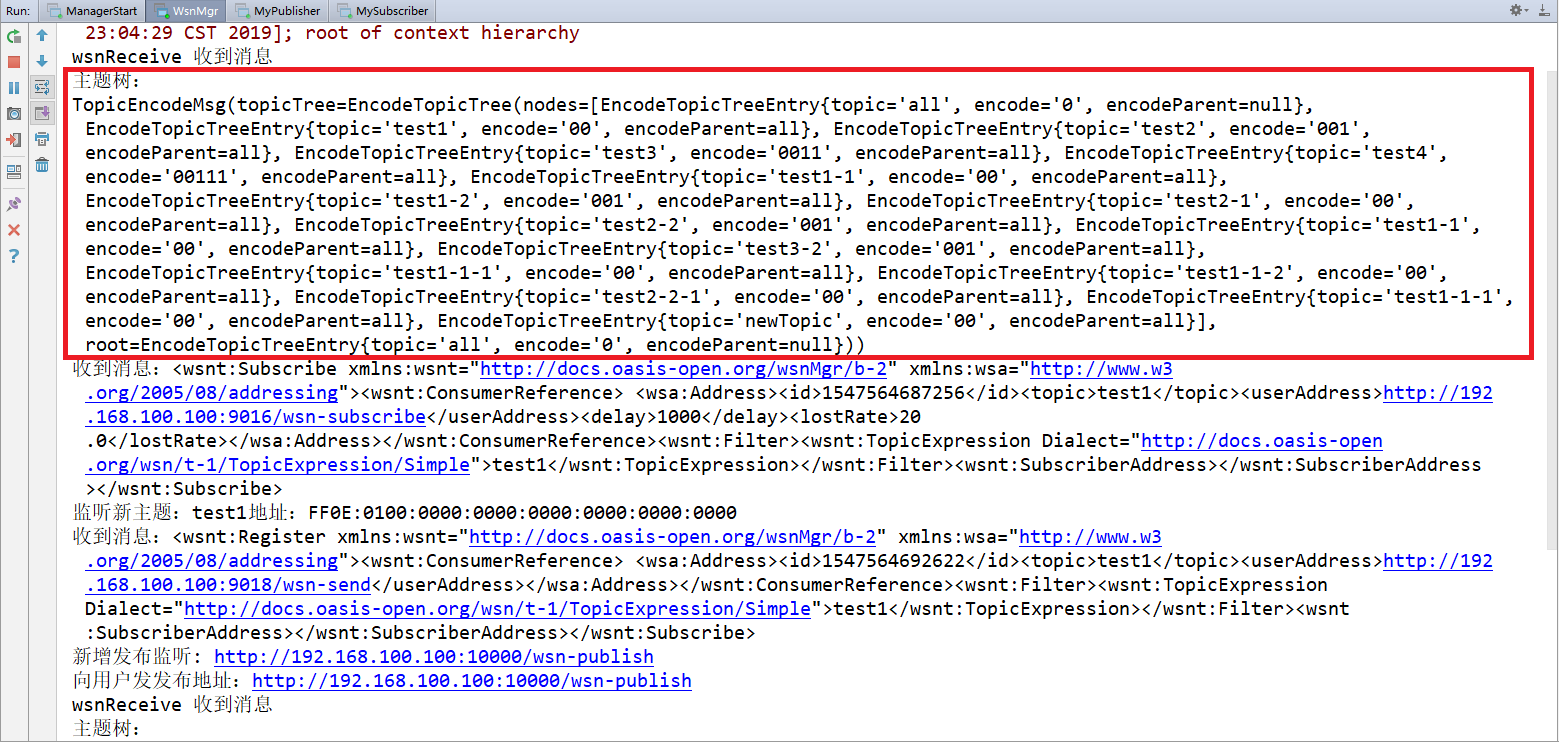


图20 wsn收到的主题树信息

1. 接收集群内用户发布订阅消息

|  |  |
| --- | --- |
| 测试项目 | 接收集群内用户发布订阅消息 |
| 测试目的 | 验证当wsn模块启动后，可以接收到下属用户的发布订阅请求 |
| 测试步骤 | 1. 启动wsn模块  2. 启动订阅用户，触发订阅请求  3. 启动发布用户，触发发布请求 |
| 预期结果 | 1. wsn收到用户的订阅请求  2. wsn收到用户的发布请求 |
| 实际结果 | 与预期相同，具体见下图：  21. 图21：wsn收到用户订阅请求  22. 图22：wsn收到用户发布请求 |
| 结果分析 | Wsn模块与用户之间采取的是ws传输方式，因此wsn需要预先发布服务，用户向指定的服务地址发送请求，这样wsn模块在收到用户请求后根据字段分类处理 |

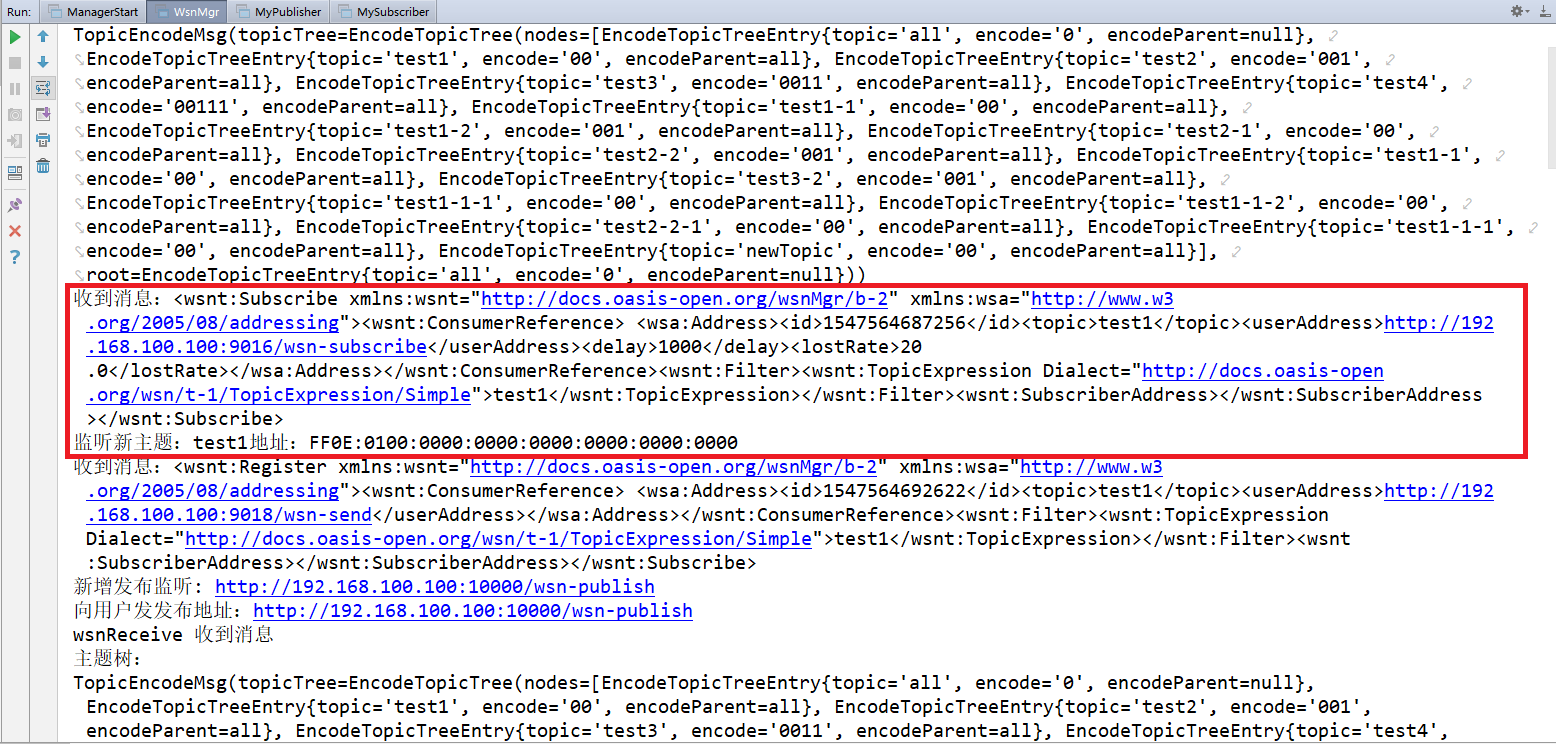


图21 wsn收到用户订阅请求



图22 wsn收到用户发布请求

1. 系统性能测试

本文所论述的性能指标主要有文本传输和视频传输两个方面，而评价标准包括传输时延、传输过程中的丢包率等。这两者侧重点不同，在现实生活中，人们对于文本传输更加强调准确性，即重视丢包率；而视频实时播放时，人们更加希望获得低时延的服务。因此，针对这两者间的不同，下面将通过测试结果分别阐述。

测试环境：G2 – G1 – G3，G1发布，G2、G3接收

1. 文本传输
   1. 不使用队列、线程池

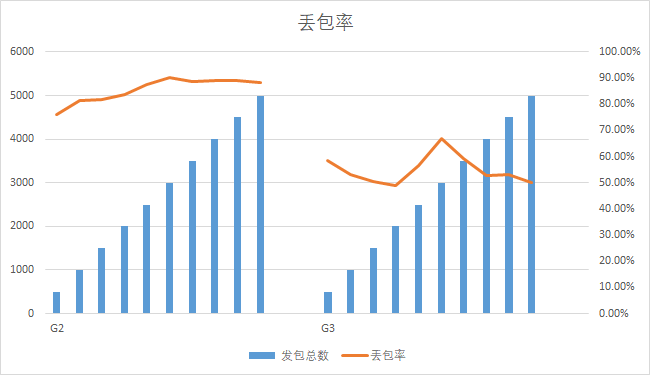


图23 丢包率

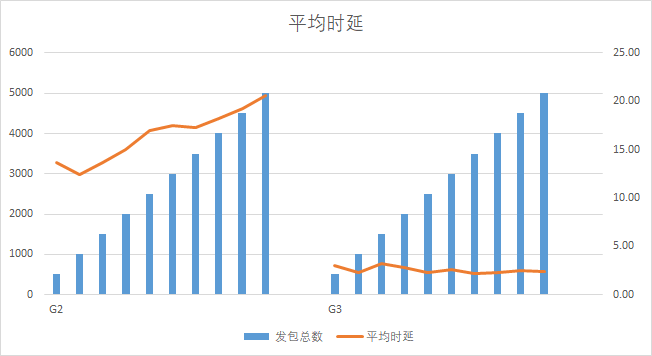


图24 平均时延

* 1. 使用主队列缓存

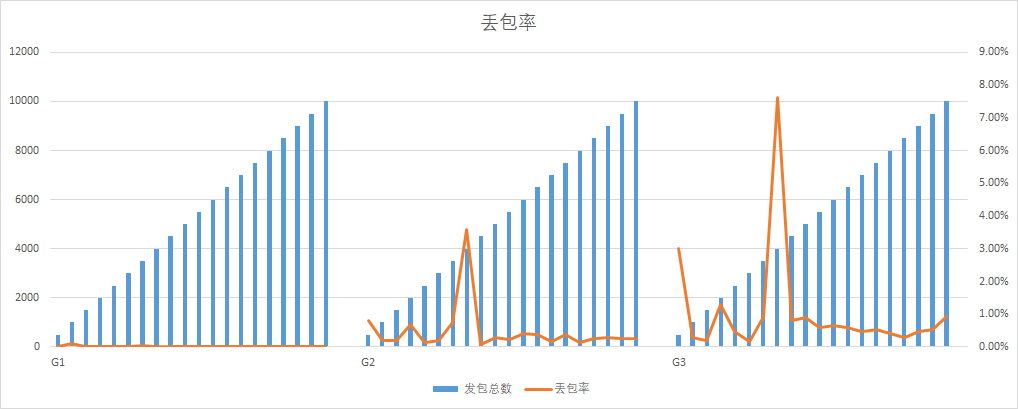


图25 丢包率

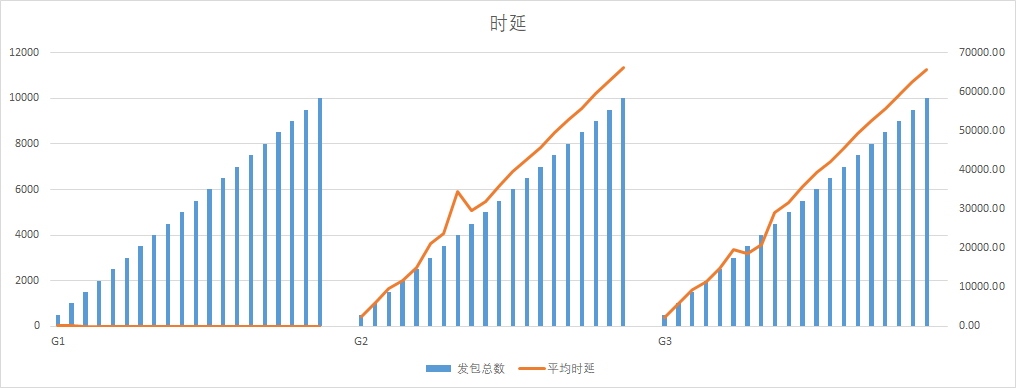


图26 平均时延

* 1. 主队列 + 子队列 + 负载均衡

//

1. 视频传输
   1. 不使用队列、线程池
   2. 使用主队列缓存
   3. 主队列 + 子队列 + 负载均衡