REST APIs 研究

作者:冀烨　　QQ：35317626

OpenDaylight SDN 研究群

@南京-冀烨

1 Topology REST APIs

在 org.opendaylight.controller.topology.northbound 包中。

Topology Northbound REST API

Authentication scheme : HTTP Basic，Authentication realm : opendaylight，Transport : HTTP and HTTPS。默认使用 HTTPS Authentication

1.1 Topology REST APIs 可提供的服务

客户端请求	**服务器响应**
GET：检测拓扑	topology
GET：检测用户配置连接	list
PUT：添加一个用户连接	topologyUserLinkConfig

1.2 Topology REST APIs 数据模型

其 REST 数据元素及相应的 XML 配置细节如表 4.2-1 所示。

Data Elements	**Data Types**	**XML Elements**	
		名称（类型）	**最大/最小 出现**
edge	edge	tailNodeConnector（nodeConnector）	0/1
		headNodeConnector（nodeConnector）	0/1
edgeProperties	edgeProperties	edge (edge)	0/1
		properties/property	0/unbounded
list	topologyUserLinks	userLinks (topologyUserLinkConfig)	0/unbounded
node	node	type (string)	0/1
		id (string)	0/1
nodeConnector	nodeConnector	type (string)	0/1
		id (string)	0/1
		node (node)	0/1
property	property		
topology	topology	edgeProperties (edgeProperties)	0/unbounded
topologyUserLinkConfig	topologyUserLinkConfig	status (string)	0/1
		srcNodeConnector (string)	0/1
		dstNodeConnector (string)	0/1
		name (string)	0/1

表 4.2-1 Topology REST APIs

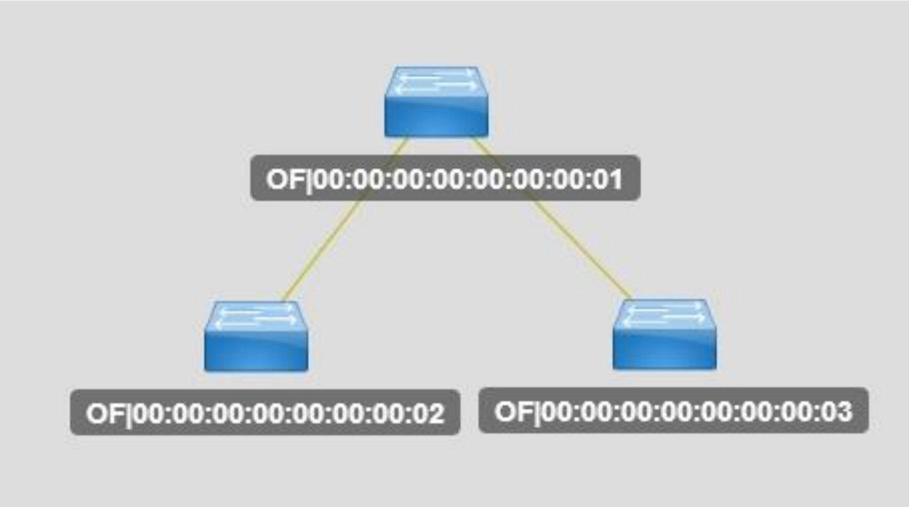
其中，Topology 的基本单元是：节点 node，node 的标识包括 id 和类型，头节点和尾节点连接器的连线组成边 edge，edge 和多个 property 的组合构成了具有实际属性的 edgeProperties 即有了一个具有实际特性的连接，若干 edgeProperties 的组合构成了 Topology。

另外该 APIs 提供了 TopologyUserLinkConfig 组成的 list，存储的是用户连接的配置信息。

UserLinks 功能有待考证。

1.3 实验分析

1.3.1 实验拓扑

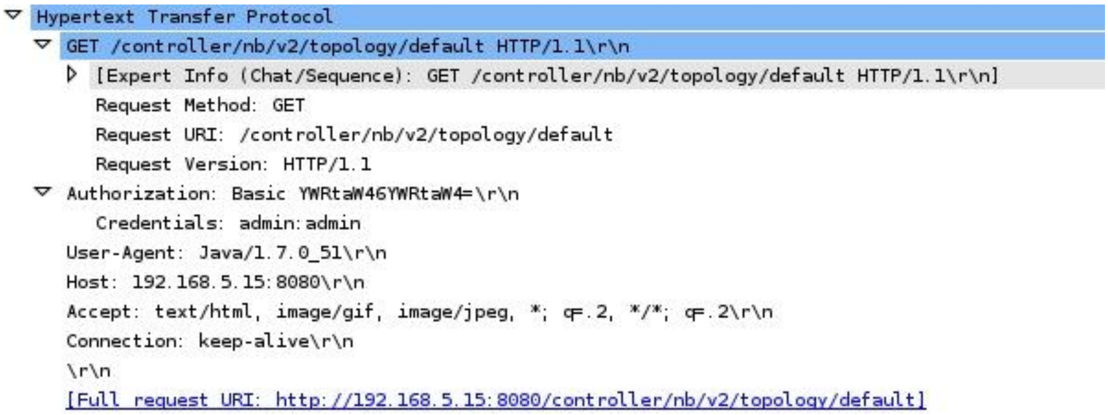

OF|00:00:00:00:00:00:00:01
OF|00:00:00:00:00:00:00:02
OF|00:00:00:00:00:00:00:03

该拓扑是通过 mininet 进行创建，即用 mininet 来模拟 SDN 网络，拓扑创建命令如下：

mn --topo tree,depth=2,fanout=2 --controller=remote,ip=xxx.xxx.xxx.xxx,port=6633

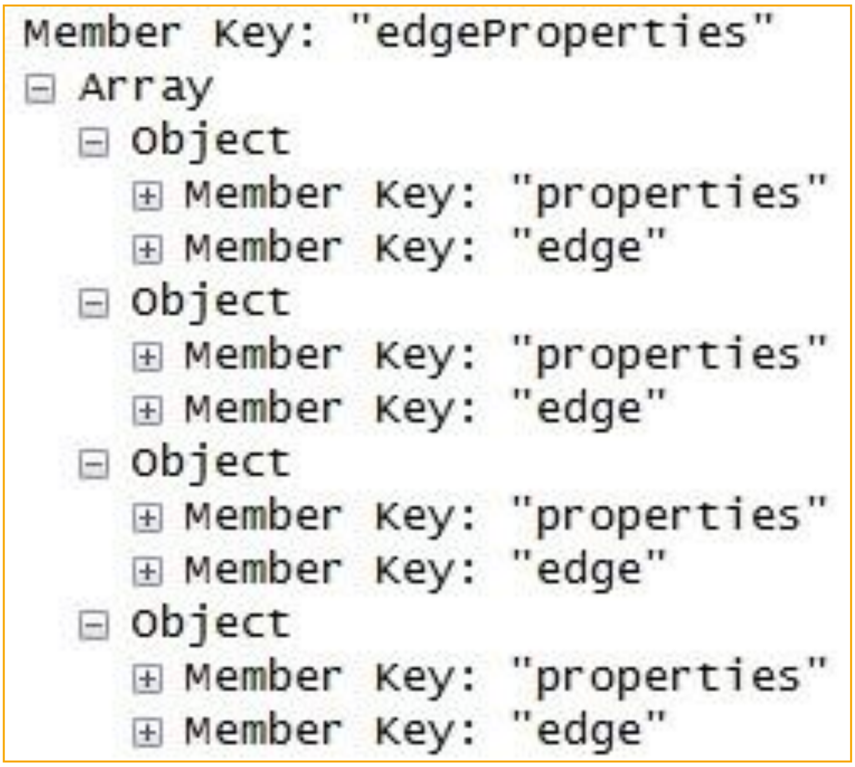
1.3.2 实验抓包情况

1）获取拓扑

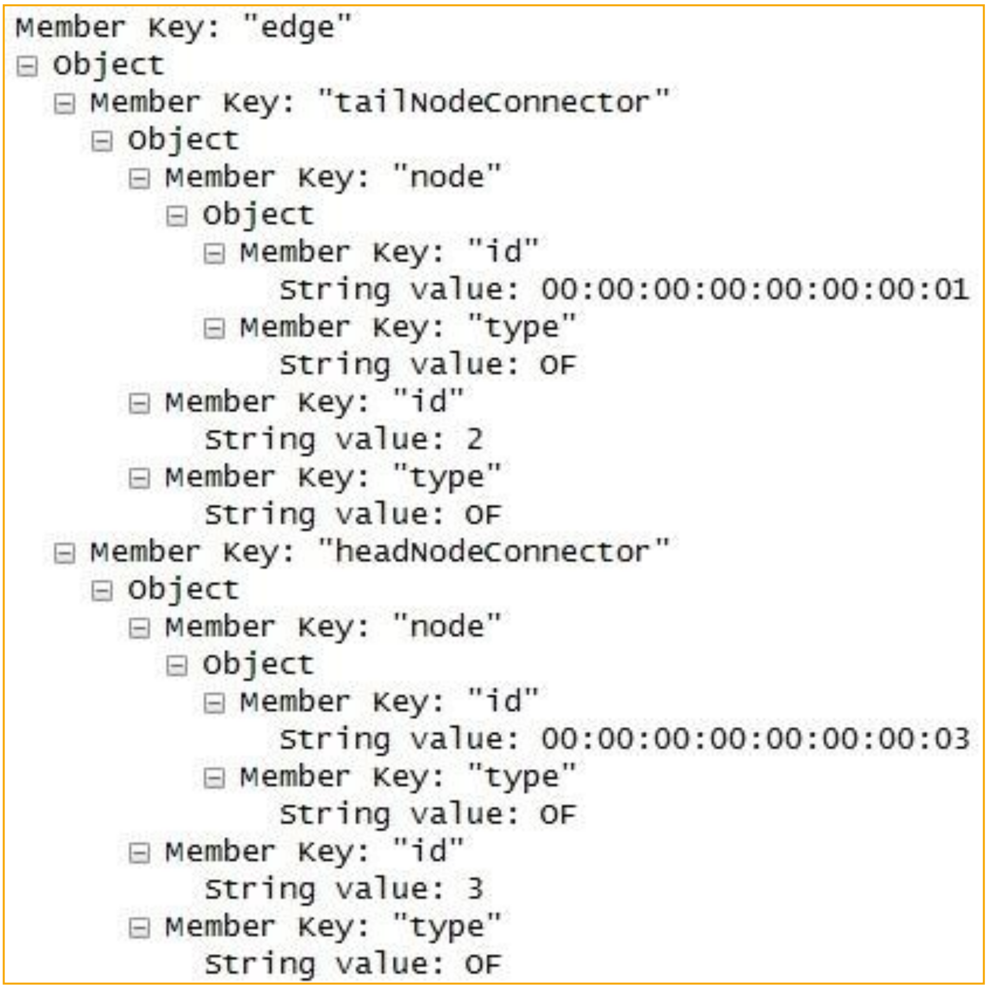


```
▽ Hypertext Transfer Protocol
  ▽ GET /controller/nb/v2/topology/default HTTP/1.1\r\n
    ▷ [Expert Info (Chat/Sequence): GET /controller/nb/v2/topology/default HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /controller/nb/v2/topology/default
      Request Version: HTTP/1.1
  ▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
      Credentials: admin:admin
    User-Agent: Java/1.7.0_51\r\n
    Host: 192.168.5.15:8080\r\n
    Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://192.168.5.15:8080/controller/nb/v2/topology/default]
```

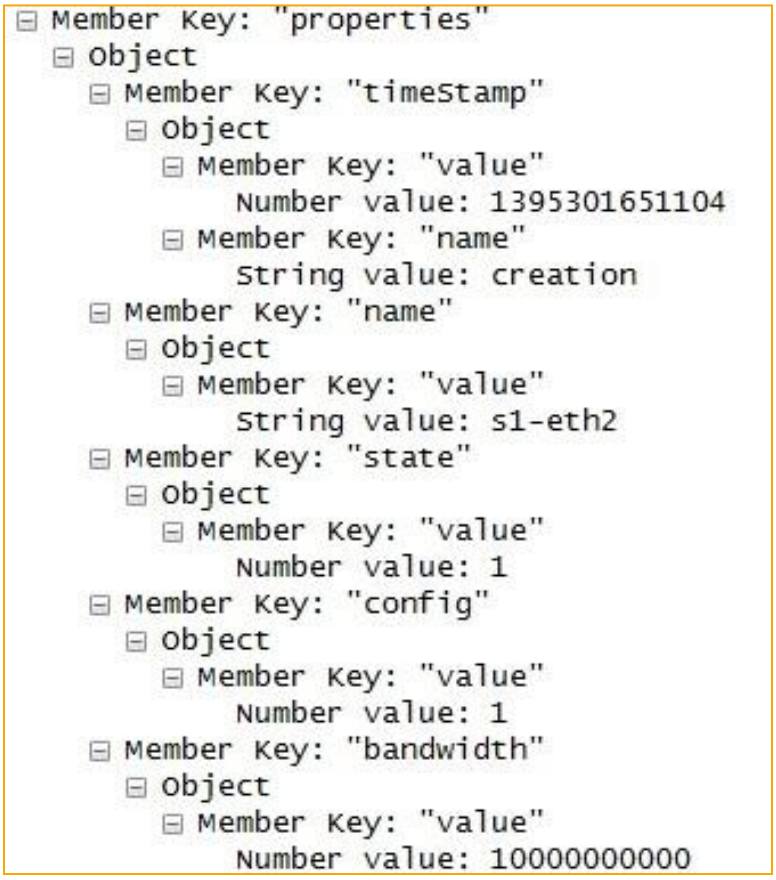
客户端请求 topology 信息



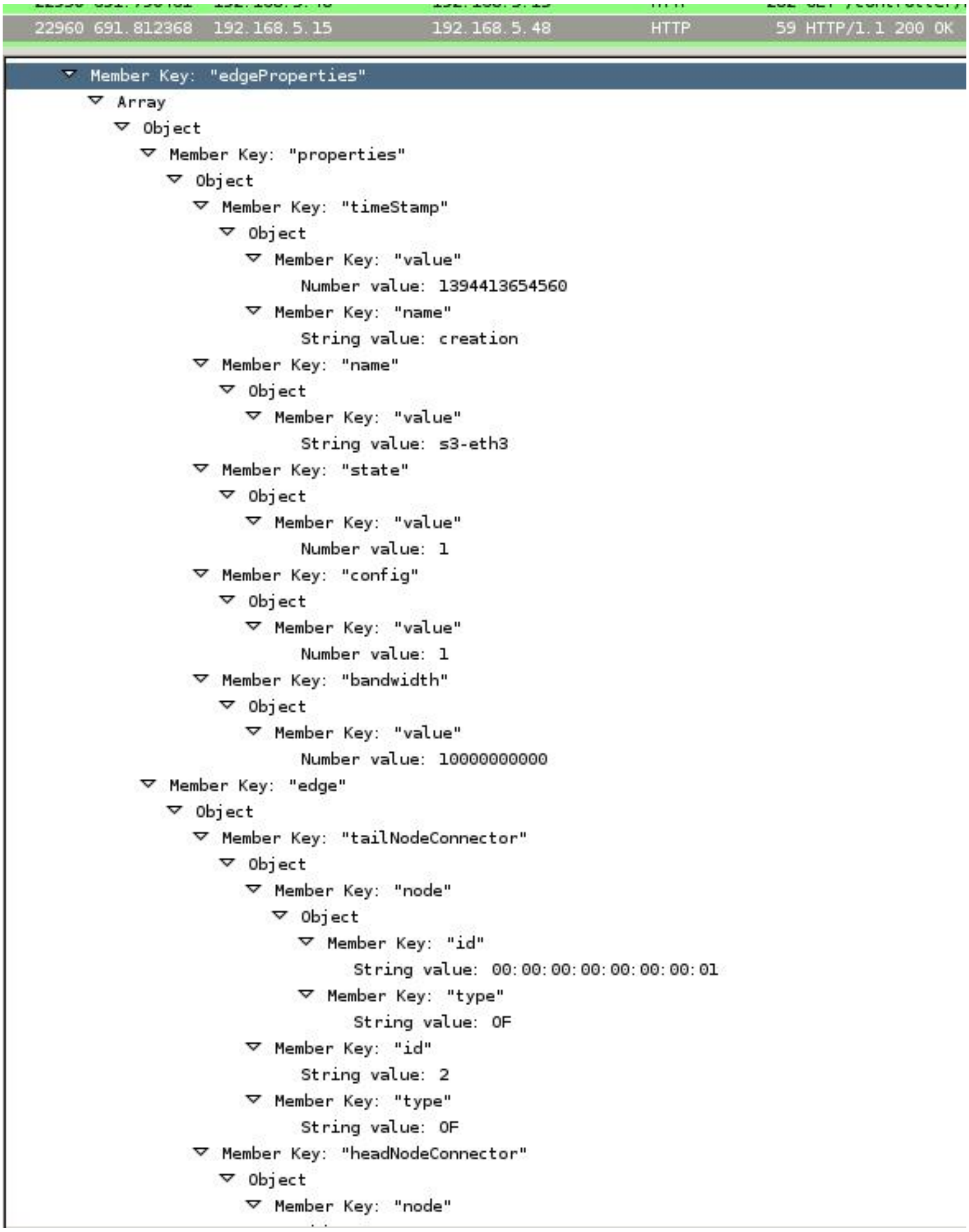
```
Member Key: "edgeProperties"
⊟ Array
   ⊟ Object
      ⊞ Member Key: "properties"
      ⊞ Member Key: "edge"
   ⊟ Object
      ⊞ Member Key: "properties"
      ⊞ Member Key: "edge"
   ⊟ Object
      ⊞ Member Key: "properties"
      ⊞ Member Key: "edge"
   ⊟ Object
      ⊞ Member Key: "properties"
      ⊞ Member Key: "edge"
```



```
Member Key: "edge"
⊟ Object
   ⊟ Member Key: "tailNodeConnector"
      ⊟ Object
         ⊟ Member Key: "node"
            ⊟ Object
               ⊟ Member Key: "id"
                    String value: 00:00:00:00:00:00:00:01
               ⊟ Member Key: "type"
                    String value: OF
         ⊟ Member Key: "id"
              String value: 2
         ⊟ Member Key: "type"
              String value: OF
   ⊟ Member Key: "headNodeConnector"
      ⊟ Object
         ⊟ Member Key: "node"
            ⊟ Object
               ⊟ Member Key: "id"
                    String value: 00:00:00:00:00:00:00:03
               ⊟ Member Key: "type"
                    String value: OF
         ⊟ Member Key: "id"
              String value: 3
         ⊟ Member Key: "type"
              String value: OF
```



```
⊟ Member Key: "properties"
  ⊟ Object
    ⊟ Member Key: "timeStamp"
      ⊟ Object
        ⊟ Member Key: "value"
            Number value: 1395301651104
        ⊟ Member Key: "name"
            String value: creation
    ⊟ Member Key: "name"
      ⊟ Object
        ⊟ Member Key: "value"
            String value: s1-eth2
    ⊟ Member Key: "state"
      ⊟ Object
        ⊟ Member Key: "value"
            Number value: 1
    ⊟ Member Key: "config"
      ⊟ Object
        ⊟ Member Key: "value"
            Number value: 1
    ⊟ Member Key: "bandwidth"
      ⊟ Object
        ⊟ Member Key: "value"
            Number value: 10000000000
```



```
22960 691.812368  192.168.5.15        192.168.5.48        HTTP        59 HTTP/1.1 200 OK

▽ Member Key: "edgeProperties"
  ▽ Array
    ▽ Object
      ▽ Member Key: "properties"
        ▽ Object
          ▽ Member Key: "timeStamp"
            ▽ Object
              ▽ Member Key: "value"
                  Number value: 1394413654560
              ▽ Member Key: "name"
                  String value: creation
          ▽ Member Key: "name"
            ▽ Object
              ▽ Member Key: "value"
                  String value: s3-eth3
          ▽ Member Key: "state"
            ▽ Object
              ▽ Member Key: "value"
                  Number value: 1
          ▽ Member Key: "config"
            ▽ Object
              ▽ Member Key: "value"
                  Number value: 1
          ▽ Member Key: "bandwidth"
            ▽ Object
              ▽ Member Key: "value"
                  Number value: 10000000000
      ▽ Member Key: "edge"
        ▽ Object
          ▽ Member Key: "tailNodeConnector"
            ▽ Object
              ▽ Member Key: "node"
                ▽ Object
                  ▽ Member Key: "id"
                      String value: 00:00:00:00:00:00:00:01
                  ▽ Member Key: "type"
                      String value: OF
              ▽ Member Key: "id"
                  String value: 2
              ▽ Member Key: "type"
                  String value: OF
          ▽ Member Key: "headNodeConnector"
            ▽ Object
              ▽ Member Key: "node"
```

服务器应答 topology 信息

拓扑信息由连接组成，该拓扑的连接组成有：

(OF|2@OF|00:00:00:00:00:00:00:01->OF|3@OF|00:00:00:00:00:00:00:03)

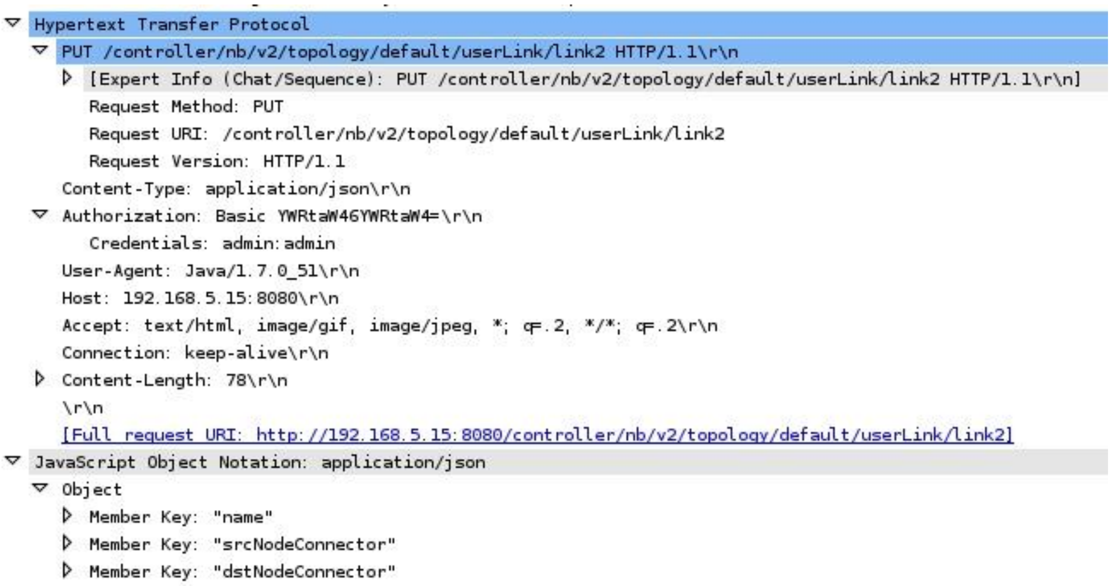
(OF|3@OF|00:00:00:00:00:00:00:03->OF|2@OF|00:00:00:00:00:00:00:01)

(OF|3@OF|00:00:00:00:00:00:00:02->OF|1@OF|00:00:00:00:00:00:00:01)

(OF|1@OF|00:00:00:00:00:00:00:01->OF|3@OF|00:00:00:00:00:00:00:02)

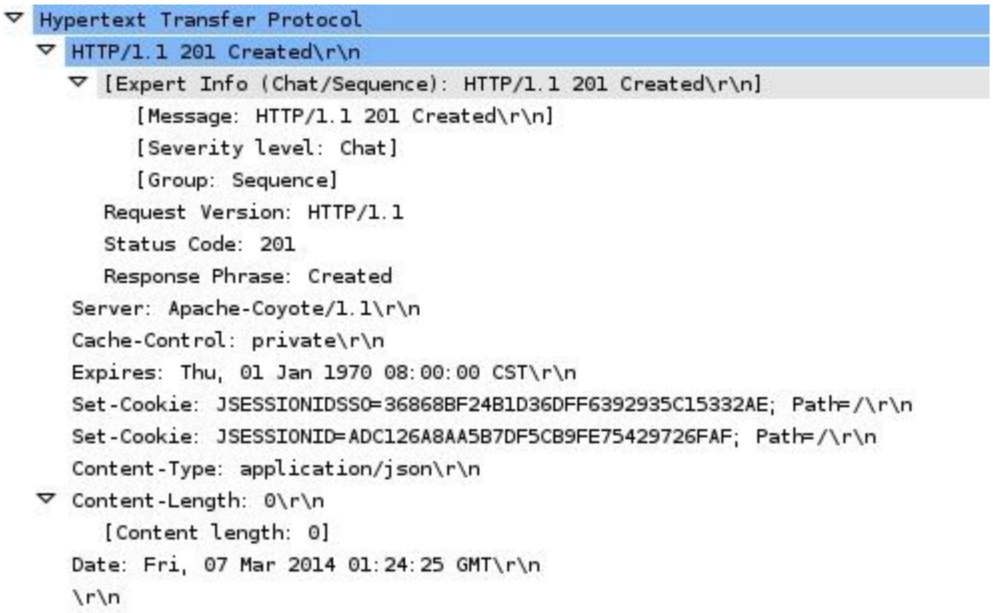
其中箭头表示连接方向，@前面的数字表示交换机的端口，后面表示连接类型，后面的 64 位字符串是交换机的 DPID。

2）配置 UserLink



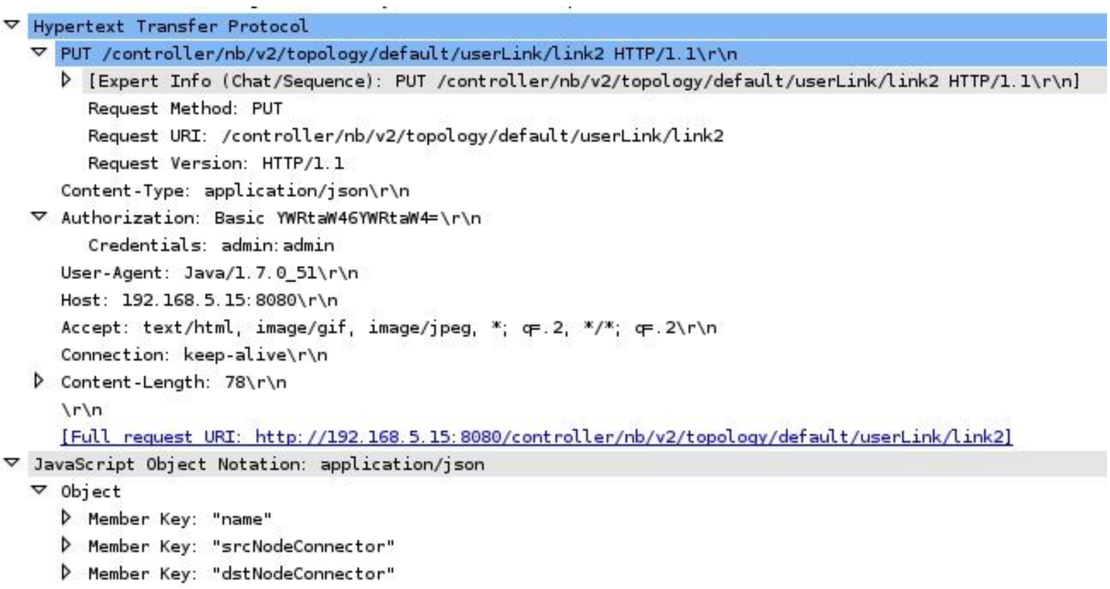
```
▽ Hypertext Transfer Protocol
  ▽ PUT /controller/nb/v2/topology/default/userLink/link2 HTTP/1.1\r\n
    ▷ [Expert Info (Chat/Sequence): PUT /controller/nb/v2/topology/default/userLink/link2 HTTP/1.1\r\n]
      Request Method: PUT
      Request URI: /controller/nb/v2/topology/default/userLink/link2
      Request Version: HTTP/1.1
    Content-Type: application/json\r\n
  ▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
      Credentials: admin:admin
    User-Agent: Java/1.7.0_51\r\n
    Host: 192.168.5.15:8080\r\n
    Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
    Connection: keep-alive\r\n
  ▷ Content-Length: 78\r\n
    \r\n
    [Full request URI: http://192.168.5.15:8080/controller/nb/v2/topology/default/userLink/link2]
▽ JavaScript Object Notation: application/json
  ▽ Object
    ▷ Member Key: "name"
    ▷ Member Key: "srcNodeConnector"
    ▷ Member Key: "dstNodeConnector"
```

客户端配置 UserLinks 信息



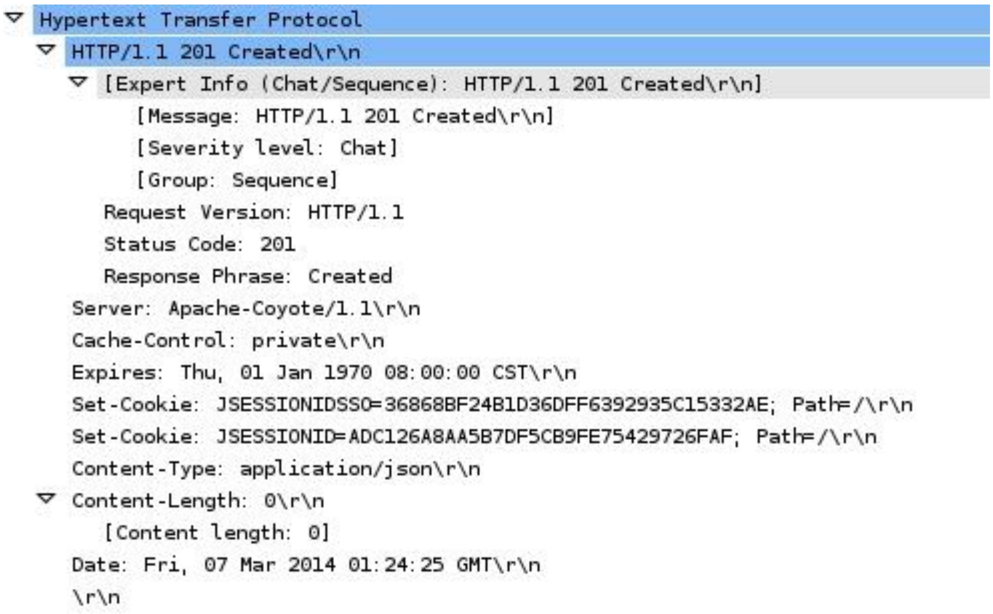
```
▽ Hypertext Transfer Protocol
  ▽ HTTP/1.1 201 Created\r\n
    ▽ [Expert Info (Chat/Sequence): HTTP/1.1 201 Created\r\n]
        [Message: HTTP/1.1 201 Created\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Version: HTTP/1.1
      Status Code: 201
      Response Phrase: Created
    Server: Apache-Coyote/1.1\r\n
    Cache-Control: private\r\n
    Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
    Set-Cookie: JSESSIONIDSSO=36868BF24B1D36DFF6392935C15332AE; Path=/\r\n
    Set-Cookie: JSESSIONID=ADC126A8AA5B7DF5CB9FE75429726FAF; Path=/\r\n
    Content-Type: application/json\r\n
  ▽ Content-Length: 0\r\n
      [Content length: 0]
    Date: Fri, 07 Mar 2014 01:24:25 GMT\r\n
    \r\n
```

服务器应答配置成功信息 201



```
▽ Hypertext Transfer Protocol
  ▽ PUT /controller/nb/v2/topology/default/userLink/link2 HTTP/1.1\r\n
    ▷ [Expert Info (Chat/Sequence): PUT /controller/nb/v2/topology/default/userLink/link2 HTTP/1.1\r\n]
      Request Method: PUT
      Request URI: /controller/nb/v2/topology/default/userLink/link2
      Request Version: HTTP/1.1
    Content-Type: application/json\r\n
  ▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
      Credentials: admin:admin
    User-Agent: Java/1.7.0_51\r\n
    Host: 192.168.5.15:8080\r\n
    Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
    Connection: keep-alive\r\n
  ▷ Content-Length: 78\r\n
    \r\n
    [Full request URI: http://192.168.5.15:8080/controller/nb/v2/topology/default/userLink/link2]
▽ JavaScript Object Notation: application/json
  ▽ Object
    ▷ Member Key: "name"
    ▷ Member Key: "srcNodeConnector"
    ▷ Member Key: "dstNodeConnector"
```

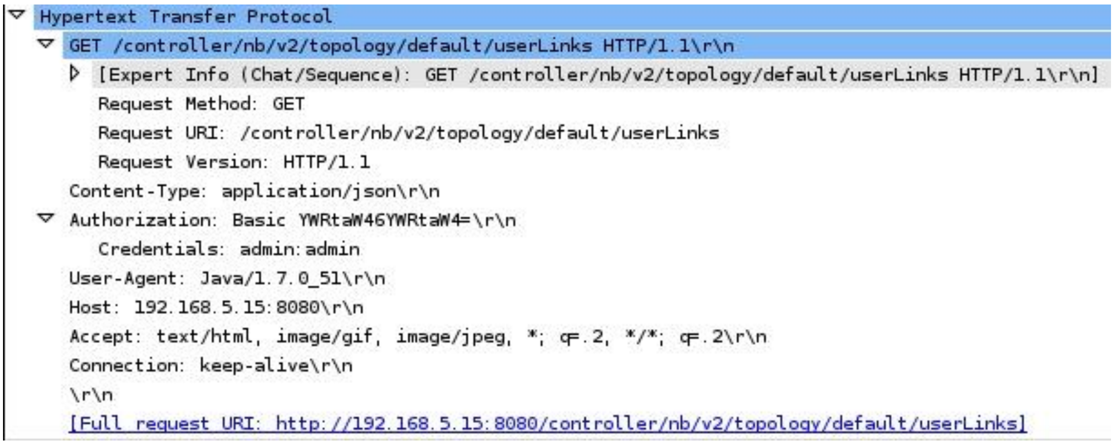
客户端配置 UserLinks 信息



```
▽ Hypertext Transfer Protocol
  ▽ HTTP/1.1 201 Created\r\n
    ▽ [Expert Info (Chat/Sequence): HTTP/1.1 201 Created\r\n]
        [Message: HTTP/1.1 201 Created\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Version: HTTP/1.1
      Status Code: 201
      Response Phrase: Created
    Server: Apache-Coyote/1.1\r\n
    Cache-Control: private\r\n
    Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
    Set-Cookie: JSESSIONIDSSO=36868BF24B1D36DFF6392935C15332AE; Path=/\r\n
    Set-Cookie: JSESSIONID=ADC126A8AA5B7DF5CB9FE75429726FAF; Path=/\r\n
    Content-Type: application/json\r\n
  ▽ Content-Length: 0\r\n
      [Content length: 0]
    Date: Fri, 07 Mar 2014 01:24:25 GMT\r\n
    \r\n
```

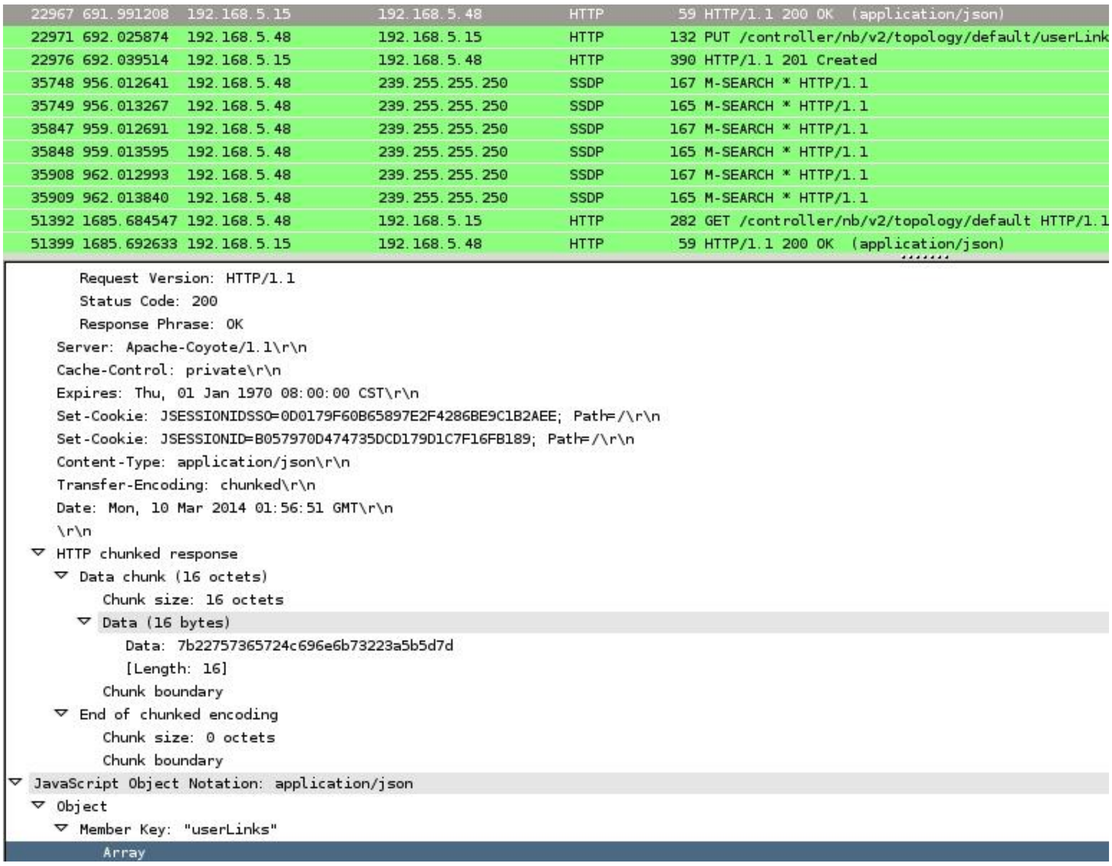
服务器应答配置成功信息 201

3）获取 UserLinks 信息



```
▽ Hypertext Transfer Protocol
  ▽ GET /controller/nb/v2/topology/default/userLinks HTTP/1.1\r\n
    ▷ [Expert Info (Chat/Sequence): GET /controller/nb/v2/topology/default/userLinks HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /controller/nb/v2/topology/default/userLinks
      Request Version: HTTP/1.1
    Content-Type: application/json\r\n
  ▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
      Credentials: admin:admin
    User-Agent: Java/1.7.0_51\r\n
    Host: 192.168.5.15:8080\r\n
    Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://192.168.5.15:8080/controller/nb/v2/topology/default/userLinks]
```

客户端请求 UserLinks 信息

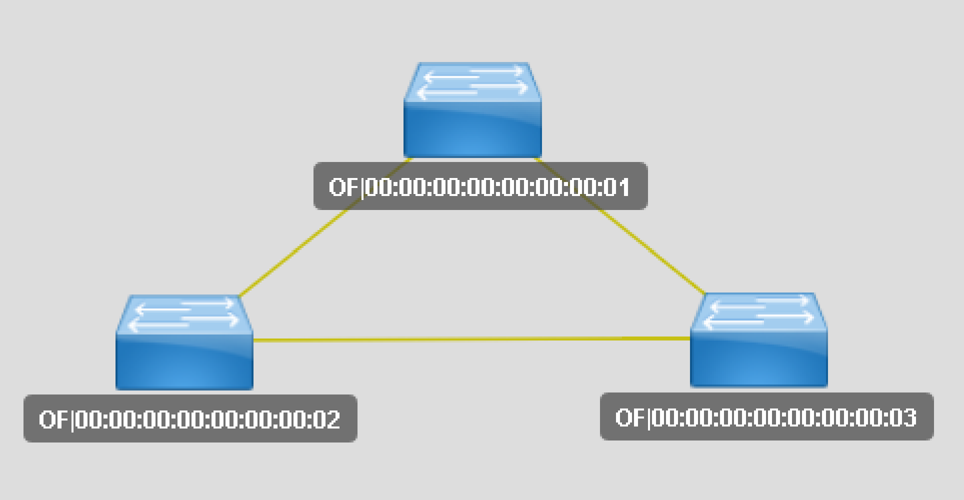


```
22967 691.991208  192.168.5.15     192.168.5.48       HTTP     59 HTTP/1.1 200 OK  (application/json)
22971 692.025874  192.168.5.48     192.168.5.15       HTTP    132 PUT /controller/nb/v2/topology/default/userLink
22976 692.039514  192.168.5.15     192.168.5.48       HTTP    390 HTTP/1.1 201 Created
35748 956.012641  192.168.5.48     239.255.255.250    SSDP    167 M-SEARCH * HTTP/1.1
35749 956.013267  192.168.5.48     239.255.255.250    SSDP    165 M-SEARCH * HTTP/1.1
35847 959.012691  192.168.5.48     239.255.255.250    SSDP    167 M-SEARCH * HTTP/1.1
35848 959.013595  192.168.5.48     239.255.255.250    SSDP    165 M-SEARCH * HTTP/1.1
35908 962.012993  192.168.5.48     239.255.255.250    SSDP    167 M-SEARCH * HTTP/1.1
35909 962.013840  192.168.5.48     239.255.255.250    SSDP    165 M-SEARCH * HTTP/1.1
51392 1685.684547 192.168.5.48     192.168.5.15       HTTP    282 GET /controller/nb/v2/topology/default HTTP/1.1
51399 1685.692633 192.168.5.15     192.168.5.48       HTTP     59 HTTP/1.1 200 OK  (application/json)
```

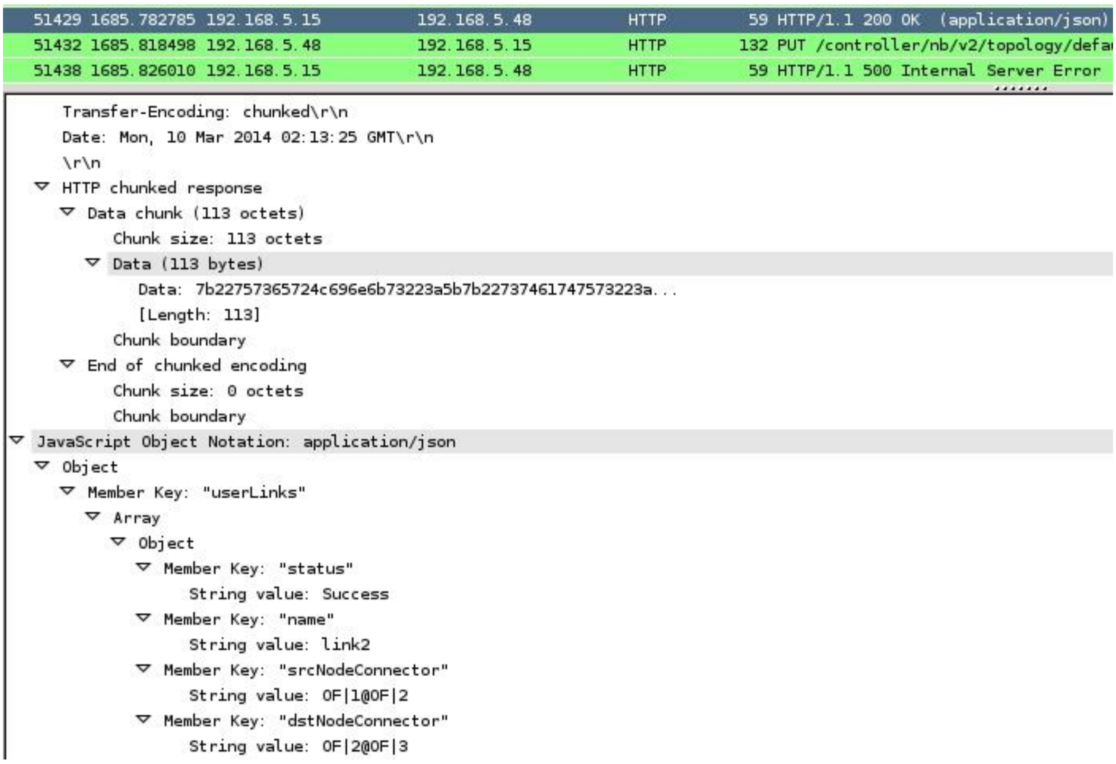
```
      Request Version: HTTP/1.1
      Status Code: 200
      Response Phrase: OK
    Server: Apache-Coyote/1.1\r\n
    Cache-Control: private\r\n
    Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
    Set-Cookie: JSESSIONIDSSO=0D0179F60B65897E2F4286BE9C1B2AEE; Path=/\r\n
    Set-Cookie: JSESSIONID=B057970D474735DCD179D1C7F16FB189; Path=/\r\n
    Content-Type: application/json\r\n
    Transfer-Encoding: chunked\r\n
    Date: Mon, 10 Mar 2014 01:56:51 GMT\r\n
    \r\n
  ▽ HTTP chunked response
    ▽ Data chunk (16 octets)
        Chunk size: 16 octets
      ▽ Data (16 bytes)
          Data: 7b22757365724c696e6b73223a5b5d7d
          [Length: 16]
        Chunk boundary
    ▽ End of chunked encoding
        Chunk size: 0 octets
        Chunk boundary
▽ JavaScript Object Notation: application/json
  ▽ Object
    ▽ Member Key: "userLinks"
        Array
```

服务器应答 UserLinks 信息（无 Userlink）

配置成功的 Userlink 将显示如下拓扑：


OF|00:00:00:00:00:00:00:01
OF|00:00:00:00:00:00:00:02
OF|00:00:00:00:00:00:00:03

该 UserLink 的配置信息为 2 号交换机的 1 端口和 3 号交换机的 2 端口连接



```
51429 1685.782785 192.168.5.15     192.168.5.48     HTTP      59 HTTP/1.1 200 OK  (application/json)
51432 1685.818498 192.168.5.48     192.168.5.15     HTTP     132 PUT /controller/nb/v2/topology/defa
51438 1685.826010 192.168.5.15     192.168.5.48     HTTP      59 HTTP/1.1 500 Internal Server Error

    Transfer-Encoding: chunked\r\n
    Date: Mon, 10 Mar 2014 02:13:25 GMT\r\n
    \r\n
  ▽ HTTP chunked response
    ▽ Data chunk (113 octets)
        Chunk size: 113 octets
      ▽ Data (113 bytes)
          Data: 7b22757365724c696e6b73223a5b7b227374617475573223a...
          [Length: 113]
        Chunk boundary
    ▽ End of chunked encoding
        Chunk size: 0 octets
        Chunk boundary
▽ JavaScript Object Notation: application/json
  ▽ Object
    ▽ Member Key: "userLinks"
      ▽ Array
        ▽ Object
          ▽ Member Key: "status"
              String value: Success
          ▽ Member Key: "name"
              String value: link2
          ▽ Member Key: "srcNodeConnector"
              String value: OF|1@OF|2
          ▽ Member Key: "dstNodeConnector"
              String value: OF|2@OF|3
```

服务器应答 UserLinks 信息（存在 Userlink）

2 Host Tracker REST APIs

本部分 APIs 的应用见 HostTrackerNorthbound

本部分 APIs 提供追踪主机在网络中的位置的功能。主机位置通过 Host node connector 表示。Host node connector 本质是一个逻辑实体，它表示一个交换机或端口。一个主机被 IP 地址和 MAC 地址所标识。

2.1 Host Tracker REST APIs 可提供的服务：

请求分类	客户端请求	服务器响应
地址	GET：返回一个同 IP 地址参数匹配的主机	hostConfig
	PUT：增加一个主机（若该主机 IP 存在，返回失败）	hostConfig
	DELETE：删除一个主机	hostConfig
活动主机	GET：返回本网络所有主机的列表（通过 PUT API 配置，在网络中动态学习）	list
不活动主机	GET：返回本网络中静态配置和 NodeConnector 掉线的主机	list

2.2 Host Tracker REST APIs 数据模型

Data Elements	Data Types	XML Elements	
		名称（类型）	最大/最小 出现
hostConfig	hostConfig	dataLayerAddress (string)	0/1
		nodeType (string)	0/1
		nodeId (string)	0/1
		nodeConnectorType (string)	0/1
		nodeConnectorId (string)	0/1
		vlan (string)	0/1
		staticHost (boolean)	0/1
		networkAddress (string)	0/1
list	hosts	hostConfig (hostConfig)	0/1

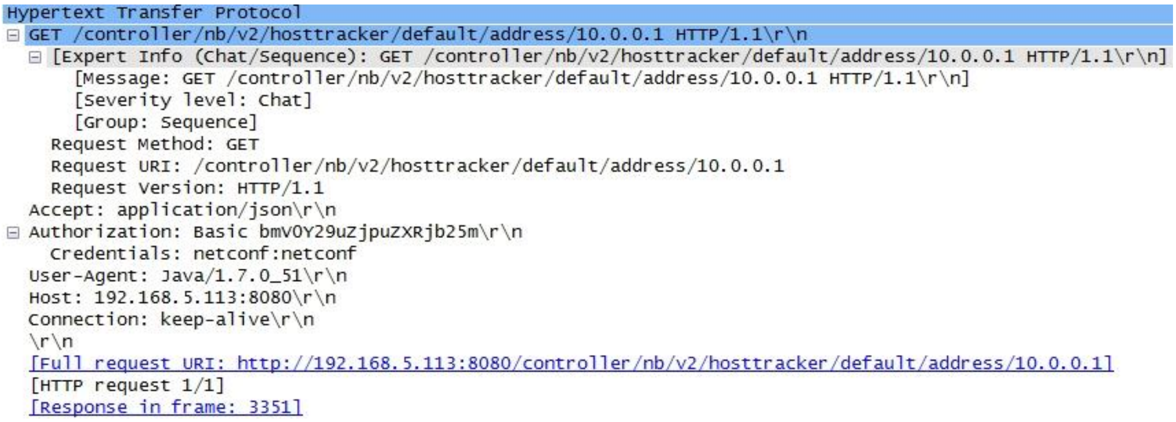
表 4.2-2 Host Tracker REST APIs

2.3 实验分析

2.3.1 抓包分析

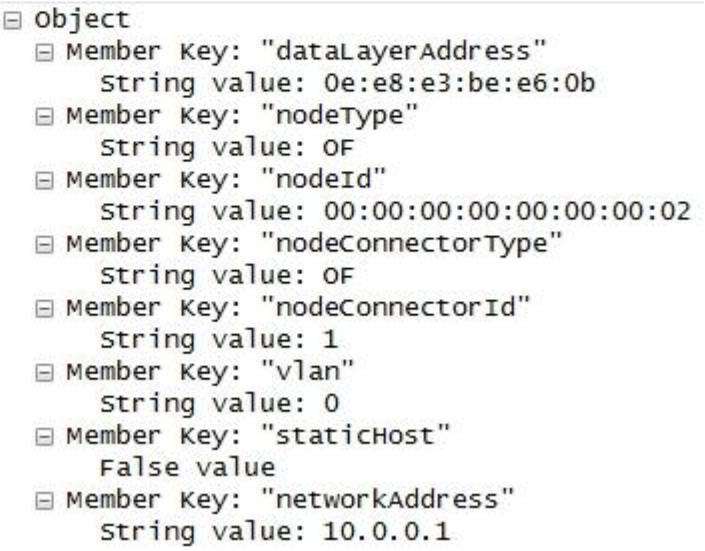
Mininet 中进行操作：h1 ping h2

1）获取指定主机信息



```
Hypertext Transfer Protocol
⊟ GET /controller/nb/v2/hosttracker/default/address/10.0.0.1 HTTP/1.1\r\n
  ⊟ [Expert Info (Chat/Sequence): GET /controller/nb/v2/hosttracker/default/address/10.0.0.1 HTTP/1.1\r\n]
      [Message: GET /controller/nb/v2/hosttracker/default/address/10.0.0.1 HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /controller/nb/v2/hosttracker/default/address/10.0.0.1
    Request Version: HTTP/1.1
  Accept: application/json\r\n
⊟ Authorization: Basic bmV0Y29uZjpuZXRjb25m\r\n
    Credentials: netconf:netconf
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.113:8080\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.113:8080/controller/nb/v2/hosttracker/default/address/10.0.0.1]
  [HTTP request 1/1]
  [Response in frame: 3351]
```

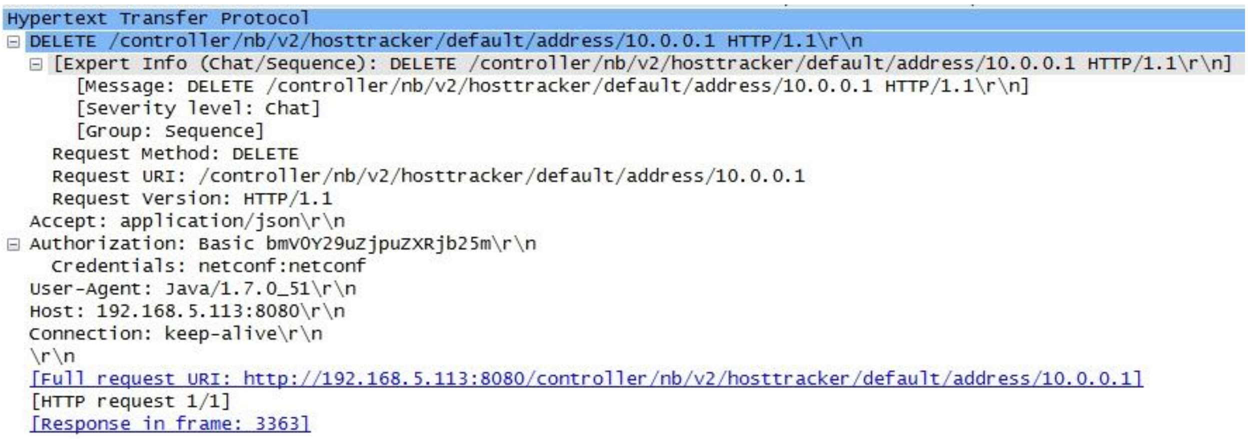
客户端根据主机 ip 发出获取主机信息请求。



```
⊟ Object
  ⊟ Member Key: "dataLayerAddress"
      String value: 0e:e8:e3:be:e6:0b
  ⊟ Member Key: "nodeType"
      String value: OF
  ⊟ Member Key: "nodeId"
      String value: 00:00:00:00:00:00:00:02
  ⊟ Member Key: "nodeConnectorType"
      String value: OF
  ⊟ Member Key: "nodeConnectorId"
      String value: 1
  ⊟ Member Key: "vlan"
      String value: 0
  ⊟ Member Key: "staticHost"
      False value
  ⊟ Member Key: "networkAddress"
      String value: 10.0.0.1
```

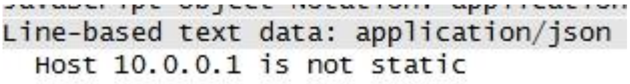
服务端返回主机信息，主要包括上述特征值。

2）删除指定主机



```
Hypertext Transfer Protocol
⊟ DELETE /controller/nb/v2/hosttracker/default/address/10.0.0.1 HTTP/1.1\r\n
  ⊟ [Expert Info (Chat/Sequence): DELETE /controller/nb/v2/hosttracker/default/address/10.0.0.1 HTTP/1.1\r\n]
      [Message: DELETE /controller/nb/v2/hosttracker/default/address/10.0.0.1 HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: DELETE
    Request URI: /controller/nb/v2/hosttracker/default/address/10.0.0.1
    Request Version: HTTP/1.1
  Accept: application/json\r\n
⊟ Authorization: Basic bmV0Y29uZjpuZXRjb25m\r\n
    Credentials: netconf:netconf
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.113:8080\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.113:8080/controller/nb/v2/hosttracker/default/address/10.0.0.1]
  [HTTP request 1/1]
  [Response in frame: 3363]
```

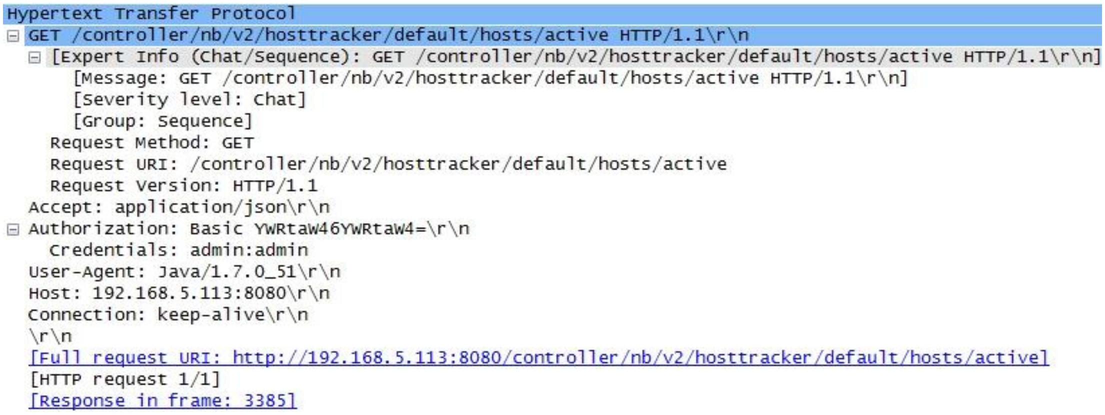
客户端发出删除指定 ip 主机的请求。



```
Line-based text data: application/json
  Host 10.0.0.1 is not static
```

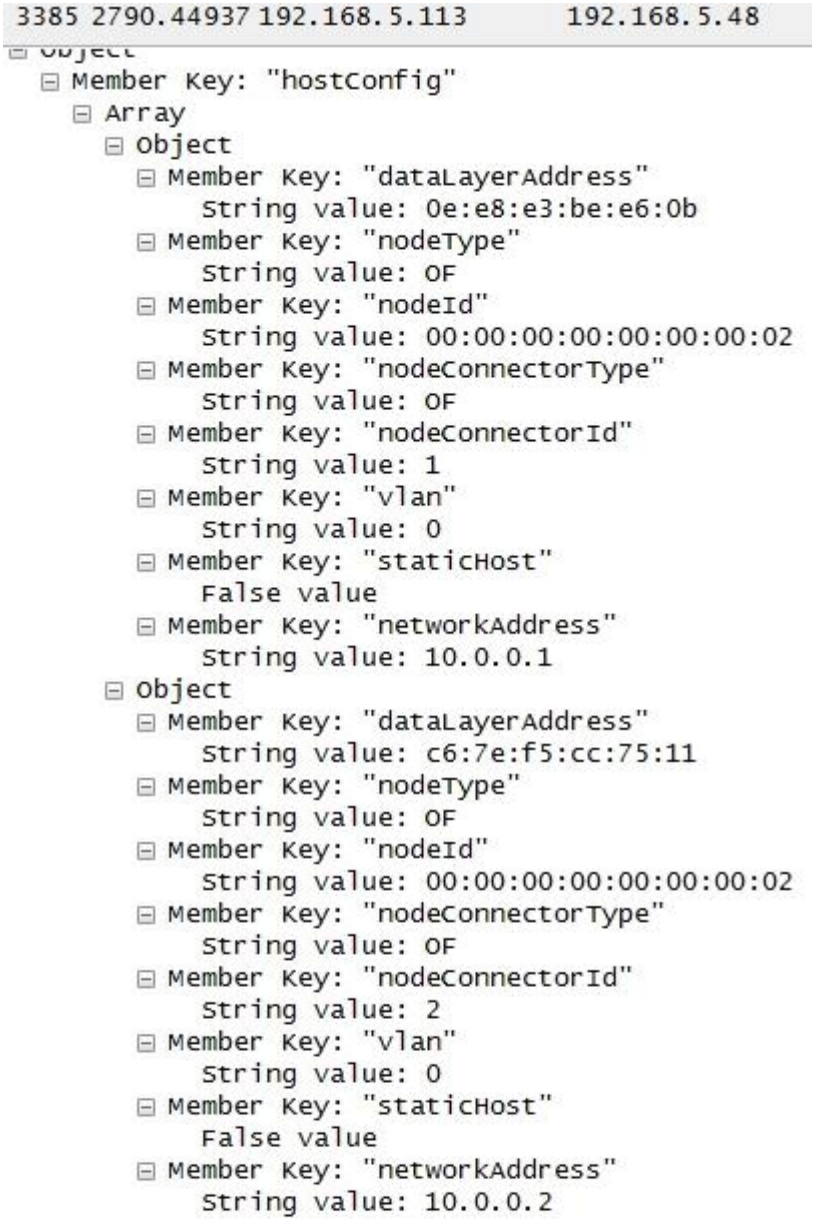
服务端返回 403 信息，并指出主机不是 static 类型。

3）获取活动主机信息



```
Hypertext Transfer Protocol
⊟ GET /controller/nb/v2/hosttracker/default/hosts/active HTTP/1.1\r\n
  ⊟ [Expert Info (Chat/Sequence): GET /controller/nb/v2/hosttracker/default/hosts/active HTTP/1.1\r\n]
      [Message: GET /controller/nb/v2/hosttracker/default/hosts/active HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /controller/nb/v2/hosttracker/default/hosts/active
    Request Version: HTTP/1.1
  Accept: application/json\r\n
⊟ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.113:8080\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.113:8080/controller/nb/v2/hosttracker/default/hosts/active]
  [HTTP request 1/1]
  [Response in frame: 3385]
```

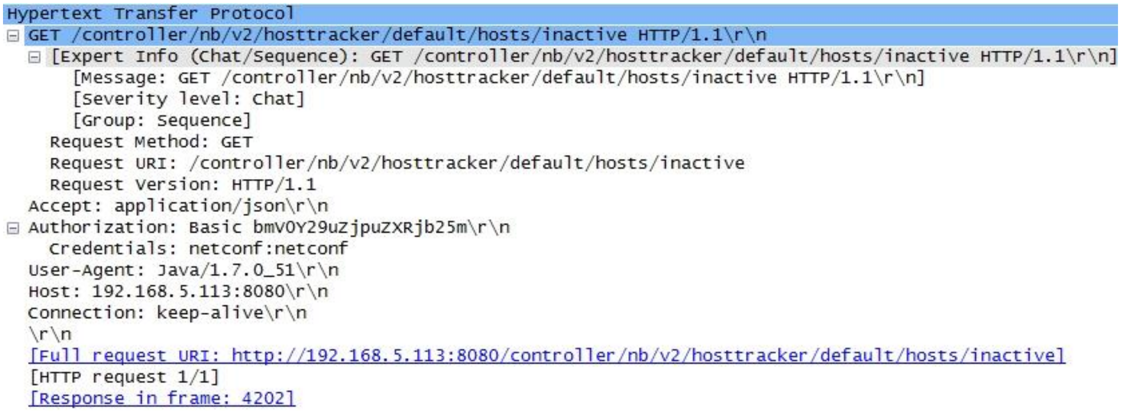
客户端发出获取活动主机请求。



```
3385 2790.44937 192.168.5.113        192.168.5.48
⊟ Object
  ⊟ Member Key: "hostConfig"
    ⊟ Array
      ⊟ Object
        ⊟ Member Key: "dataLayerAddress"
            String value: 0e:e8:e3:be:e6:0b
        ⊟ Member Key: "nodeType"
            String value: OF
        ⊟ Member Key: "nodeId"
            String value: 00:00:00:00:00:00:00:02
        ⊟ Member Key: "nodeConnectorType"
            String value: OF
        ⊟ Member Key: "nodeConnectorId"
            String value: 1
        ⊟ Member Key: "vlan"
            String value: 0
        ⊟ Member Key: "staticHost"
            False value
        ⊟ Member Key: "networkAddress"
            String value: 10.0.0.1
      ⊟ Object
        ⊟ Member Key: "dataLayerAddress"
            String value: c6:7e:f5:cc:75:11
        ⊟ Member Key: "nodeType"
            String value: OF
        ⊟ Member Key: "nodeId"
            String value: 00:00:00:00:00:00:00:02
        ⊟ Member Key: "nodeConnectorType"
            String value: OF
        ⊟ Member Key: "nodeConnectorId"
            String value: 2
        ⊟ Member Key: "vlan"
            String value: 0
        ⊟ Member Key: "staticHost"
            False value
        ⊟ Member Key: "networkAddress"
            String value: 10.0.0.2
```

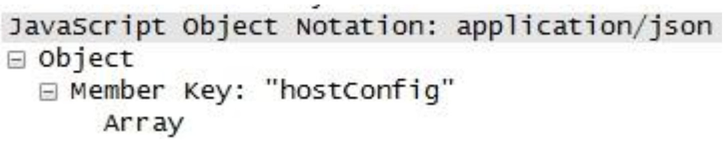
由于操作是：h1 ping h2。所以返回活动的主机是 10.0.0.1 和 10.0.0.2.

4）获取不活动主机信息



```
Hypertext Transfer Protocol
⊟ GET /controller/nb/v2/hosttracker/default/hosts/inactive HTTP/1.1\r\n
  ⊟ [Expert Info (Chat/Sequence): GET /controller/nb/v2/hosttracker/default/hosts/inactive HTTP/1.1\r\n]
      [Message: GET /controller/nb/v2/hosttracker/default/hosts/inactive HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /controller/nb/v2/hosttracker/default/hosts/inactive
    Request Version: HTTP/1.1
  Accept: application/json\r\n
⊟ Authorization: Basic bmV0Y29uZjpuZXRjb25m\r\n
    Credentials: netconf:netconf
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.113:8080\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.113:8080/controller/nb/v2/hosttracker/default/hosts/inactive]
  [HTTP request 1/1]
  [Response in frame: 4202]
```

客户端发出获取不活动主机请求。



```
JavaScript Object Notation: application/json
⊟ Object
  ⊟ Member Key: "hostConfig"
      Array
```

由于主机未活动，故未能看到主机的信息，应当是只有连接到交换机时方能获得信息。

3 Flow Programmer REST APIs

本部分 APIs 的应用见 FlowProgrammerNorthbound
本部分 APIs 提供对流性能的编程。

3.1 Flow Programmer REST APIs 可提供的服务：

请求分类	客户端请求	服务器响应
一般的流	GET：返回指定流的配置列表	list
指定节点上的流	GET：返回在某节点上的流的配置列表	list
指定节点上的静态流	GET：返回与可读的名字、nodeId 匹配的流	flowConfig
	PUT：增加或修改一个流的配置。若流存在则替换最近的流。	flowConfig
	DELETE:删除流配置	(custom)
	POST：切换流配置	(custom)

注：POST 给服务器添加信息（例如，注释）

3.2 Flow Programmer REST APIs 数据模型

Data Elements	Data Types	XML Elements	
		名称（类型）	最大/最小 出现
flowConfig		vlanPriority (string)	0/1
		nwDst (string)	0/1
		idleTimeout (string)	0/1
		tosBits (string)	0/1
		name (string)	0/1
		dlSrc (string)	0/1
		hardTimeout (string)	0/1
		protocol (string)	0/1
		node (node)	0/1
		cookie (string)	0/1
		installInHw (string)	0/1
		tpDst (string)	0/1
		nwSrc (string)	0/1
		vlanId (string)	0/1
		tpSrc (string)	0/1
		ingressPort (string)	0/1
		etherType (string)	0/1
		dlDst (string)	0/1
		priority (string)	0/1
		actions (string)	0/unbounded
list	flowConfigs	flowConfig (flowConfig)	0/unbounded
Node	node	type (string)	0/1
		id (string)	0/1

表 4.2-3 Flow Programmer REST APIs

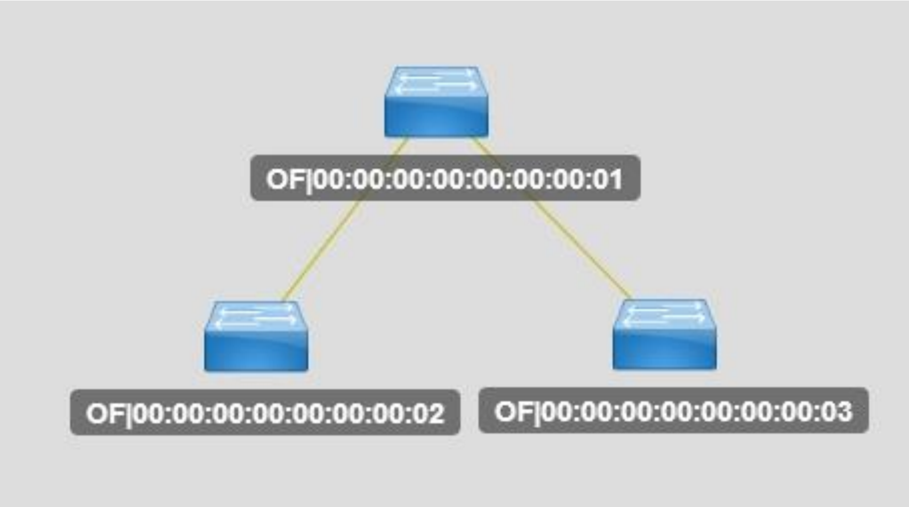
3.3 实验分析

3.3.1 实验拓扑

mininet 输入如下命令：

mn --topo tree,depth=2,fanout=2 --controller=remote,p=192.168.5.113,port=6633

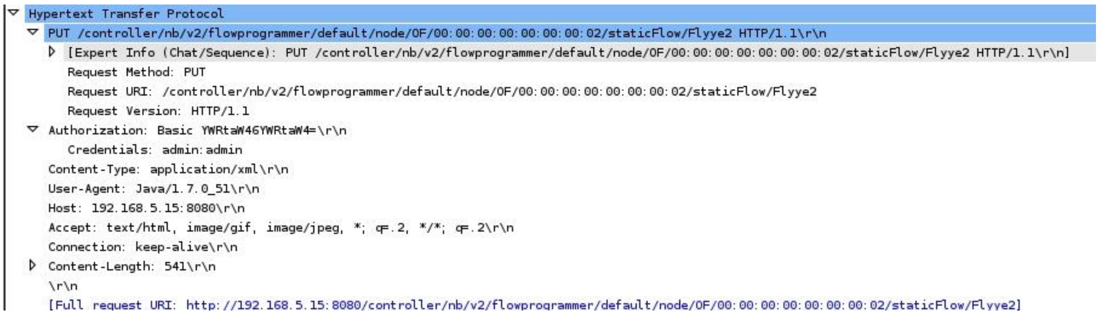
将产生如下拓扑：


OF|00:00:00:00:00:00:00:01
OF|00:00:00:00:00:00:00:02
OF|00:00:00:00:00:00:00:03

B.在交换机 S2 中写入流表项，发往 h3 主机的数据报都进行丢弃。
C.在 mininet 中输入命令：h1 ping h2

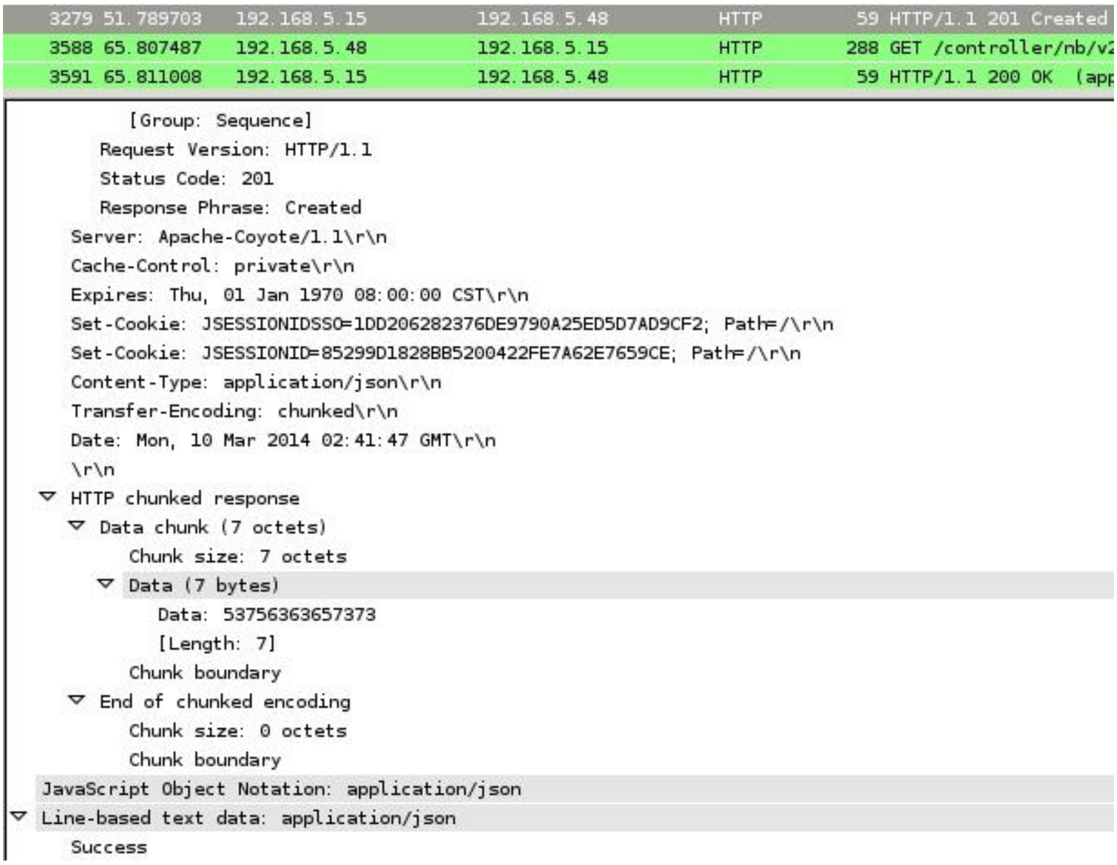
3.3.2 实验抓包分析

1）下发流表即 FlowMod



```
▽ Hypertext Transfer Protocol
  ▽ PUT /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:02/staticFlow/Flyye2 HTTP/1.1\r\n
    ▷ [Expert Info (Chat/Sequence): PUT /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:02/staticFlow/Flyye2 HTTP/1.1\r\n]
      Request Method: PUT
      Request URI: /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:02/staticFlow/Flyye2
      Request Version: HTTP/1.1
  ▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
      Credentials: admin:admin
    Content-Type: application/xml\r\n
    User-Agent: Java/1.7.0_51\r\n
    Host: 192.168.5.15:8080\r\n
    Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
    Connection: keep-alive\r\n
  ▷ Content-Length: 541\r\n
    \r\n
    [Full request URI: http://192.168.5.15:8080/controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:02/staticFlow/Flyye2]
```

客户端响应



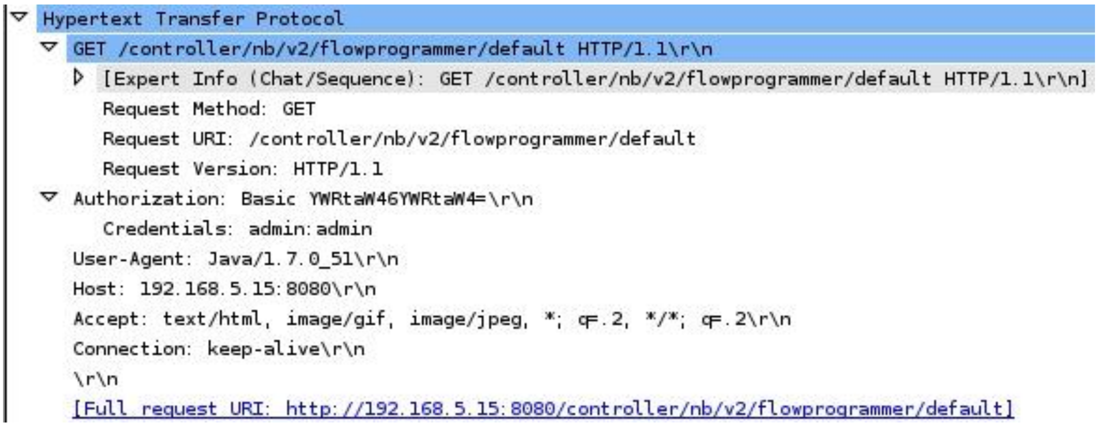
```
3279 51.789703   192.168.5.15      192.168.5.48      HTTP      59 HTTP/1.1 201 Created
3588 65.807487   192.168.5.48      192.168.5.15      HTTP     288 GET /controller/nb/v2
3591 65.811008   192.168.5.15      192.168.5.48      HTTP      59 HTTP/1.1 200 OK  (app
```

```
          [Group: Sequence]
        Request Version: HTTP/1.1
        Status Code: 201
        Response Phrase: Created
      Server: Apache-Coyote/1.1\r\n
      Cache-Control: private\r\n
      Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
      Set-Cookie: JSESSIONIDSSO=1DD206282376DE9790A25ED5D7AD9CF2; Path=/\r\n
      Set-Cookie: JSESSIONID=85299D1828BB5200422FE7A62E7659CE; Path=/\r\n
      Content-Type: application/json\r\n
      Transfer-Encoding: chunked\r\n
      Date: Mon, 10 Mar 2014 02:41:47 GMT\r\n
      \r\n
    ▽ HTTP chunked response
      ▽ Data chunk (7 octets)
          Chunk size: 7 octets
        ▽ Data (7 bytes)
            Data: 53756363657373
            [Length: 7]
          Chunk boundary
      ▽ End of chunked encoding
          Chunk size: 0 octets
          Chunk boundary
    JavaScript Object Notation: application/json
  ▽ Line-based text data: application/json
      Success
```

服务器应答

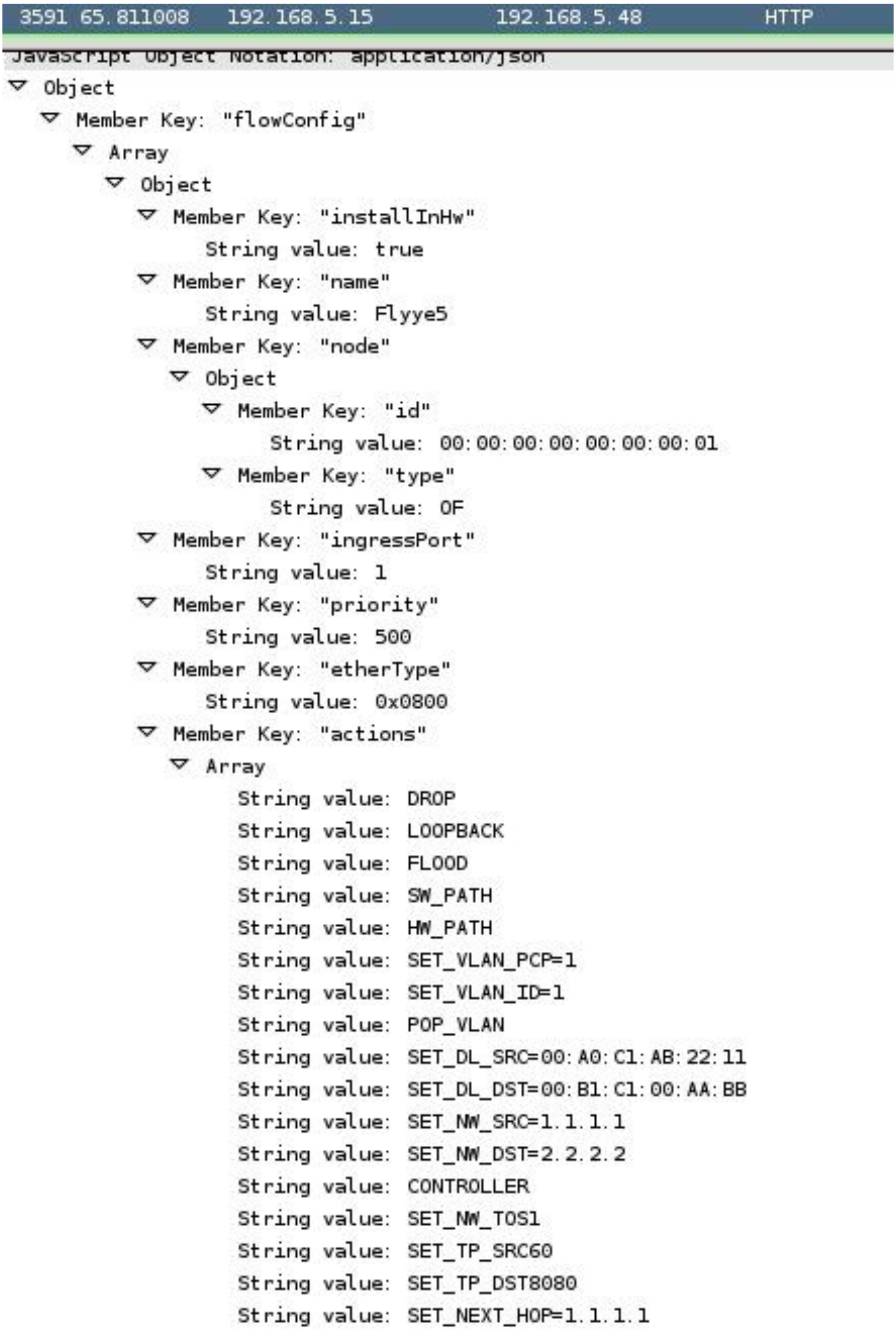
图中的信息 Success 表示创建成功，返回 201。

2）获取流表信息



```
▽ Hypertext Transfer Protocol
  ▽ GET /controller/nb/v2/flowprogrammer/default HTTP/1.1\r\n
    ▷ [Expert Info (Chat/Sequence): GET /controller/nb/v2/flowprogrammer/default HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /controller/nb/v2/flowprogrammer/default
      Request Version: HTTP/1.1
  ▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
      Credentials: admin:admin
    User-Agent: Java/1.7.0_51\r\n
    Host: 192.168.5.15:8080\r\n
    Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://192.168.5.15:8080/controller/nb/v2/flowprogrammer/default]
```

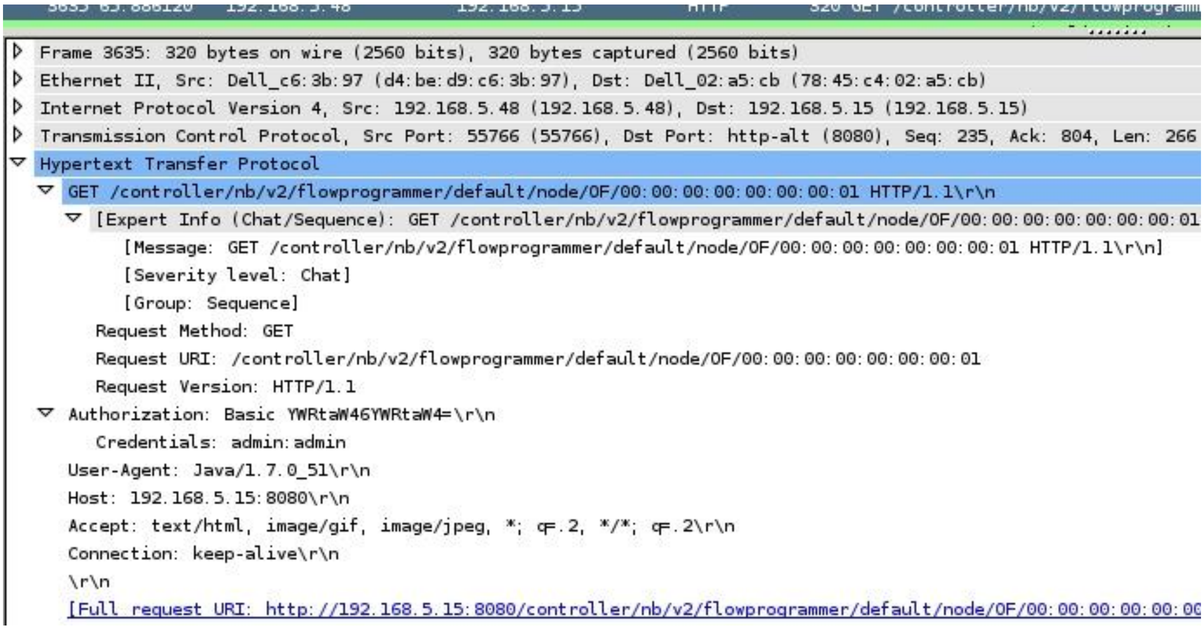
客户端请求



```
3591 65.811008    192.168.5.15          192.168.5.48          HTTP
JavaScript Object Notation: application/json
▽ Object
  ▽ Member Key: "flowConfig"
    ▽ Array
      ▽ Object
        ▽ Member Key: "installInHw"
            String value: true
        ▽ Member Key: "name"
            String value: Flyye5
        ▽ Member Key: "node"
          ▽ Object
            ▽ Member Key: "id"
                String value: 00:00:00:00:00:00:00:01
            ▽ Member Key: "type"
                String value: OF
        ▽ Member Key: "ingressPort"
            String value: 1
        ▽ Member Key: "priority"
            String value: 500
        ▽ Member Key: "etherType"
            String value: 0x0800
        ▽ Member Key: "actions"
          ▽ Array
              String value: DROP
              String value: LOOPBACK
              String value: FLOOD
              String value: SW_PATH
              String value: HW_PATH
              String value: SET_VLAN_PCP=1
              String value: SET_VLAN_ID=1
              String value: POP_VLAN
              String value: SET_DL_SRC=00:A0:C1:AB:22:11
              String value: SET_DL_DST=00:B1:C1:00:AA:BB
              String value: SET_NW_SRC=1.1.1.1
              String value: SET_NW_DST=2.2.2.2
              String value: CONTROLLER
              String value: SET_NW_TOS1
              String value: SET_TP_SRC60
              String value: SET_TP_DST8080
              String value: SET_NEXT_HOP=1.1.1.1
```

服务端响应流表信息

3）获取指定交换机流表

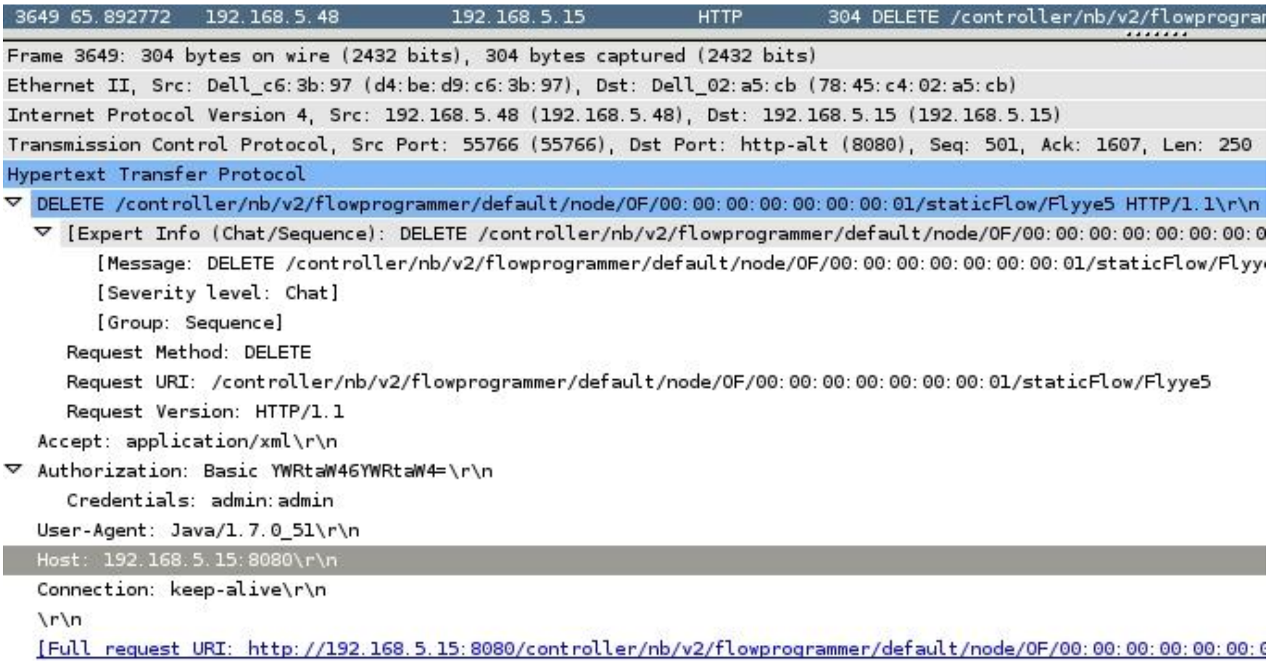


```
3635 65.888120   192.168.5.48          192.168.5.15          HTTP        320 GET /controller/nb/v2/flowprogramm
▷ Frame 3635: 320 bytes on wire (2560 bits), 320 bytes captured (2560 bits)
▷ Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
▷ Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
▷ Transmission Control Protocol, Src Port: 55766 (55766), Dst Port: http-alt (8080), Seq: 235, Ack: 804, Len: 266
▽ Hypertext Transfer Protocol
  ▽ GET /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:01 HTTP/1.1\r\n
    ▽ [Expert Info (Chat/Sequence): GET /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:01
        [Message: GET /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:01 HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:01
      Request Version: HTTP/1.1
  ▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
      Credentials: admin:admin
    User-Agent: Java/1.7.0_51\r\n
    Host: 192.168.5.15:8080\r\n
    Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://192.168.5.15:8080/controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00
```

客户端请求

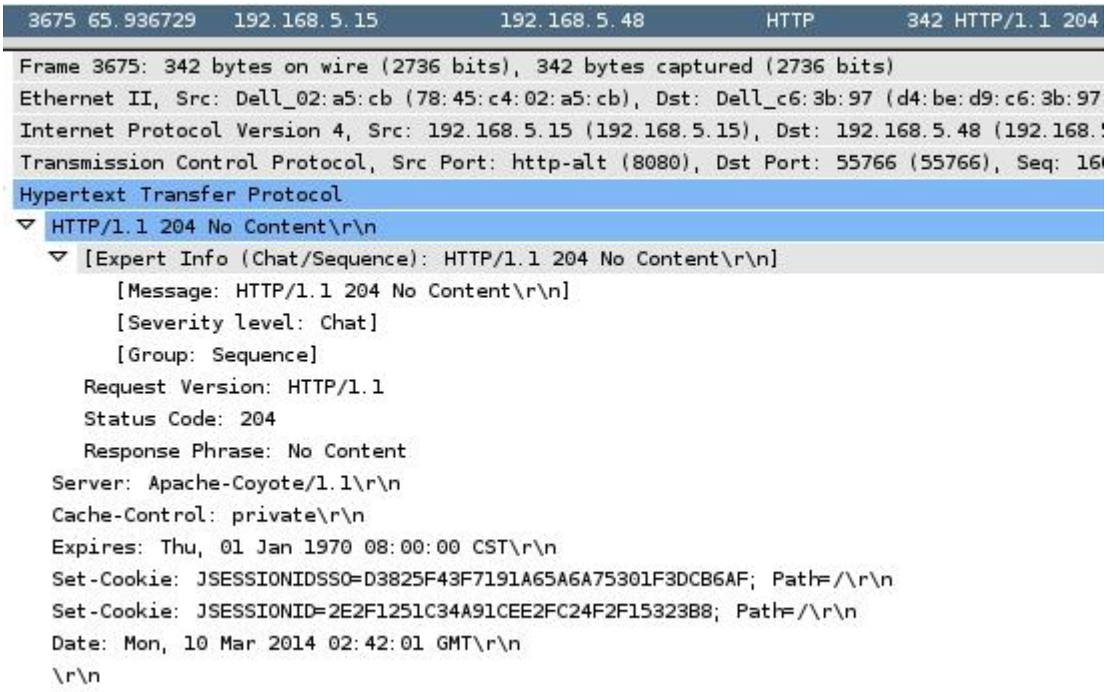
由于该网络中的交换机只有一个流表，所以服务端的返回信息同（2）。

4）删除流表



```
3649 65.892772   192.168.5.48          192.168.5.15          HTTP        304 DELETE /controller/nb/v2/flowprogra
Frame 3649: 304 bytes on wire (2432 bits), 304 bytes captured (2432 bits)
Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
Transmission Control Protocol, Src Port: 55766 (55766), Dst Port: http-alt (8080), Seq: 501, Ack: 1607, Len: 250
Hypertext Transfer Protocol
▽ DELETE /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:01/staticFlow/Flyye5 HTTP/1.1\r\n
  ▽ [Expert Info (Chat/Sequence): DELETE /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:0
      [Message: DELETE /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:01/staticFlow/Flyy
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: DELETE
    Request URI: /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:00:01/staticFlow/Flyye5
    Request Version: HTTP/1.1
  Accept: application/xml\r\n
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:0
```

客户端删除流表请求



```
 3675 65.936729   192.168.5.15          192.168.5.48          HTTP        342 HTTP/1.1 204
Frame 3675: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
Ethernet II, Src: Dell_02:a5:cb (78:45:c4:02:a5:cb), Dst: Dell_c6:3b:97 (d4:be:d9:c6:3b:97
Internet Protocol Version 4, Src: 192.168.5.15 (192.168.5.15), Dst: 192.168.5.48 (192.168.
Transmission Control Protocol, Src Port: http-alt (8080), Dst Port: 55766 (55766), Seq: 16
Hypertext Transfer Protocol
▽ HTTP/1.1 204 No Content\r\n
  ▽ [Expert Info (Chat/Sequence): HTTP/1.1 204 No Content\r\n]
       [Message: HTTP/1.1 204 No Content\r\n]
       [Severity level: Chat]
       [Group: Sequence]
     Request Version: HTTP/1.1
     Status Code: 204
     Response Phrase: No Content
  Server: Apache-Coyote/1.1\r\n
  Cache-Control: private\r\n
  Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
  Set-Cookie: JSESSIONIDSSO=D3825F43F7191A65A6A75301F3DCB6AF; Path=/\r\n
  Set-Cookie: JSESSIONID=2E2F1251C34A91CEE2FC24F2F15323B8; Path=/\r\n
  Date: Mon, 10 Mar 2014 02:42:01 GMT\r\n
  \r\n
```

服务器响应删除流表请求

删除成功，返回 204。

4 Static Routing REST APIs

本部分 APIs 的应用见 StaticRoutingNorthbound
本部分 APIs 涉及到静态路由的管理。

4.1 Static Routing REST APIs 可提供服务：

请求分类	客户端请求	服务器响应
routes	GET:返回静态路由现状的列表	list
Route	GET:返回所提供配置名称的静态路由	staticRoute
	PUT:增加新的静态路由（如果该路由存在，返回错误）	staticRoute
	DELETE:删除静态路由	(custom)

4.2 Static Routing REST APIs 数据模型

Data Elements	Data Types	XML Elements	
		名称（类型）	最大/最小 出现
list	staticRoutes	staticRoute (staticRoute)	0/unbounded

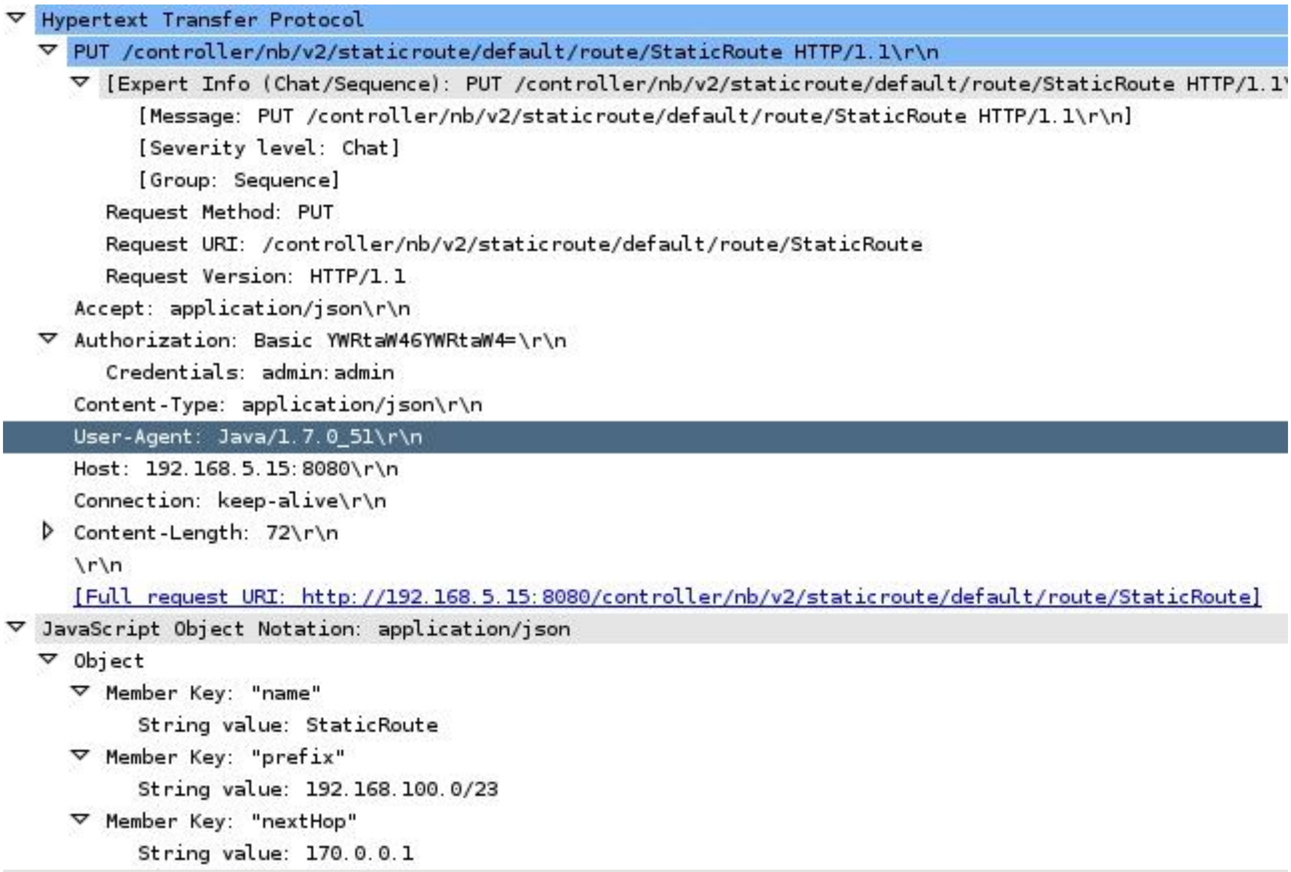
staticRoute	staticRoute	name (string) 注：The name of the static route.	0/1
		prefix (string) 注：The prefix for the route. Format: A.B.C.D/MM Where A.B.C.D is the Default Gateway IP (L3) or ARP Querier IP (L2)	0/1
		nextHop (string) 注：NextHop IP-Address (or) datapath ID/port list: xx:xx:xx:xx:xx:xx:xx:xx/a,b,c-m,r-t,y	0/1

表 4.2-4 Static Routing REST APIs

4.3 实验分析

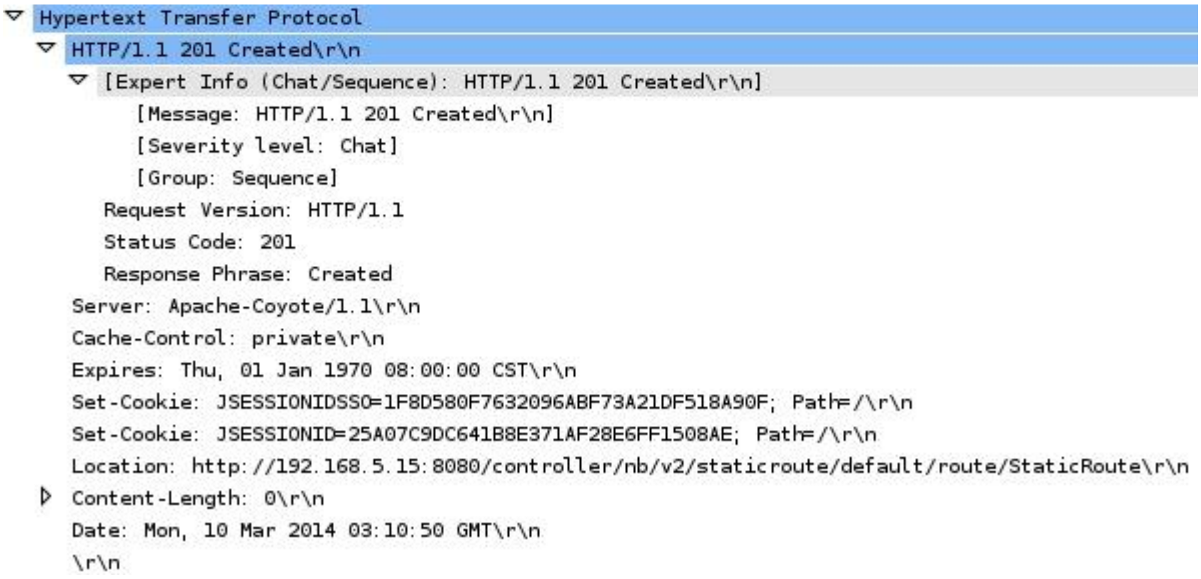
4.3.1 抓包情况分析

1）设置静态路由



```
▽ Hypertext Transfer Protocol
  ▽ PUT /controller/nb/v2/staticroute/default/route/StaticRoute HTTP/1.1\r\n
    ▽ [Expert Info (Chat/Sequence): PUT /controller/nb/v2/staticroute/default/route/StaticRoute HTTP/1.1\
        [Message: PUT /controller/nb/v2/staticroute/default/route/StaticRoute HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: PUT
      Request URI: /controller/nb/v2/staticroute/default/route/StaticRoute
      Request Version: HTTP/1.1
    Accept: application/json\r\n
  ▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
      Credentials: admin:admin
    Content-Type: application/json\r\n
    User-Agent: Java/1.7.0_51\r\n
    Host: 192.168.5.15:8080\r\n
    Connection: keep-alive\r\n
  ▷ Content-Length: 72\r\n
    \r\n
    [Full request URI: http://192.168.5.15:8080/controller/nb/v2/staticroute/default/route/StaticRoute]
▽ JavaScript Object Notation: application/json
  ▽ Object
    ▽ Member Key: "name"
        String value: StaticRoute
    ▽ Member Key: "prefix"
        String value: 192.168.100.0/23
    ▽ Member Key: "nextHop"
        String value: 170.0.0.1
```

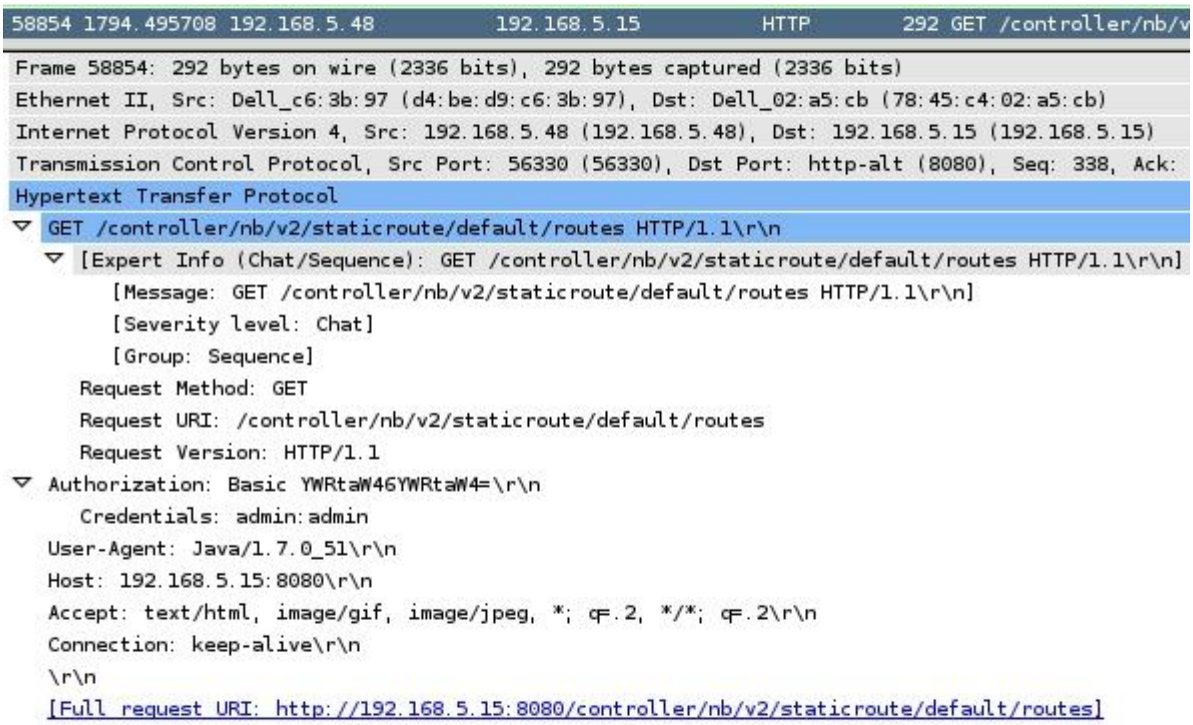
客户端向服务端设置静态路由请求。



```
Hypertext Transfer Protocol
  HTTP/1.1 201 Created\r\n
    [Expert Info (Chat/Sequence): HTTP/1.1 201 Created\r\n]
      [Message: HTTP/1.1 201 Created\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Version: HTTP/1.1
    Status Code: 201
    Response Phrase: Created
  Server: Apache-Coyote/1.1\r\n
  Cache-Control: private\r\n
  Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
  Set-Cookie: JSESSIONIDSSO=1F8D580F7632096ABF73A21DF518A90F; Path=/\r\n
  Set-Cookie: JSESSIONID=25A07C9DC641B8E371AF28E6FF1508AE; Path=/\r\n
  Location: http://192.168.5.15:8080/controller/nb/v2/staticroute/default/route/StaticRoute\r\n
  Content-Length: 0\r\n
  Date: Mon, 10 Mar 2014 03:10:50 GMT\r\n
  \r\n
```

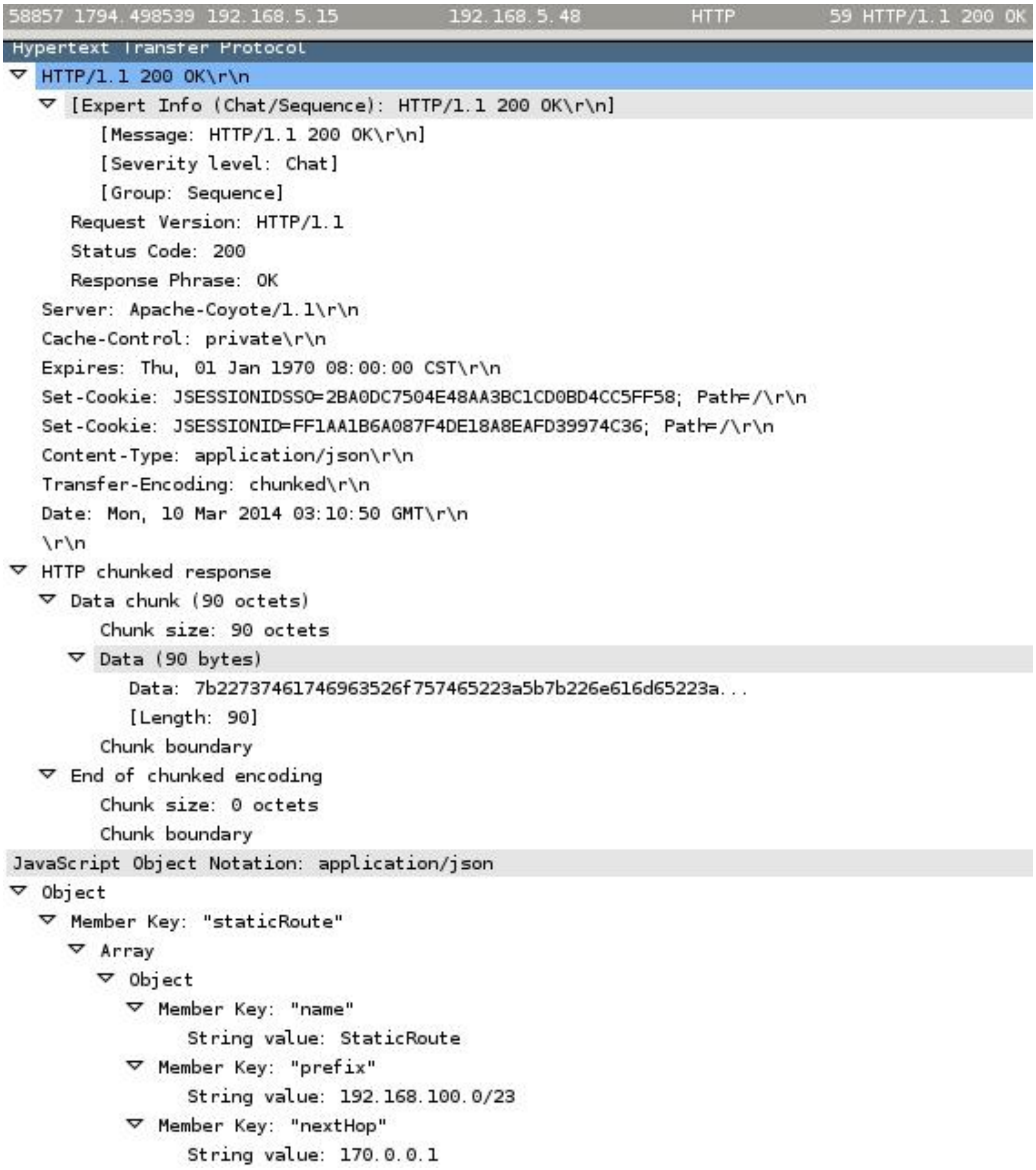
服务端响应，返回 201，创建成功

2）获取静态路由信息



```
58854 1794.495708 192.168.5.48          192.168.5.15          HTTP        292 GET /controller/nb/v
Frame 58854: 292 bytes on wire (2336 bits), 292 bytes captured (2336 bits)
Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
Transmission Control Protocol, Src Port: 56330 (56330), Dst Port: http-alt (8080), Seq: 338, Ack:
Hypertext Transfer Protocol
  GET /controller/nb/v2/staticroute/default/routes HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /controller/nb/v2/staticroute/default/routes HTTP/1.1\r\n]
      [Message: GET /controller/nb/v2/staticroute/default/routes HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /controller/nb/v2/staticroute/default/routes
    Request Version: HTTP/1.1
  Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/staticroute/default/routes]
```

客户端请求静态路由信息



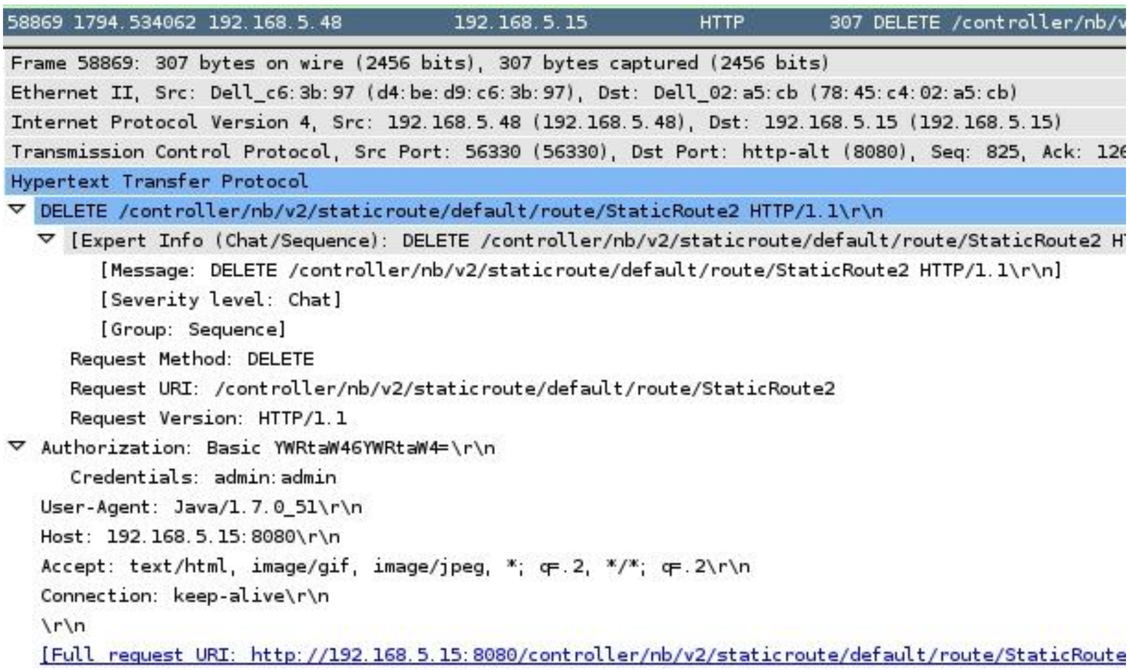
```
58857 1794.498539 192.168.5.15          192.168.5.48          HTTP          59 HTTP/1.1 200 OK
Hypertext Transfer Protocol
▽ HTTP/1.1 200 OK\r\n
  ▽ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [Message: HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Version: HTTP/1.1
    Status Code: 200
    Response Phrase: OK
  Server: Apache-Coyote/1.1\r\n
  Cache-Control: private\r\n
  Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
  Set-Cookie: JSESSIONIDSSO=2BA0DC7504E48AA3BC1CD0BD4CC5FF58; Path=/\r\n
  Set-Cookie: JSESSIONID=FF1AA1B6A087F4DE18A8EAFD39974C36; Path=/\r\n
  Content-Type: application/json\r\n
  Transfer-Encoding: chunked\r\n
  Date: Mon, 10 Mar 2014 03:10:50 GMT\r\n
  \r\n
▽ HTTP chunked response
  ▽ Data chunk (90 octets)
      Chunk size: 90 octets
    ▽ Data (90 bytes)
        Data: 7b227374617469635265f757465223a5b7b226e616d65223a...
        [Length: 90]
      Chunk boundary
  ▽ End of chunked encoding
      Chunk size: 0 octets
      Chunk boundary
JavaScript Object Notation: application/json
▽ Object
  ▽ Member Key: "staticRoute"
    ▽ Array
      ▽ Object
        ▽ Member Key: "name"
            String value: StaticRoute
        ▽ Member Key: "prefix"
            String value: 192.168.100.0/23
        ▽ Member Key: "nextHop"
            String value: 170.0.0.1
```

请求成功，响应 200，并返回静态路由信息。

```
----------StaticRoute-----------
StaticRoute:name = StaticRoute
StaticRoute:nexthop = 170.0.0.1
StaticRoute:prefix = 192.168.100.0/23
```

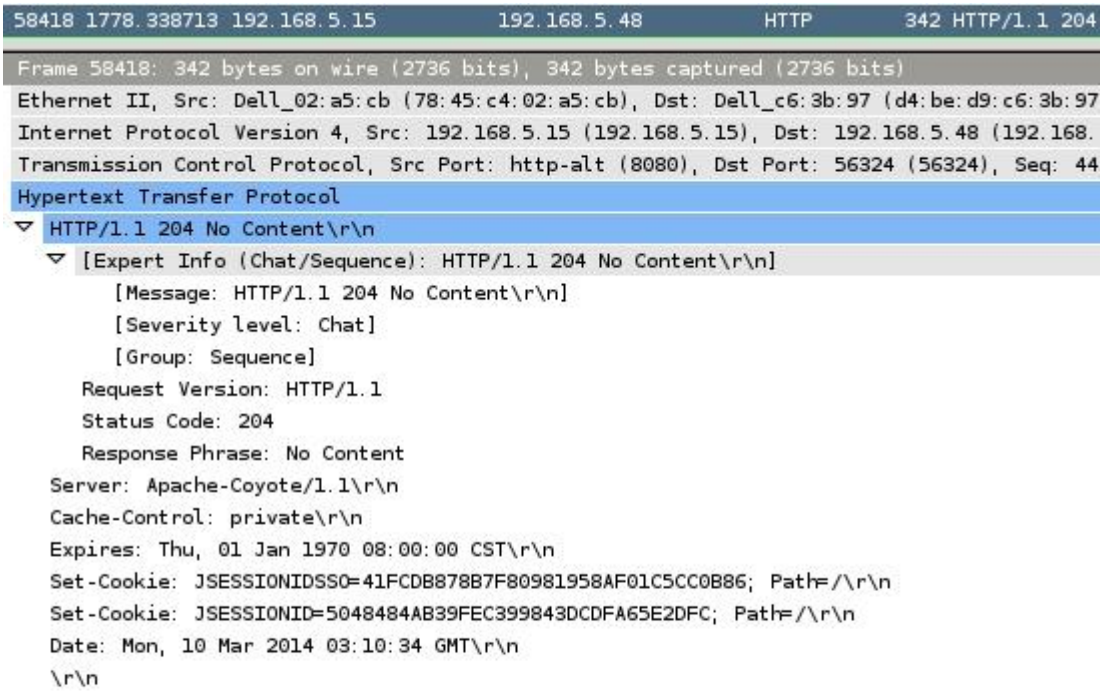
包括路由名称、网络号、下一跳地址

3）删除静态路由



```
58869 1794.534062 192.168.5.48          192.168.5.15          HTTP        307 DELETE /controller/nb/v
Frame 58869: 307 bytes on wire (2456 bits), 307 bytes captured (2456 bits)
Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
Transmission Control Protocol, Src Port: 56330 (56330), Dst Port: http-alt (8080), Seq: 825, Ack: 12
Hypertext Transfer Protocol
▽ DELETE /controller/nb/v2/staticroute/default/route/StaticRoute2 HTTP/1.1\r\n
  ▽ [Expert Info (Chat/Sequence): DELETE /controller/nb/v2/staticroute/default/route/StaticRoute2 H
      [Message: DELETE /controller/nb/v2/staticroute/default/route/StaticRoute2 HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: DELETE
    Request URI: /controller/nb/v2/staticroute/default/route/StaticRoute2
    Request Version: HTTP/1.1
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/staticroute/default/route/StaticRoute
```

客户端删除流表请求



```
58418 1778.338713 192.168.5.15          192.168.5.48          HTTP        342 HTTP/1.1 204
Frame 58418: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
Ethernet II, Src: Dell_02:a5:cb (78:45:c4:02:a5:cb), Dst: Dell_c6:3b:97 (d4:be:d9:c6:3b:97
Internet Protocol Version 4, Src: 192.168.5.15 (192.168.5.15), Dst: 192.168.5.48 (192.168.
Transmission Control Protocol, Src Port: http-alt (8080), Dst Port: 56324 (56324), Seq: 44
Hypertext Transfer Protocol
▽ HTTP/1.1 204 No Content\r\n
  ▽ [Expert Info (Chat/Sequence): HTTP/1.1 204 No Content\r\n]
      [Message: HTTP/1.1 204 No Content\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Version: HTTP/1.1
    Status Code: 204
    Response Phrase: No Content
  Server: Apache-Coyote/1.1\r\n
  Cache-Control: private\r\n
  Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
  Set-Cookie: JSESSIONIDSSO=41FCDB878B7F80981958AF01C5CC0B86; Path=/\r\n
  Set-Cookie: JSESSIONID=5048484AB39FEC399843DCDFA65E2DFC; Path=/\r\n
  Date: Mon, 10 Mar 2014 03:10:34 GMT\r\n
  \r\n
```

删除成功，返回204.

5 Statistics REST APIs

本部分 APIs 的应用见 StatisticsNorthbound
本部分 APIs 返回被南向的协议插件（如 OpenFlow）公开的多种统计信息。

5.1 Statistics REST APIs 可提供的服务：

请求分类	客户端请求	服务器响应
flow	GET:返回所有节点的所有流的统计信息的列表	list
port	GET:返回所有经过节点上的 NodeConnector 的端口统计信息	list
table	GET：返回在所有节点上的链表信息	list
Flow on a node	GET:返回指定节点上的流的统计信息的列表	nodeFlowStatistics
Port on a node	GET:返回所有经过指定节点上的 NodeConnector 的端口统计信息	nodePortStatistics
table on a node	GET：返回在指定节点上的链表信息	nodeTableStatistics

5.2 Statistics REST APIs 数据模型

Data Elements	Data Types	XML Elements	
		名称（类型）	最大/最小 出现
action	action	type (actionType)	0/1
flow	flow	actions (action)	0/unbounded
		hardTimeout (short)	1/1
		idleTimeout (short)	1/1
		match (match)	0/1
		priority (short)	1/1
		id (long)	1/1
FlowStat	flowOnNode	flow (flow)	0/1
		tableId (byte)	1/1
		byteCount (long)	1/1
		durationSeconds (int)	1/1
		packetCount (long)	1/1
		durationNanoseconds (int)	1/1
list	allTableStatistics	tableStatistics (tableStatistics)	0/unbounded
match	match	matchField (matchField)	0/unbounded
matchField	matchField		
node	node	type (string)	0/1
		id (string)	0/1
nodeConnector	nodeConnector	type (string)	0/1

		id (string)	0/1
		node (node)	0/1
nodeConnectorStatistics	nodeConnectorStatistics	receivePackets (long)	1/1
		receiveBytes (long)	1/1
		transmitErrors (long)	1/1
		transmitPackets (long)	1/1
		receiveDrops (long)	1/1
		receiveOverRunError (long)	1/1
		receiveErrors (long)	1/1
		nodeConnector (nodeConnector)	0/1
		transmitDrops (long)	1/1
		transmitBytes (long)	1/1
		collisionCount (long)	1/1
		receiveCrcError (long)	1/1
		receiveFrameError (long)	1/1
nodeFlowStatistics	flowStatistics	node (node)	0/1
		flowStatistic (flowOnNode)	0/unbounded
nodePortStatistics	portStatistics	node (node)	0/1
		portStatistic (nodeConnectorStatistics)	0/unbounded
nodeTable	nodeTable	id (string)	0/1
		node (node)	0/1
nodeTableStatistics	tableStatistics	node (node)	0/1
		tableStatistic (nodeTableStatistics)	0/unbounded
	nodeTableStatistics	maximumEntries (int)	1/1
		name (string)	0/1
		matchedCount (long)	1/1
		activeCount (int)	1/1
		lookupCount (long)	1/1
		nodeTable (nodeTable)	0/1

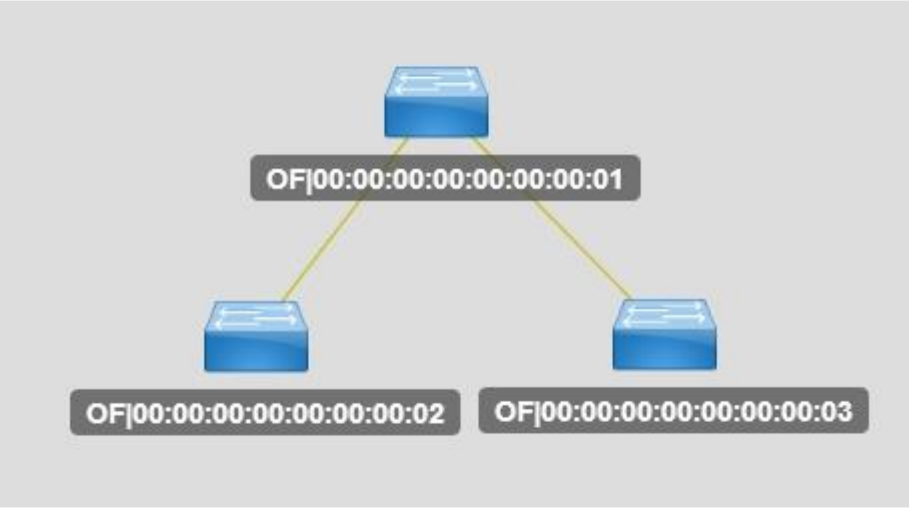
表 4.2-5 Statistics REST APIs

5.3 实验分析

5.3.1 实验拓扑

```
mn --topo tree,depth=2,fanout=2 --controller=remote,ip=192.168.5.15,port=6633
```

即产生如下拓扑：


OF|00:00:00:00:00:00:00:01
OF|00:00:00:00:00:00:00:02
OF|00:00:00:00:00:00:00:03

5.3.2 抓包情况分析

统计模块分别统计了三个交换机上的流的状况。除了一些交换机的基本信息外还会返回一些南向接口向上公开的统计信息，故可以调用 SAL 上的 reader 包中的类获取更详细的数据。

Flow 信息包括：

getByteCount()

getDurationNanoseconds()

getDurationSeconds()

getPacketCount()

getTableId()

getFlow()

Port 信息包括：

getReceivePacketCount

getTransmitPacketCount

getReceiveByteCount

getTransmitByteCount

getReceiveDropCount

getTransmitDropCount

getReceiveErrorCount

getTransmitErrorCount

getReceiveFrameErrorCount

getReceiveOverRunErrorCount

getReceiveCRCErrorCount

getCollisionCount

Table 信息包括：

getName()

getActiveCount()

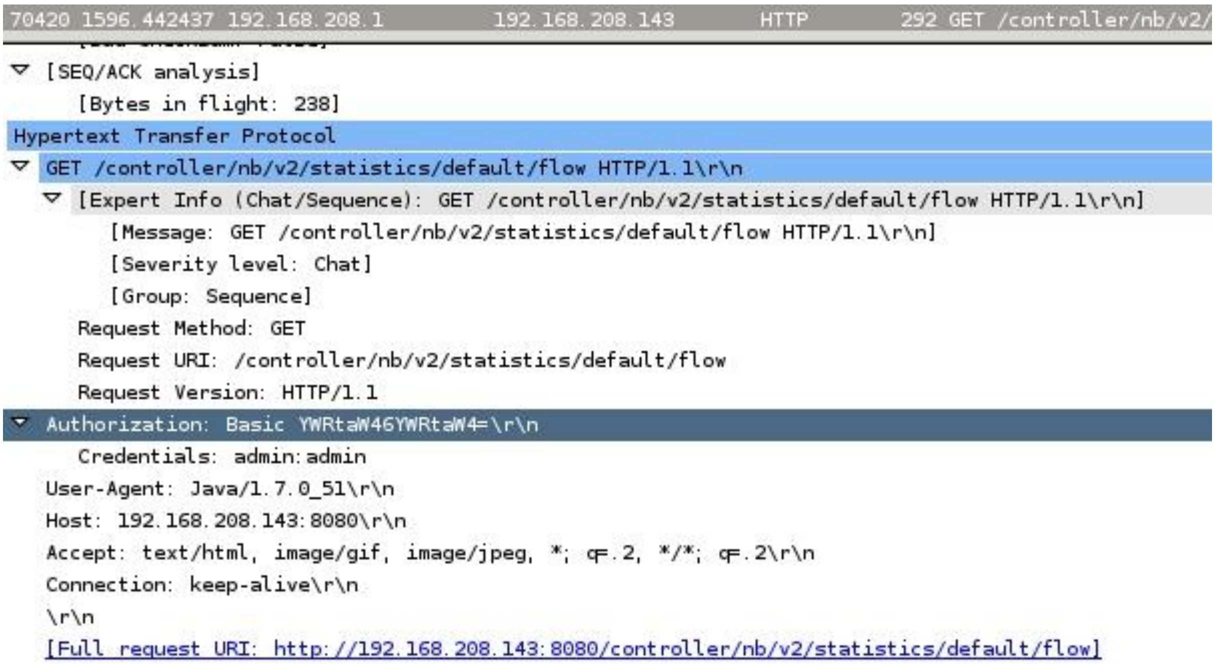
getLookupCount()

getLookupCount()

getMaximumEntries()：默认 10000000

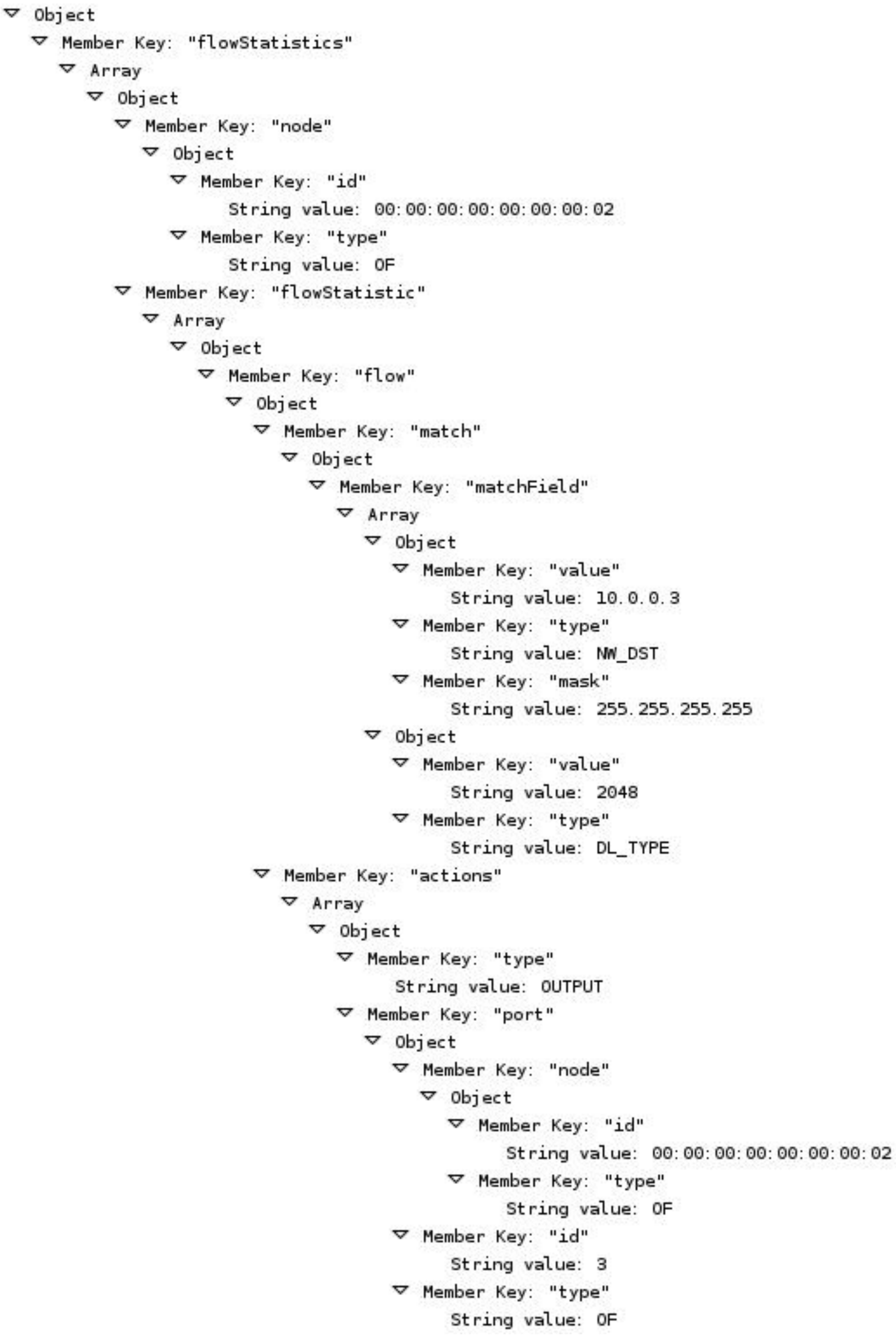
1）获取流统计信息

因之前在 mininet 中做过一次平 all 操作，所以这里面的流表略显多。

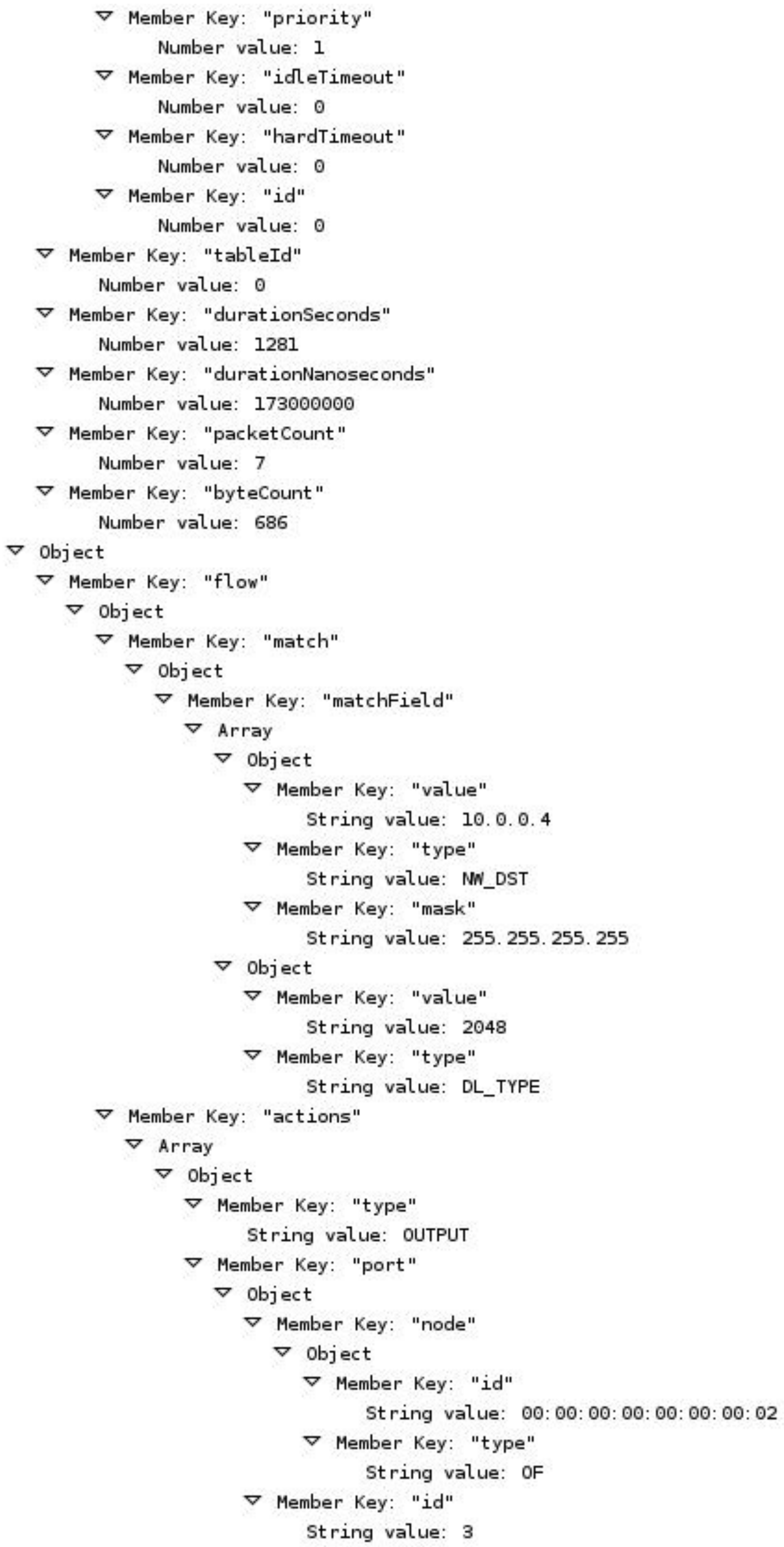


```
70420 1596.442437 192.168.208.1          192.168.208.143       HTTP        292 GET /controller/nb/v2/
▽ [SEQ/ACK analysis]
    [Bytes in flight: 238]
Hypertext Transfer Protocol
▽ GET /controller/nb/v2/statistics/default/flow HTTP/1.1\r\n
  ▽ [Expert Info (Chat/Sequence): GET /controller/nb/v2/statistics/default/flow HTTP/1.1\r\n]
      [Message: GET /controller/nb/v2/statistics/default/flow HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /controller/nb/v2/statistics/default/flow
    Request Version: HTTP/1.1
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.208.143:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.208.143:8080/controller/nb/v2/statistics/default/flow]
```

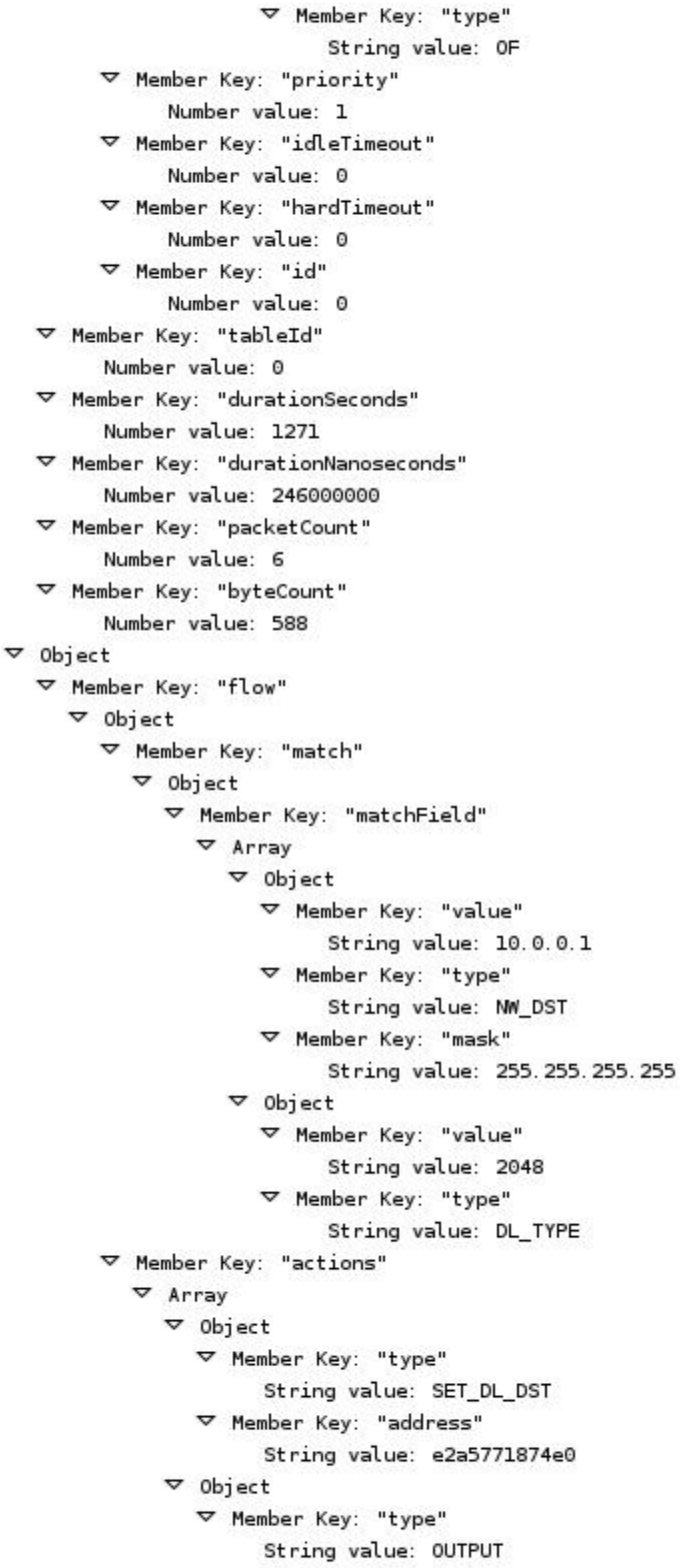
客户端发出获取流统计的请求。



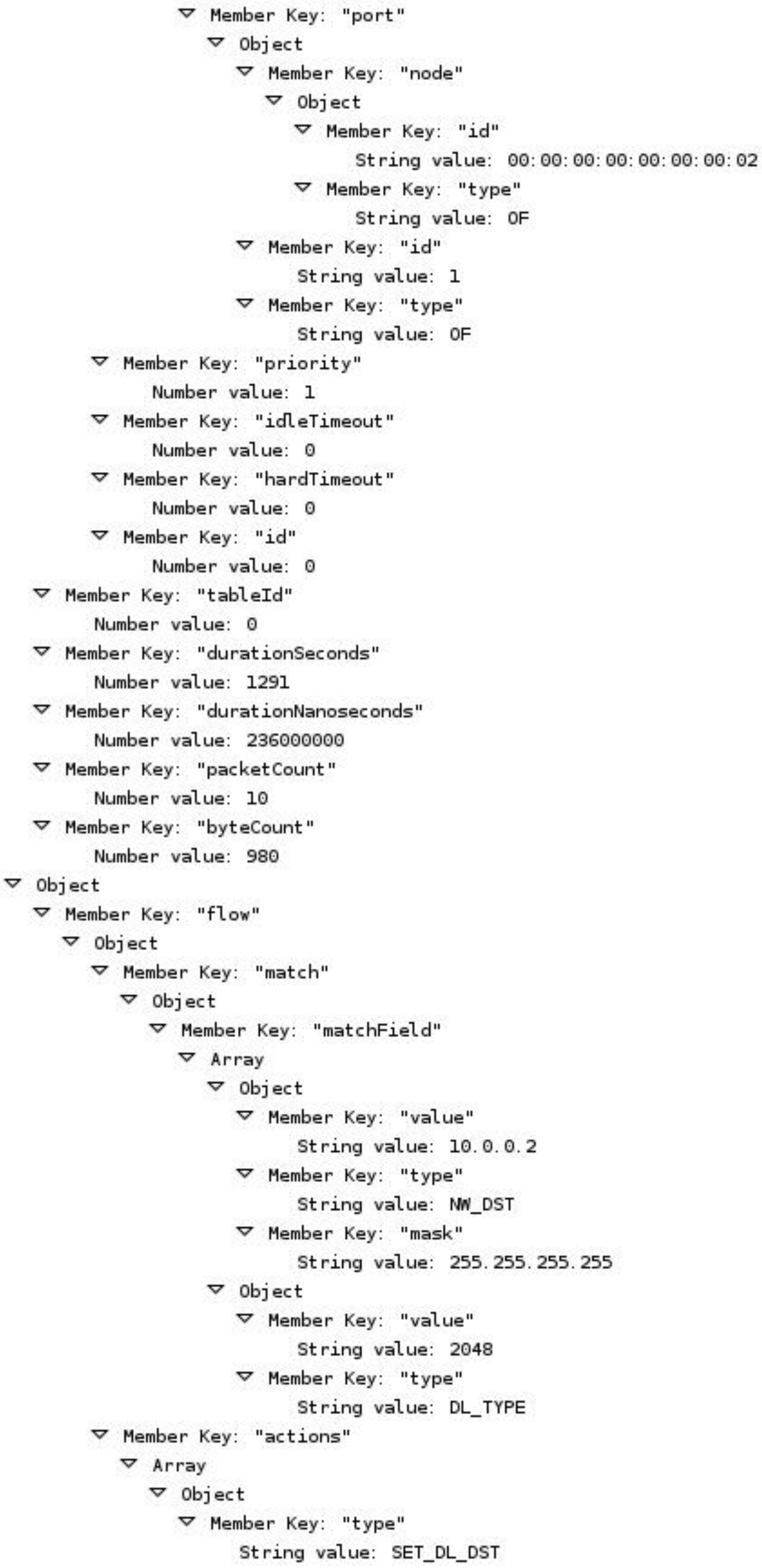
```
▽ Object
  ▽ Member Key: "flowStatistics"
    ▽ Array
      ▽ Object
        ▽ Member Key: "node"
          ▽ Object
            ▽ Member Key: "id"
                String value: 00:00:00:00:00:00:00:02
            ▽ Member Key: "type"
                String value: OF
        ▽ Member Key: "flowStatistic"
          ▽ Array
            ▽ Object
              ▽ Member Key: "flow"
                ▽ Object
                  ▽ Member Key: "match"
                    ▽ Object
                      ▽ Member Key: "matchField"
                        ▽ Array
                          ▽ Object
                            ▽ Member Key: "value"
                                String value: 10.0.0.3
                            ▽ Member Key: "type"
                                String value: NW_DST
                            ▽ Member Key: "mask"
                                String value: 255.255.255.255
                          ▽ Object
                            ▽ Member Key: "value"
                                String value: 2048
                            ▽ Member Key: "type"
                                String value: DL_TYPE
                  ▽ Member Key: "actions"
                    ▽ Array
                      ▽ Object
                        ▽ Member Key: "type"
                            String value: OUTPUT
                        ▽ Member Key: "port"
                          ▽ Object
                            ▽ Member Key: "node"
                              ▽ Object
                                ▽ Member Key: "id"
                                    String value: 00:00:00:00:00:00:00:02
                                ▽ Member Key: "type"
                                    String value: OF
                            ▽ Member Key: "id"
                                String value: 3
                            ▽ Member Key: "type"
                                String value: OF
```



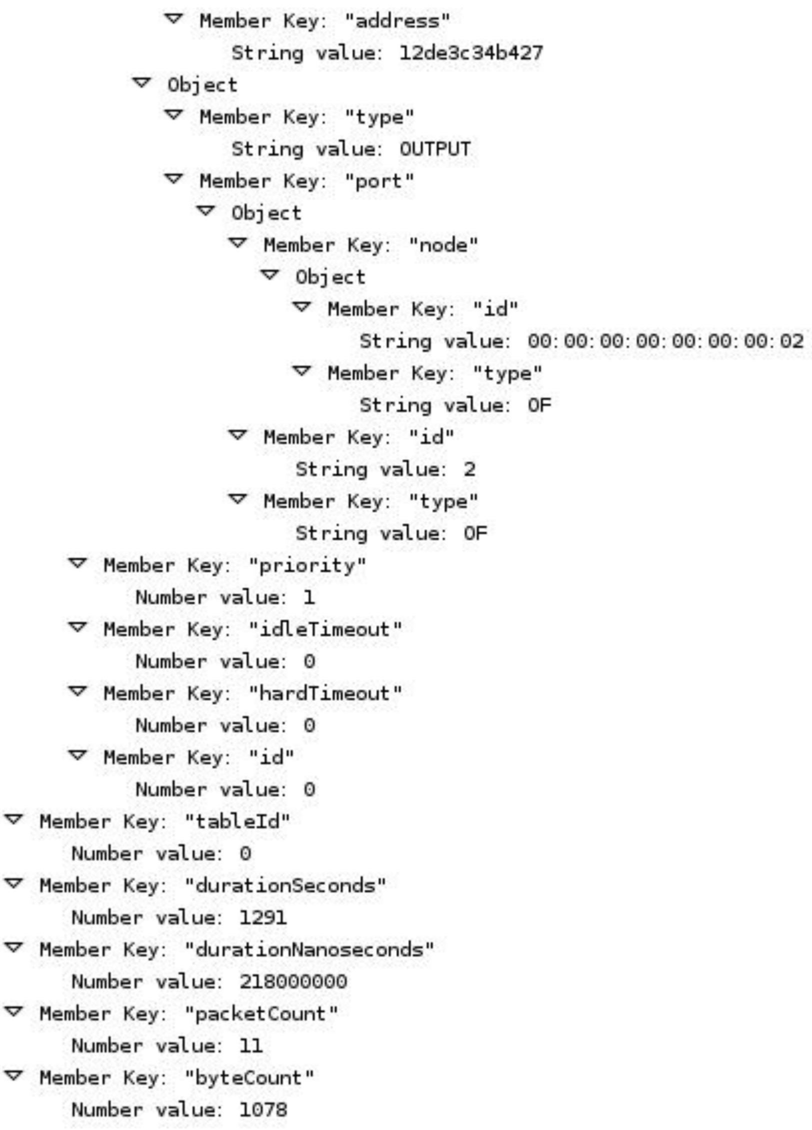
```
      ▽ Member Key: "priority"
          Number value: 1
      ▽ Member Key: "idleTimeout"
          Number value: 0
      ▽ Member Key: "hardTimeout"
          Number value: 0
      ▽ Member Key: "id"
          Number value: 0
  ▽ Member Key: "tableId"
      Number value: 0
  ▽ Member Key: "durationSeconds"
      Number value: 1281
  ▽ Member Key: "durationNanoseconds"
      Number value: 173000000
  ▽ Member Key: "packetCount"
      Number value: 7
  ▽ Member Key: "byteCount"
      Number value: 686
▽ Object
  ▽ Member Key: "flow"
    ▽ Object
      ▽ Member Key: "match"
        ▽ Object
          ▽ Member Key: "matchField"
            ▽ Array
              ▽ Object
                ▽ Member Key: "value"
                    String value: 10.0.0.4
                ▽ Member Key: "type"
                    String value: NW_DST
                ▽ Member Key: "mask"
                    String value: 255.255.255.255
              ▽ Object
                ▽ Member Key: "value"
                    String value: 2048
                ▽ Member Key: "type"
                    String value: DL_TYPE
      ▽ Member Key: "actions"
        ▽ Array
          ▽ Object
            ▽ Member Key: "type"
                String value: OUTPUT
            ▽ Member Key: "port"
              ▽ Object
                ▽ Member Key: "node"
                  ▽ Object
                    ▽ Member Key: "id"
                        String value: 00:00:00:00:00:00:00:02
                    ▽ Member Key: "type"
                        String value: OF
                ▽ Member Key: "id"
                    String value: 3
```



```
                ▽ Member Key: "type"
                    String value: OF
      ▽ Member Key: "priority"
          Number value: 1
      ▽ Member Key: "idleTimeout"
          Number value: 0
      ▽ Member Key: "hardTimeout"
          Number value: 0
      ▽ Member Key: "id"
          Number value: 0
  ▽ Member Key: "tableId"
      Number value: 0
  ▽ Member Key: "durationSeconds"
      Number value: 1271
  ▽ Member Key: "durationNanoseconds"
      Number value: 246000000
  ▽ Member Key: "packetCount"
      Number value: 6
  ▽ Member Key: "byteCount"
      Number value: 588
▽ Object
  ▽ Member Key: "flow"
    ▽ Object
      ▽ Member Key: "match"
        ▽ Object
          ▽ Member Key: "matchField"
            ▽ Array
              ▽ Object
                ▽ Member Key: "value"
                    String value: 10.0.0.1
                ▽ Member Key: "type"
                    String value: NW_DST
                ▽ Member Key: "mask"
                    String value: 255.255.255.255
              ▽ Object
                ▽ Member Key: "value"
                    String value: 2048
                ▽ Member Key: "type"
                    String value: DL_TYPE
      ▽ Member Key: "actions"
        ▽ Array
          ▽ Object
            ▽ Member Key: "type"
                String value: SET_DL_DST
            ▽ Member Key: "address"
                String value: e2a5771874e0
          ▽ Object
            ▽ Member Key: "type"
                String value: OUTPUT
```

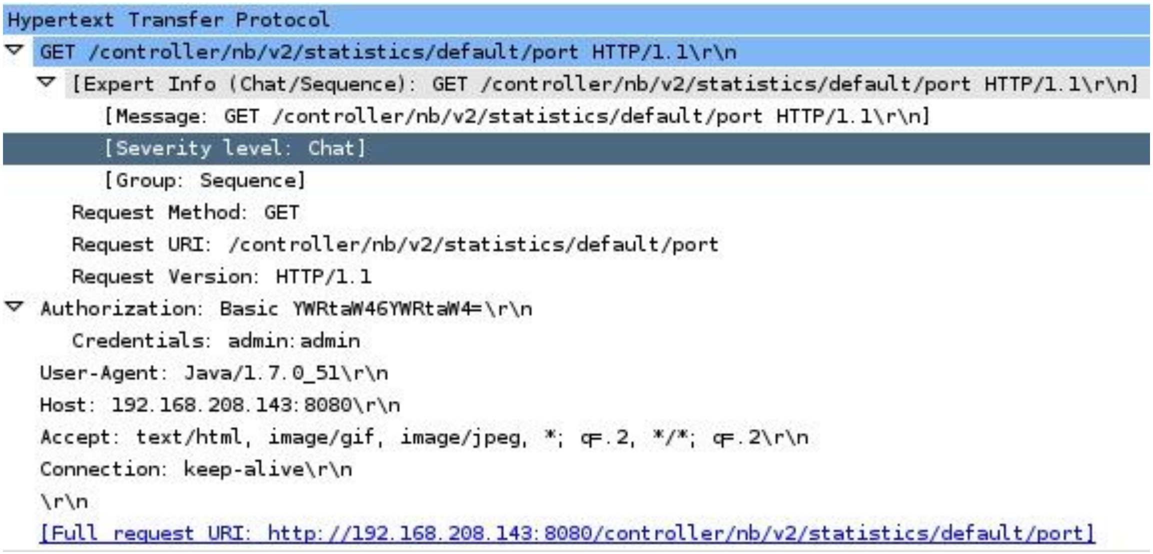


```
            ▽ Member Key: "port"
              ▽ Object
                ▽ Member Key: "node"
                  ▽ Object
                    ▽ Member Key: "id"
                        String value: 00:00:00:00:00:00:00:02
                    ▽ Member Key: "type"
                        String value: OF
                ▽ Member Key: "id"
                    String value: 1
                ▽ Member Key: "type"
                    String value: OF
        ▽ Member Key: "priority"
            Number value: 1
        ▽ Member Key: "idleTimeout"
            Number value: 0
        ▽ Member Key: "hardTimeout"
            Number value: 0
        ▽ Member Key: "id"
            Number value: 0
  ▽ Member Key: "tableId"
      Number value: 0
  ▽ Member Key: "durationSeconds"
      Number value: 1291
  ▽ Member Key: "durationNanoseconds"
      Number value: 236000000
  ▽ Member Key: "packetCount"
      Number value: 10
  ▽ Member Key: "byteCount"
      Number value: 980
▽ Object
  ▽ Member Key: "flow"
    ▽ Object
      ▽ Member Key: "match"
        ▽ Object
          ▽ Member Key: "matchField"
            ▽ Array
              ▽ Object
                ▽ Member Key: "value"
                    String value: 10.0.0.2
                ▽ Member Key: "type"
                    String value: NW_DST
                ▽ Member Key: "mask"
                    String value: 255.255.255.255
              ▽ Object
                ▽ Member Key: "value"
                    String value: 2048
                ▽ Member Key: "type"
                    String value: DL_TYPE
      ▽ Member Key: "actions"
        ▽ Array
          ▽ Object
            ▽ Member Key: "type"
                String value: SET_DL_DST
```

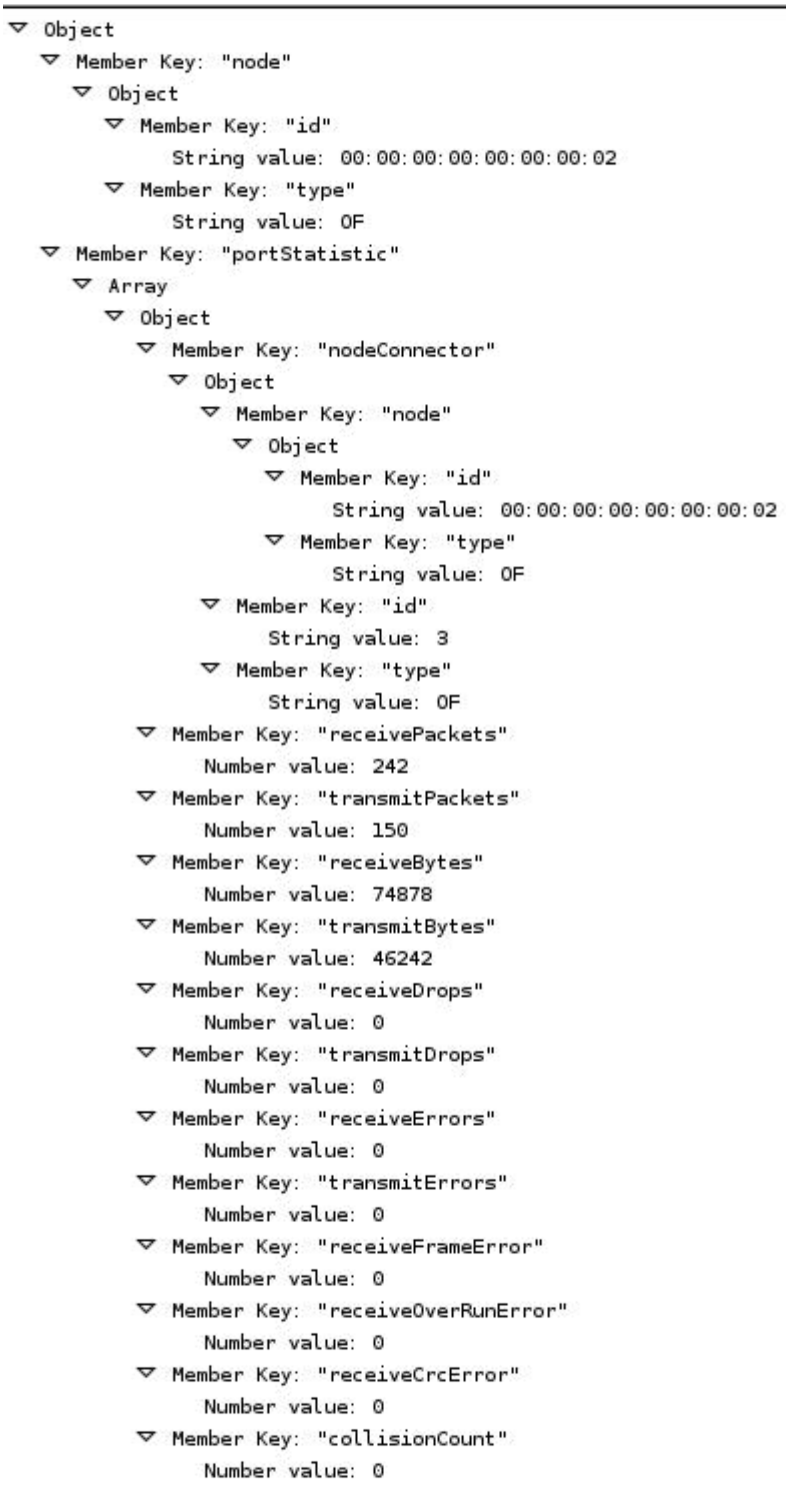


```
          ▽ Member Key: "address"
              String value: 12de3c34b427
        ▽ Object
          ▽ Member Key: "type"
              String value: OUTPUT
          ▽ Member Key: "port"
            ▽ Object
              ▽ Member Key: "node"
                ▽ Object
                  ▽ Member Key: "id"
                      String value: 00:00:00:00:00:00:00:02
                  ▽ Member Key: "type"
                      String value: OF
              ▽ Member Key: "id"
                  String value: 2
              ▽ Member Key: "type"
                  String value: OF
    ▽ Member Key: "priority"
        Number value: 1
    ▽ Member Key: "idleTimeout"
        Number value: 0
    ▽ Member Key: "hardTimeout"
        Number value: 0
    ▽ Member Key: "id"
        Number value: 0
▽ Member Key: "tableId"
    Number value: 0
▽ Member Key: "durationSeconds"
    Number value: 1291
▽ Member Key: "durationNanoseconds"
    Number value: 218000000
▽ Member Key: "packetCount"
    Number value: 11
▽ Member Key: "byteCount"
    Number value: 1078
```

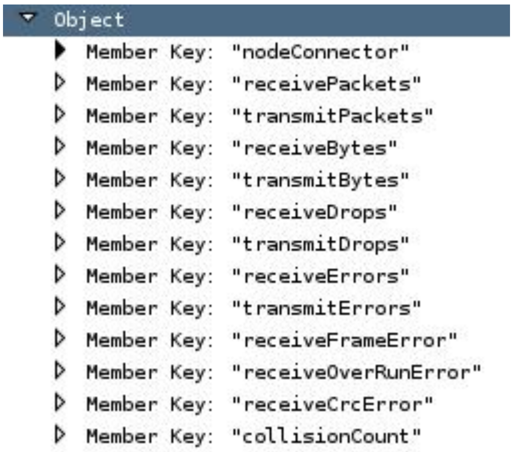
2）获取端口统计信息



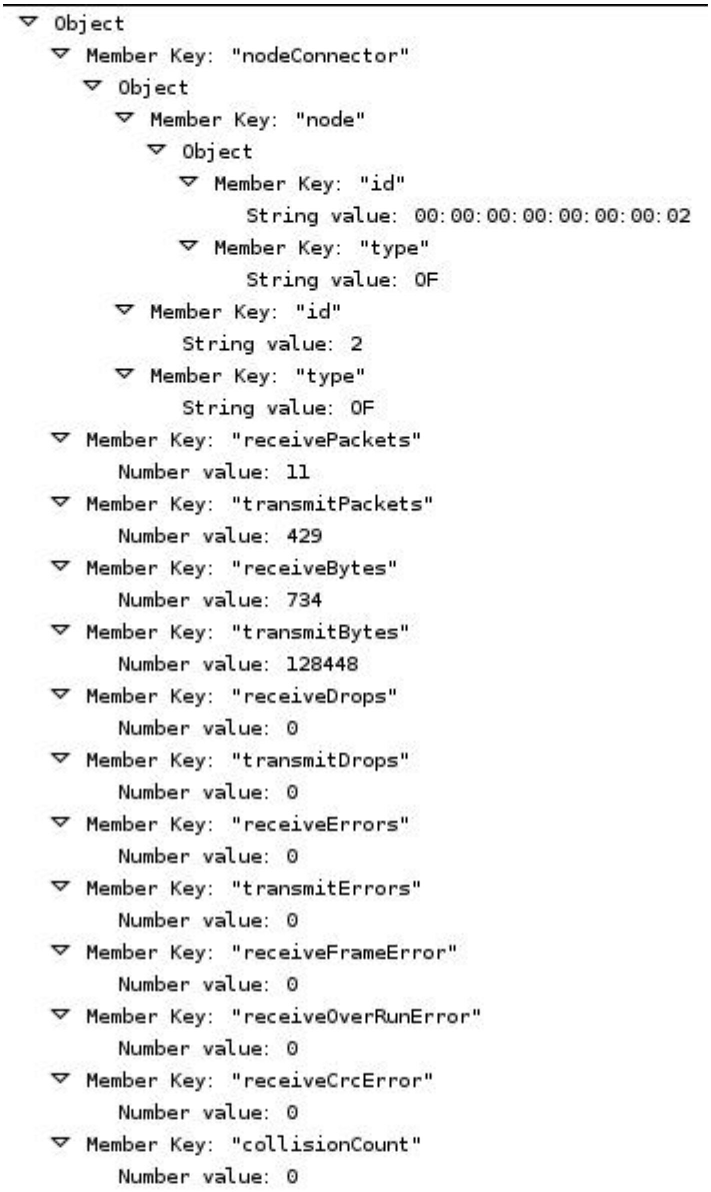
```
Hypertext Transfer Protocol
▽ GET /controller/nb/v2/statistics/default/port HTTP/1.1\r\n
  ▽ [Expert Info (Chat/Sequence): GET /controller/nb/v2/statistics/default/port HTTP/1.1\r\n]
      [Message: GET /controller/nb/v2/statistics/default/port HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /controller/nb/v2/statistics/default/port
    Request Version: HTTP/1.1
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.208.143:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.208.143:8080/controller/nb/v2/statistics/default/port]
```



```
▽ Object
  ▽ Member Key: "node"
    ▽ Object
      ▽ Member Key: "id"
          String value: 00:00:00:00:00:00:00:02
      ▽ Member Key: "type"
          String value: OF
  ▽ Member Key: "portStatistic"
    ▽ Array
      ▽ Object
        ▽ Member Key: "nodeConnector"
          ▽ Object
            ▽ Member Key: "node"
              ▽ Object
                ▽ Member Key: "id"
                    String value: 00:00:00:00:00:00:00:02
                ▽ Member Key: "type"
                    String value: OF
            ▽ Member Key: "id"
                String value: 3
            ▽ Member Key: "type"
                String value: OF
        ▽ Member Key: "receivePackets"
            Number value: 242
        ▽ Member Key: "transmitPackets"
            Number value: 150
        ▽ Member Key: "receiveBytes"
            Number value: 74878
        ▽ Member Key: "transmitBytes"
            Number value: 46242
        ▽ Member Key: "receiveDrops"
            Number value: 0
        ▽ Member Key: "transmitDrops"
            Number value: 0
        ▽ Member Key: "receiveErrors"
            Number value: 0
        ▽ Member Key: "transmitErrors"
            Number value: 0
        ▽ Member Key: "receiveFrameError"
            Number value: 0
        ▽ Member Key: "receiveOverRunError"
            Number value: 0
        ▽ Member Key: "receiveCrcError"
            Number value: 0
        ▽ Member Key: "collisionCount"
            Number value: 0
```

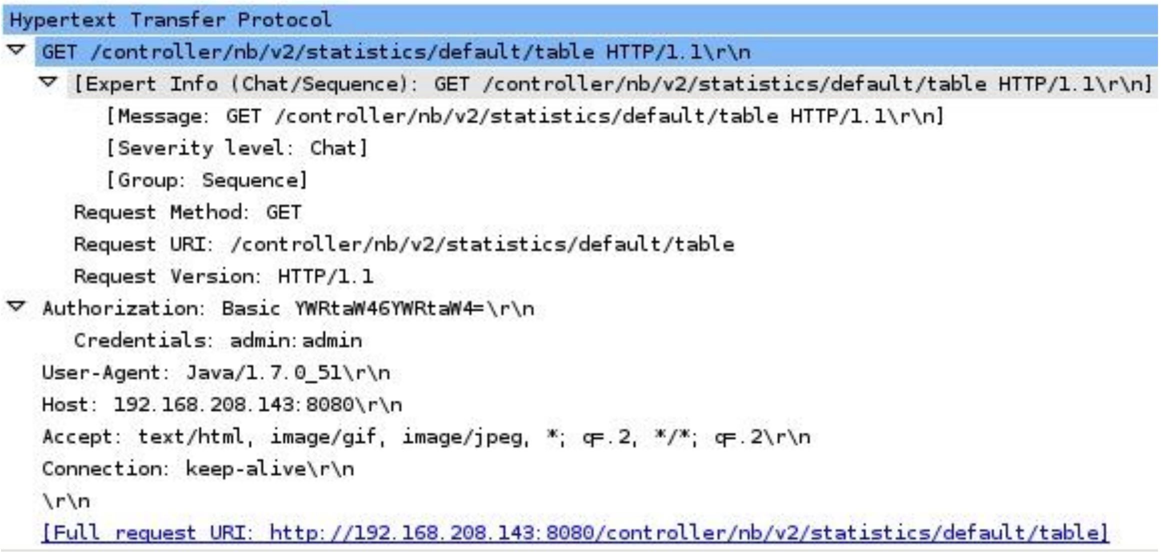


```
▽ Object
    ▶ Member Key: "nodeConnector"
    ▷ Member Key: "receivePackets"
    ▷ Member Key: "transmitPackets"
    ▷ Member Key: "receiveBytes"
    ▷ Member Key: "transmitBytes"
    ▷ Member Key: "receiveDrops"
    ▷ Member Key: "transmitDrops"
    ▷ Member Key: "receiveErrors"
    ▷ Member Key: "transmitErrors"
    ▷ Member Key: "receiveFrameError"
    ▷ Member Key: "receiveOverRunError"
    ▷ Member Key: "receiveCrcError"
    ▷ Member Key: "collisionCount"
```



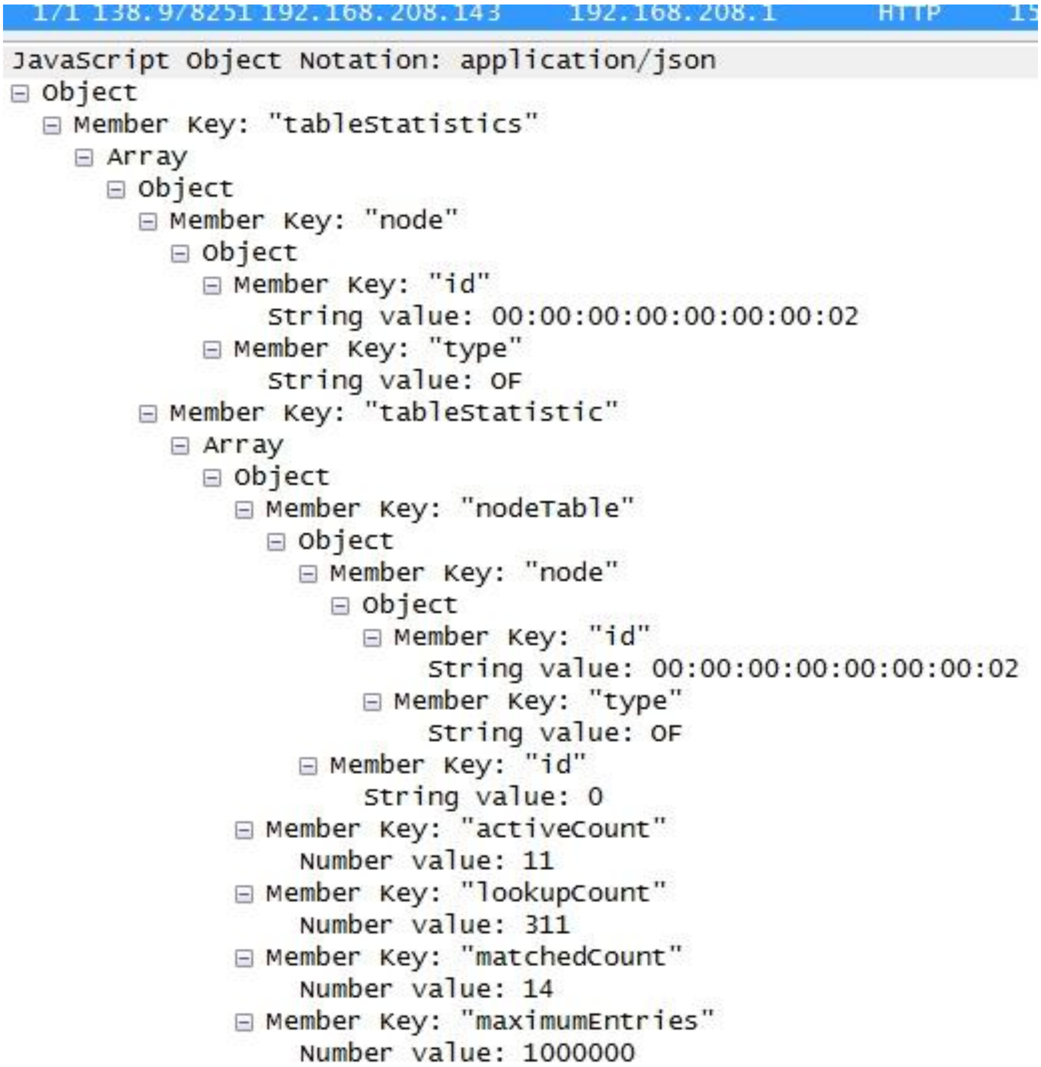
```
▽ Object
    ▽ Member Key: "nodeConnector"
        ▽ Object
            ▽ Member Key: "node"
                ▽ Object
                    ▽ Member Key: "id"
                          String value: 00:00:00:00:00:00:00:02
                    ▽ Member Key: "type"
                          String value: OF
            ▽ Member Key: "id"
                  String value: 2
            ▽ Member Key: "type"
                  String value: OF
    ▽ Member Key: "receivePackets"
          Number value: 11
    ▽ Member Key: "transmitPackets"
          Number value: 429
    ▽ Member Key: "receiveBytes"
          Number value: 734
    ▽ Member Key: "transmitBytes"
          Number value: 128448
    ▽ Member Key: "receiveDrops"
          Number value: 0
    ▽ Member Key: "transmitDrops"
          Number value: 0
    ▽ Member Key: "receiveErrors"
          Number value: 0
    ▽ Member Key: "transmitErrors"
          Number value: 0
    ▽ Member Key: "receiveFrameError"
          Number value: 0
    ▽ Member Key: "receiveOverRunError"
          Number value: 0
    ▽ Member Key: "receiveCrcError"
          Number value: 0
    ▽ Member Key: "collisionCount"
          Number value: 0
```

3）获取流表信息



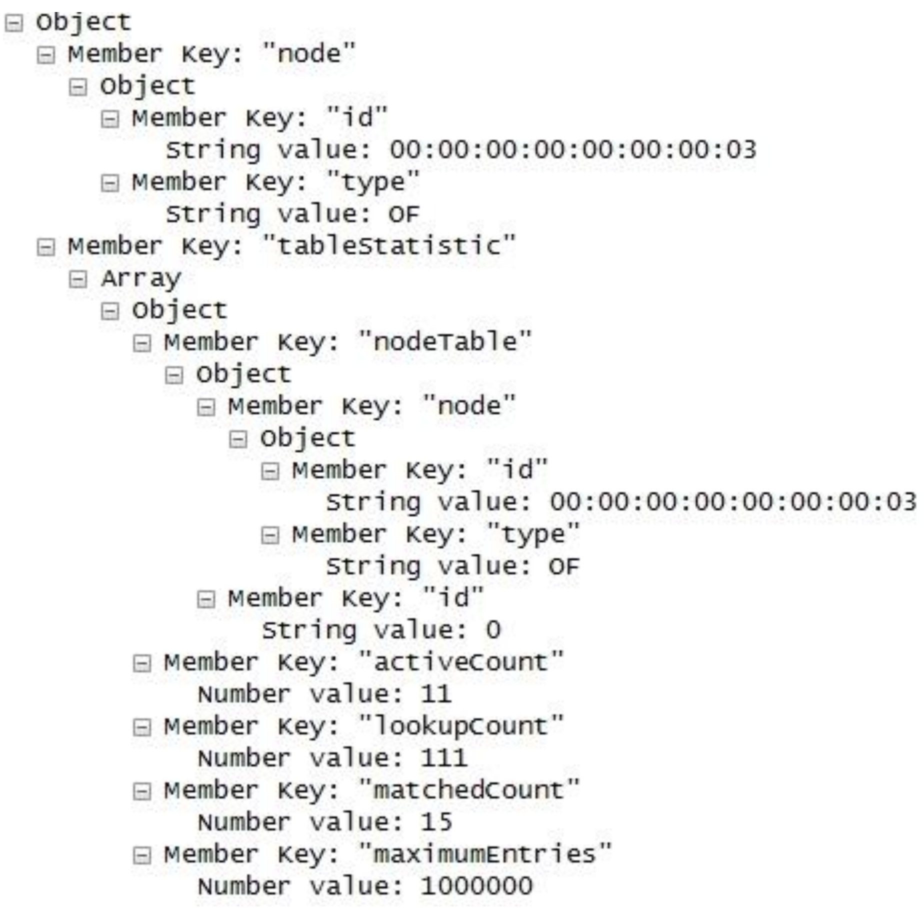
```
Hypertext Transfer Protocol
▽ GET /controller/nb/v2/statistics/default/table HTTP/1.1\r\n
  ▽ [Expert Info (Chat/Sequence): GET /controller/nb/v2/statistics/default/table HTTP/1.1\r\n]
      [Message: GET /controller/nb/v2/statistics/default/table HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /controller/nb/v2/statistics/default/table
    Request Version: HTTP/1.1
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.208.143:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.208.143:8080/controller/nb/v2/statistics/default/table]
```

客户端发出获取流表信息的请求。



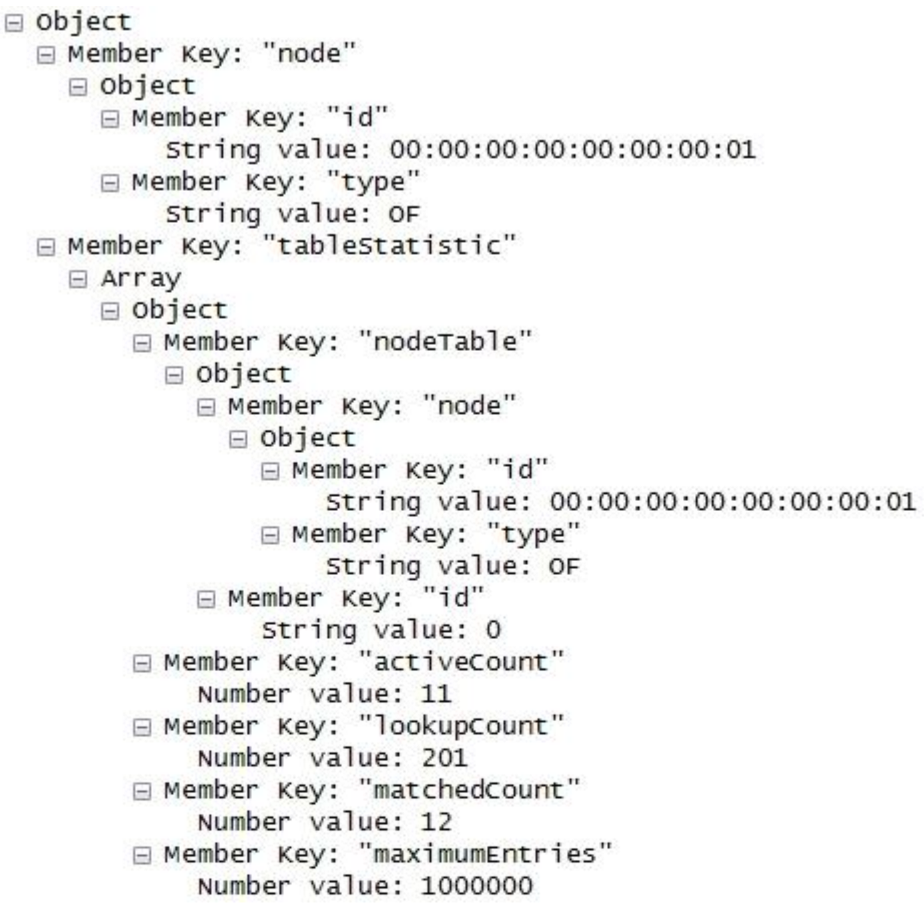
```
  171 138.978251 192.168.208.143     192.168.208.1       HTTP     15
JavaScript Object Notation: application/json
⊟ Object
  ⊟ Member Key: "tableStatistics"
    ⊟ Array
      ⊟ Object
        ⊟ Member Key: "node"
          ⊟ Object
            ⊟ Member Key: "id"
                String value: 00:00:00:00:00:00:00:02
            ⊟ Member Key: "type"
                String value: OF
        ⊟ Member Key: "tableStatistic"
          ⊟ Array
            ⊟ Object
              ⊟ Member Key: "nodeTable"
                ⊟ Object
                  ⊟ Member Key: "node"
                    ⊟ Object
                      ⊟ Member Key: "id"
                          String value: 00:00:00:00:00:00:00:02
                      ⊟ Member Key: "type"
                          String value: OF
                  ⊟ Member Key: "id"
                      String value: 0
              ⊟ Member Key: "activeCount"
                  Number value: 11
              ⊟ Member Key: "lookupCount"
                  Number value: 311
              ⊟ Member Key: "matchedCount"
                  Number value: 14
              ⊟ Member Key: "maximumEntries"
                  Number value: 1000000
```

交换机2的流表



```
⊟ Object
  ⊟ Member Key: "node"
    ⊟ Object
      ⊟ Member Key: "id"
          String value: 00:00:00:00:00:00:00:03
      ⊟ Member Key: "type"
          String value: OF
  ⊟ Member Key: "tableStatistic"
    ⊟ Array
      ⊟ Object
        ⊟ Member Key: "nodeTable"
          ⊟ Object
            ⊟ Member Key: "node"
              ⊟ Object
                ⊟ Member Key: "id"
                    String value: 00:00:00:00:00:00:00:03
                ⊟ Member Key: "type"
                    String value: OF
            ⊟ Member Key: "id"
                String value: 0
        ⊟ Member Key: "activeCount"
            Number value: 11
        ⊟ Member Key: "lookupCount"
            Number value: 111
        ⊟ Member Key: "matchedCount"
            Number value: 15
        ⊟ Member Key: "maximumEntries"
            Number value: 1000000
```

交换机3的流表



```
⊟ Object
  ⊟ Member Key: "node"
    ⊟ Object
      ⊟ Member Key: "id"
          String value: 00:00:00:00:00:00:00:01
      ⊟ Member Key: "type"
          String value: OF
  ⊟ Member Key: "tableStatistic"
    ⊟ Array
      ⊟ Object
        ⊟ Member Key: "nodeTable"
          ⊟ Object
            ⊟ Member Key: "node"
              ⊟ Object
                ⊟ Member Key: "id"
                    String value: 00:00:00:00:00:00:00:01
                ⊟ Member Key: "type"
                    String value: OF
            ⊟ Member Key: "id"
                String value: 0
        ⊟ Member Key: "activeCount"
            Number value: 11
        ⊟ Member Key: "lookupCount"
            Number value: 201
        ⊟ Member Key: "matchedCount"
            Number value: 12
        ⊟ Member Key: "maximumEntries"
            Number value: 1000000
```

交换机1的流表

服务端返回流表信息，由于每个交换机中的流表项默认有255项，服务端会把便利后的结果全数返回，所以数据量会很大，但真正有用的数据却不多，故取其中一个交换机的一项进行解析。

6 Subnets REST APIs

本部分 APIs 的应用见 SubnetsNorthbound
本部分 APIs 用来管理子网。

6.1 Subnets RESTAPIs 可提供的服务：

请求分类	客户端请求	服务器响应
subnets	GET:列出给定范围的所有子网	list
subnet	GET:列出给定区域一个子网的配置	subnetConfig
	PUT：增加一个子网到指定的区域，节点连接器可选	subnetConfig
	DELETE：从指定区域删除子网	(custom)
	POST：修改一个子网。用新的代替旧的。只有端口的列表可以修	subnetConfig

	改。若不存在则新建子网。	

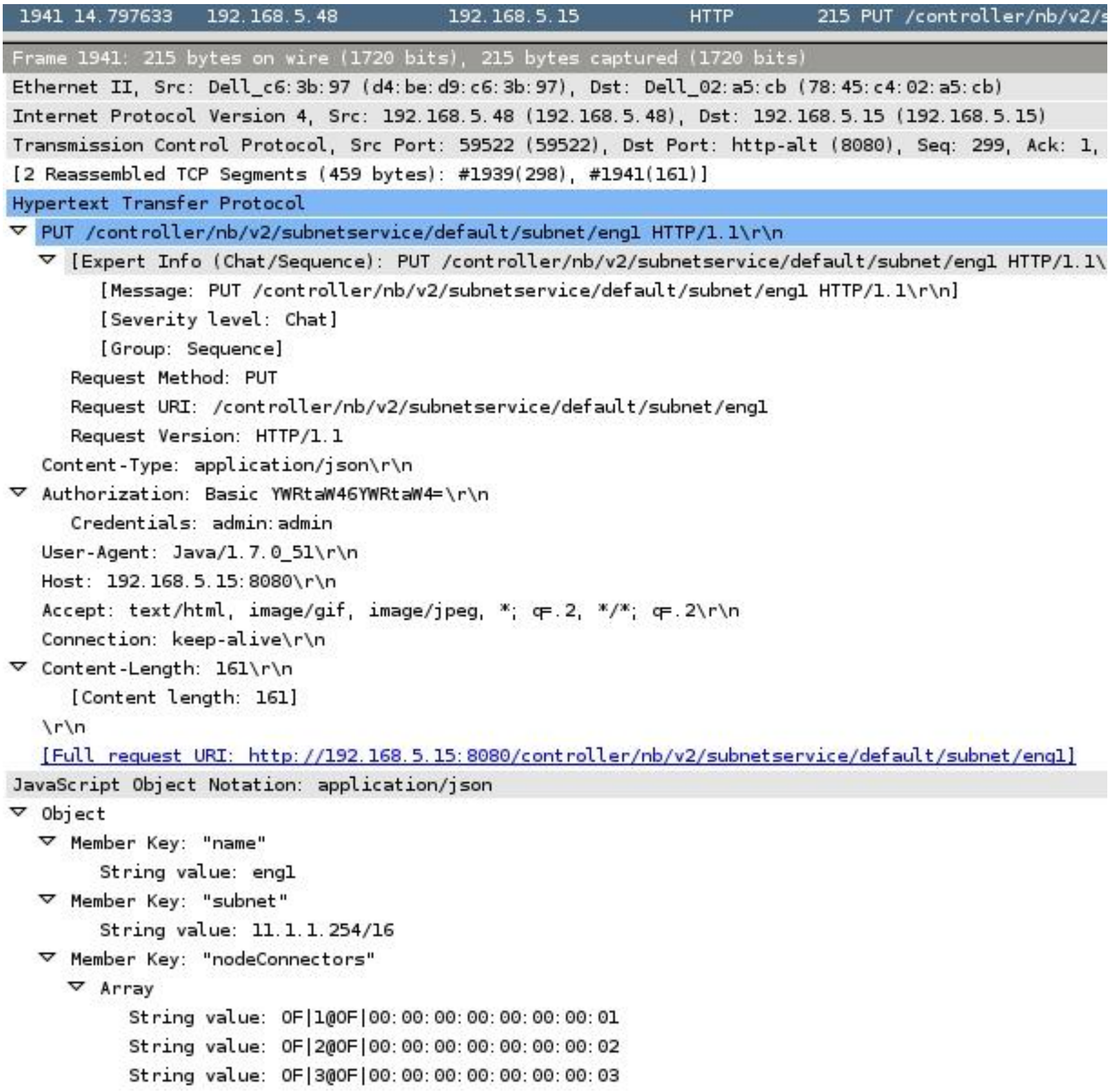
6.2 Subnets REST APIs 数据模型

Data Elements	**Data Types**	**XML Elements**	
		名称（类型）	**最大/最小 出现**
list	subnetConfigs	subnetConfig (subnetConfig)	0/unbounded
subnetConfig	subnetConfig	subnet (string)	0/1
		name (string)	0/1
		nodeConnectors (string)	0/unbounded

6.3 实验分析

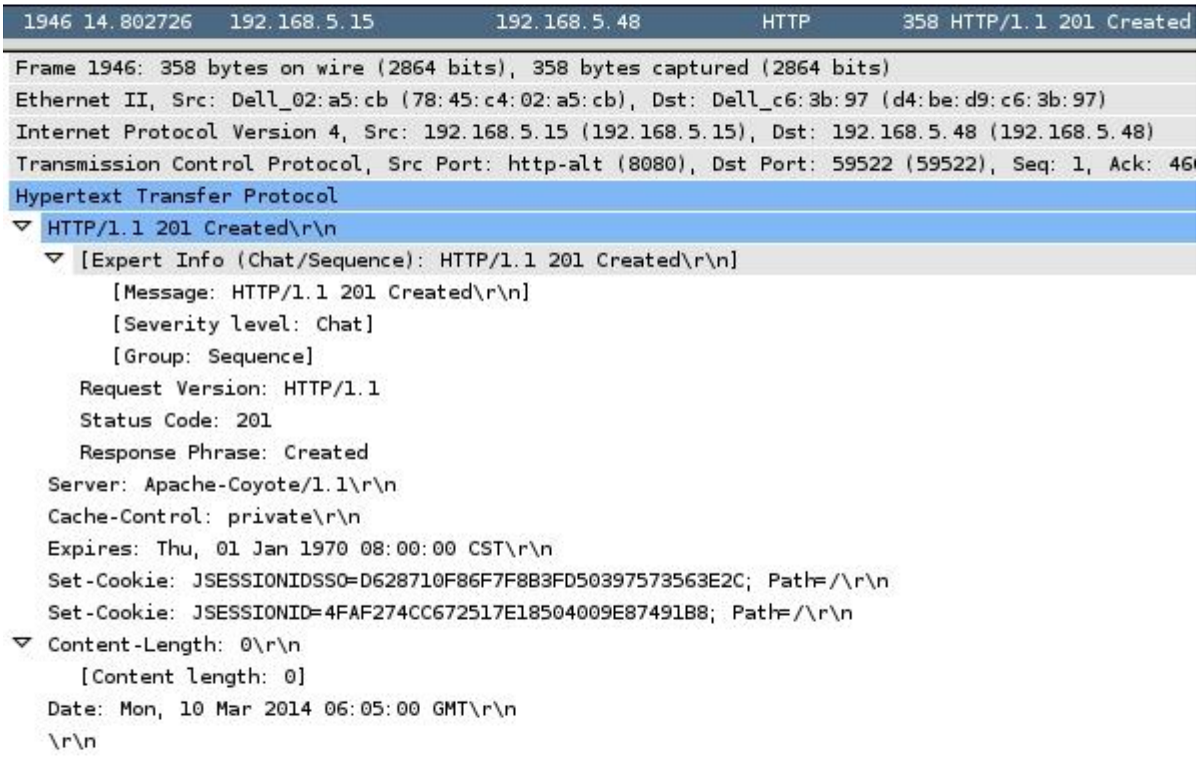
6.3.1 抓包情况分析

1）添加新的子网



```
1941 14.797633   192.168.5.48          192.168.5.15         HTTP        215 PUT /controller/nb/v2/s
Frame 1941: 215 bytes on wire (1720 bits), 215 bytes captured (1720 bits)
Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
Transmission Control Protocol, Src Port: 59522 (59522), Dst Port: http-alt (8080), Seq: 299, Ack: 1,
[2 Reassembled TCP Segments (459 bytes): #1939(298), #1941(161)]
Hypertext Transfer Protocol
▽ PUT /controller/nb/v2/subnetservice/default/subnet/engl HTTP/1.1\r\n
  ▽ [Expert Info (Chat/Sequence): PUT /controller/nb/v2/subnetservice/default/subnet/engl HTTP/1.1\
      [Message: PUT /controller/nb/v2/subnetservice/default/subnet/engl HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: PUT
    Request URI: /controller/nb/v2/subnetservice/default/subnet/engl
    Request Version: HTTP/1.1
  Content-Type: application/json\r\n
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
▽ Content-Length: 161\r\n
    [Content length: 161]
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/subnetservice/default/subnet/engl]
JavaScript Object Notation: application/json
▽ Object
  ▽ Member Key: "name"
      String value: engl
  ▽ Member Key: "subnet"
      String value: 11.1.1.254/16
  ▽ Member Key: "nodeConnectors"
    ▽ Array
        String value: OF|1@OF|00:00:00:00:00:00:00:01
        String value: OF|2@OF|00:00:00:00:00:00:00:02
        String value: OF|3@OF|00:00:00:00:00:00:00:03
```

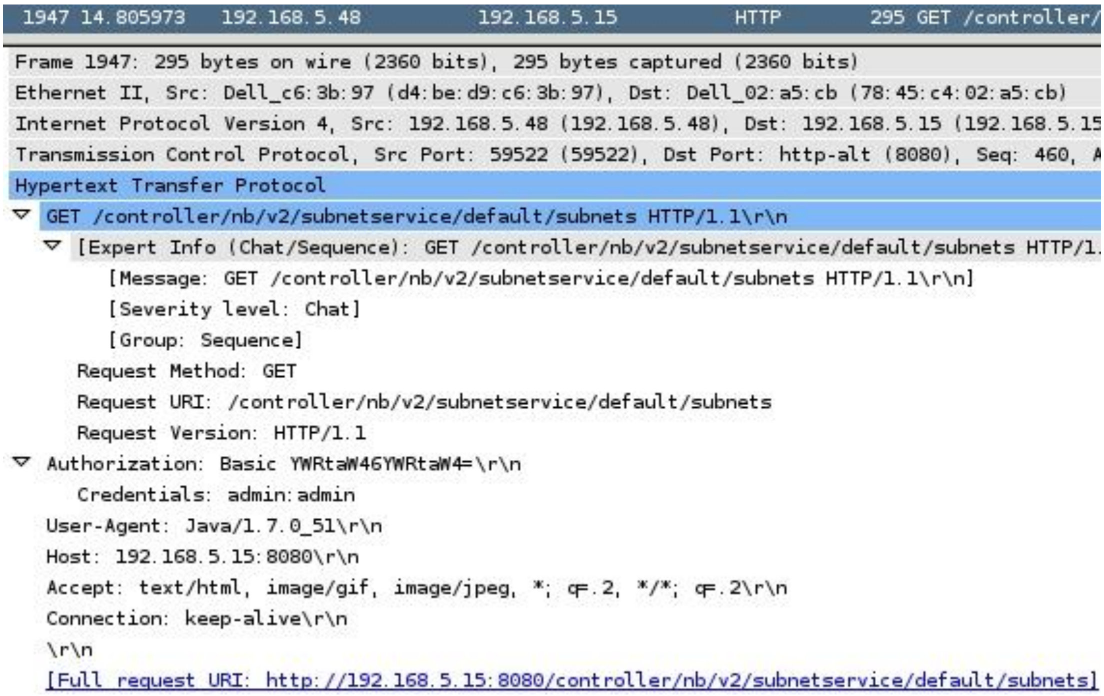
创建子网 11.1.1.254/16，其包括的交换机有 1 号交换机的 1 端口、2 号交换机的 2 端口、3 号交换机的 3 端口。



```
1946 14.802726    192.168.5.15           192.168.5.48          HTTP        358 HTTP/1.1 201 Created
Frame 1946: 358 bytes on wire (2864 bits), 358 bytes captured (2864 bits)
Ethernet II, Src: Dell_02:a5:cb (78:45:c4:02:a5:cb), Dst: Dell_c6:3b:97 (d4:be:d9:c6:3b:97)
Internet Protocol Version 4, Src: 192.168.5.15 (192.168.5.15), Dst: 192.168.5.48 (192.168.5.48)
Transmission Control Protocol, Src Port: http-alt (8080), Dst Port: 59522 (59522), Seq: 1, Ack: 46
Hypertext Transfer Protocol
▽ HTTP/1.1 201 Created\r\n
  ▽ [Expert Info (Chat/Sequence): HTTP/1.1 201 Created\r\n]
      [Message: HTTP/1.1 201 Created\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Version: HTTP/1.1
    Status Code: 201
    Response Phrase: Created
  Server: Apache-Coyote/1.1\r\n
  Cache-Control: private\r\n
  Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
  Set-Cookie: JSESSIONIDSSO=D628710F86F7F8B3FD50397573563E2C; Path=/\r\n
  Set-Cookie: JSESSIONID=4FAF274CC672517E18504009E87491B8; Path=/\r\n
▽ Content-Length: 0\r\n
    [Content length: 0]
  Date: Mon, 10 Mar 2014 06:05:00 GMT\r\n
  \r\n
```

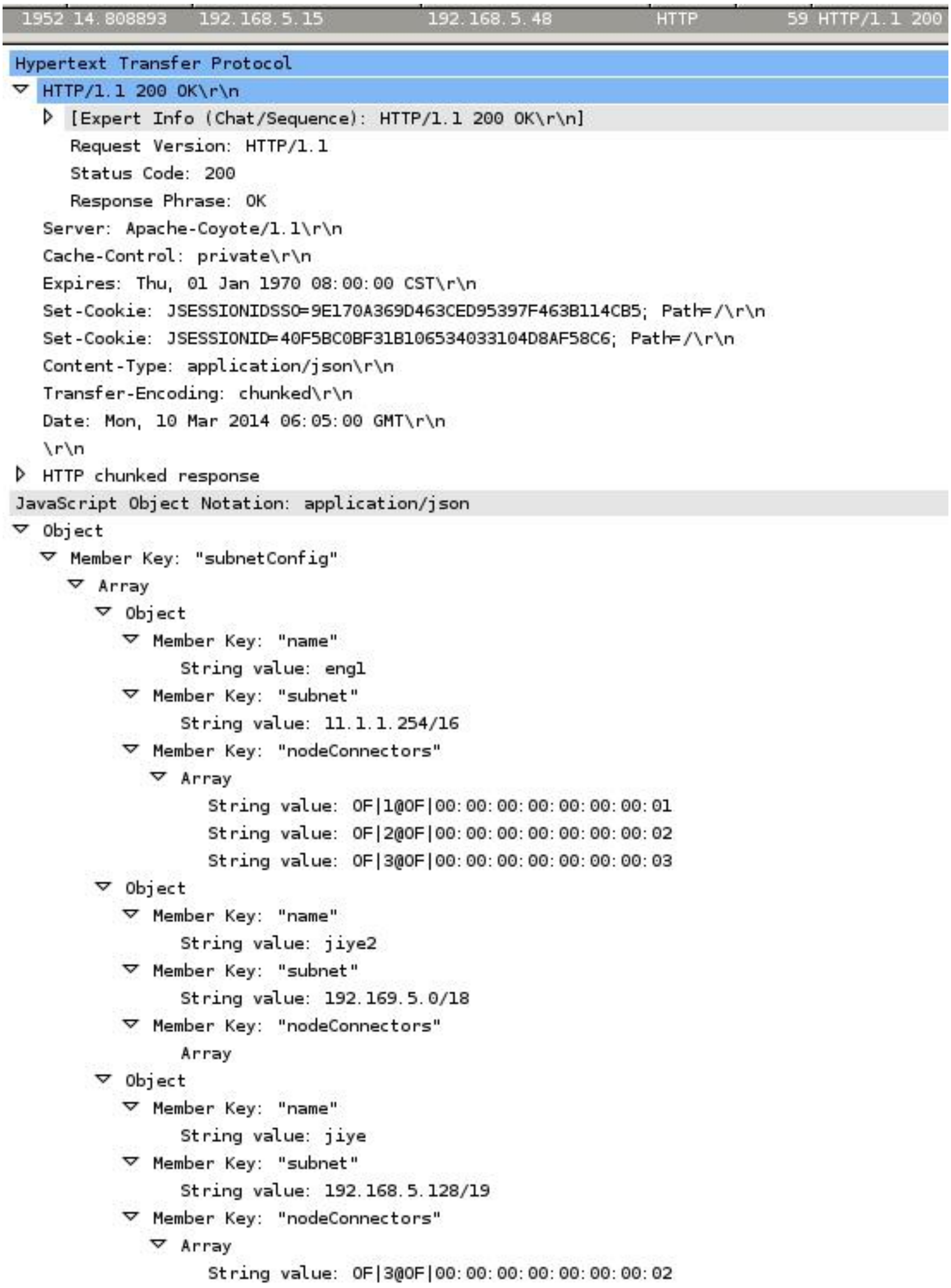
创建成功，返回 201。

2）获取子网信息



```
1947 14.805973    192.168.5.48          192.168.5.15          HTTP       295 GET /controller/
Frame 1947: 295 bytes on wire (2360 bits), 295 bytes captured (2360 bits)
Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15
Transmission Control Protocol, Src Port: 59522 (59522), Dst Port: http-alt (8080), Seq: 460, A
Hypertext Transfer Protocol
▽ GET /controller/nb/v2/subnetservice/default/subnets HTTP/1.1\r\n
  ▽ [Expert Info (Chat/Sequence): GET /controller/nb/v2/subnetservice/default/subnets HTTP/1.
      [Message: GET /controller/nb/v2/subnetservice/default/subnets HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /controller/nb/v2/subnetservice/default/subnets
    Request Version: HTTP/1.1
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/subnetservice/default/subnets]
```

客户端向服务端发出子网信息请求。



```
 1952 14.808893   192.168.5.15          192.168.5.48         HTTP          59 HTTP/1.1 200
Hypertext Transfer Protocol
▽ HTTP/1.1 200 OK\r\n
  ▷ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Request Version: HTTP/1.1
    Status Code: 200
    Response Phrase: OK
  Server: Apache-Coyote/1.1\r\n
  Cache-Control: private\r\n
  Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
  Set-Cookie: JSESSIONIDSSO=9E170A369D463CED95397F463B114CB5; Path=/\r\n
  Set-Cookie: JSESSIONID=40F5BC0BF31B106534033104D8AF58C6; Path=/\r\n
  Content-Type: application/json\r\n
  Transfer-Encoding: chunked\r\n
  Date: Mon, 10 Mar 2014 06:05:00 GMT\r\n
  \r\n
▷ HTTP chunked response
JavaScript Object Notation: application/json
▽ Object
  ▽ Member Key: "subnetConfig"
    ▽ Array
      ▽ Object
        ▽ Member Key: "name"
            String value: eng1
        ▽ Member Key: "subnet"
            String value: 11.1.1.254/16
        ▽ Member Key: "nodeConnectors"
          ▽ Array
              String value: OF|1@OF|00:00:00:00:00:00:00:01
              String value: OF|2@OF|00:00:00:00:00:00:00:02
              String value: OF|3@OF|00:00:00:00:00:00:00:03
      ▽ Object
        ▽ Member Key: "name"
            String value: jiye2
        ▽ Member Key: "subnet"
            String value: 192.169.5.0/18
        ▽ Member Key: "nodeConnectors"
            Array
      ▽ Object
        ▽ Member Key: "name"
            String value: jiye
        ▽ Member Key: "subnet"
            String value: 192.168.5.128/19
        ▽ Member Key: "nodeConnectors"
          ▽ Array
              String value: OF|3@OF|00:00:00:00:00:00:00:02
```

服务端返回子网信息，总共返回三个子网信息。

A.子网 1

子网名：eng1

地址：11.1.1.254/16

包含内容：交换机 1：端口 1，交换机 2：端口 2，交换机 3：端口 3。

B.子网 2

子网名：jiye2

地址：192.168.5.0/18

包含内容：无

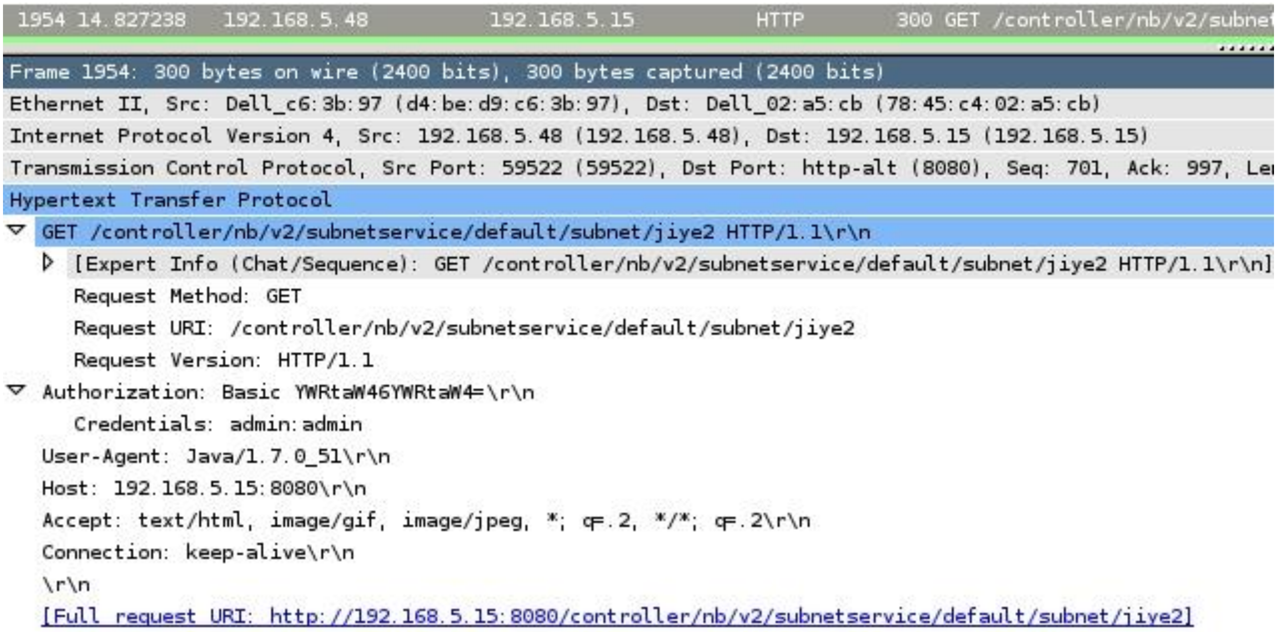
C.子网 3

子网名：jiye

地址：192.168.5.128/19

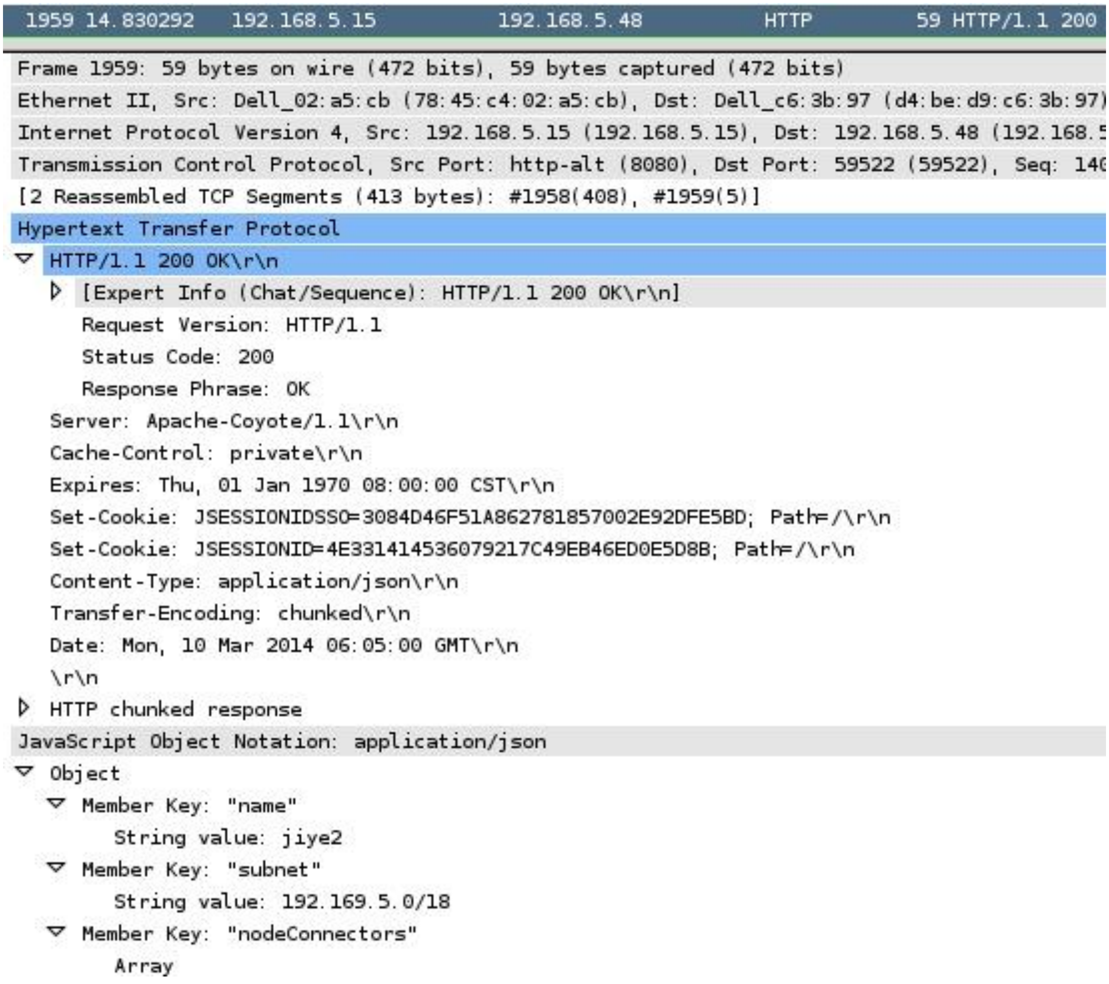
包含内容：交换机 2：端口 3

3）获取指定子网信息



```
1954 14.827238   192.168.5.48          192.168.5.15          HTTP        300 GET /controller/nb/v2/subne
Frame 1954: 300 bytes on wire (2400 bits), 300 bytes captured (2400 bits)
Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
Transmission Control Protocol, Src Port: 59522 (59522), Dst Port: http-alt (8080), Seq: 701, Ack: 997, Le
Hypertext Transfer Protocol
  GET /controller/nb/v2/subnetservice/default/subnet/jiye2 HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /controller/nb/v2/subnetservice/default/subnet/jiye2 HTTP/1.1\r\n]
    Request Method: GET
    Request URI: /controller/nb/v2/subnetservice/default/subnet/jiye2
    Request Version: HTTP/1.1
  Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/subnetservice/default/subnet/jiye2]
```

客户端发出获取指定子网信息的请求。



```
 1959 14.830292   192.168.5.15          192.168.5.48          HTTP         59 HTTP/1.1 200

Frame 1959: 59 bytes on wire (472 bits), 59 bytes captured (472 bits)
Ethernet II, Src: Dell_02:a5:cb (78:45:c4:02:a5:cb), Dst: Dell_c6:3b:97 (d4:be:d9:c6:3b:97)
Internet Protocol Version 4, Src: 192.168.5.15 (192.168.5.15), Dst: 192.168.5.48 (192.168.5
Transmission Control Protocol, Src Port: http-alt (8080), Dst Port: 59522 (59522), Seq: 14(
[2 Reassembled TCP Segments (413 bytes): #1958(408), #1959(5)]
Hypertext Transfer Protocol
▽ HTTP/1.1 200 OK\r\n
  ▷ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Request Version: HTTP/1.1
    Status Code: 200
    Response Phrase: OK
  Server: Apache-Coyote/1.1\r\n
  Cache-Control: private\r\n
  Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
  Set-Cookie: JSESSIONIDSSO=3084D46F51A862781857002E92DFE5BD; Path=/\r\n
  Set-Cookie: JSESSIONID=4E331414536079217C49EB46ED0E5D8B; Path=/\r\n
  Content-Type: application/json\r\n
  Transfer-Encoding: chunked\r\n
  Date: Mon, 10 Mar 2014 06:05:00 GMT\r\n
  \r\n
▷ HTTP chunked response
JavaScript Object Notation: application/json
▽ Object
  ▽ Member Key: "name"
      String value: jiye2
  ▽ Member Key: "subnet"
      String value: 192.169.5.0/18
  ▽ Member Key: "nodeConnectors"
      Array
```

获取成功，返回 200。

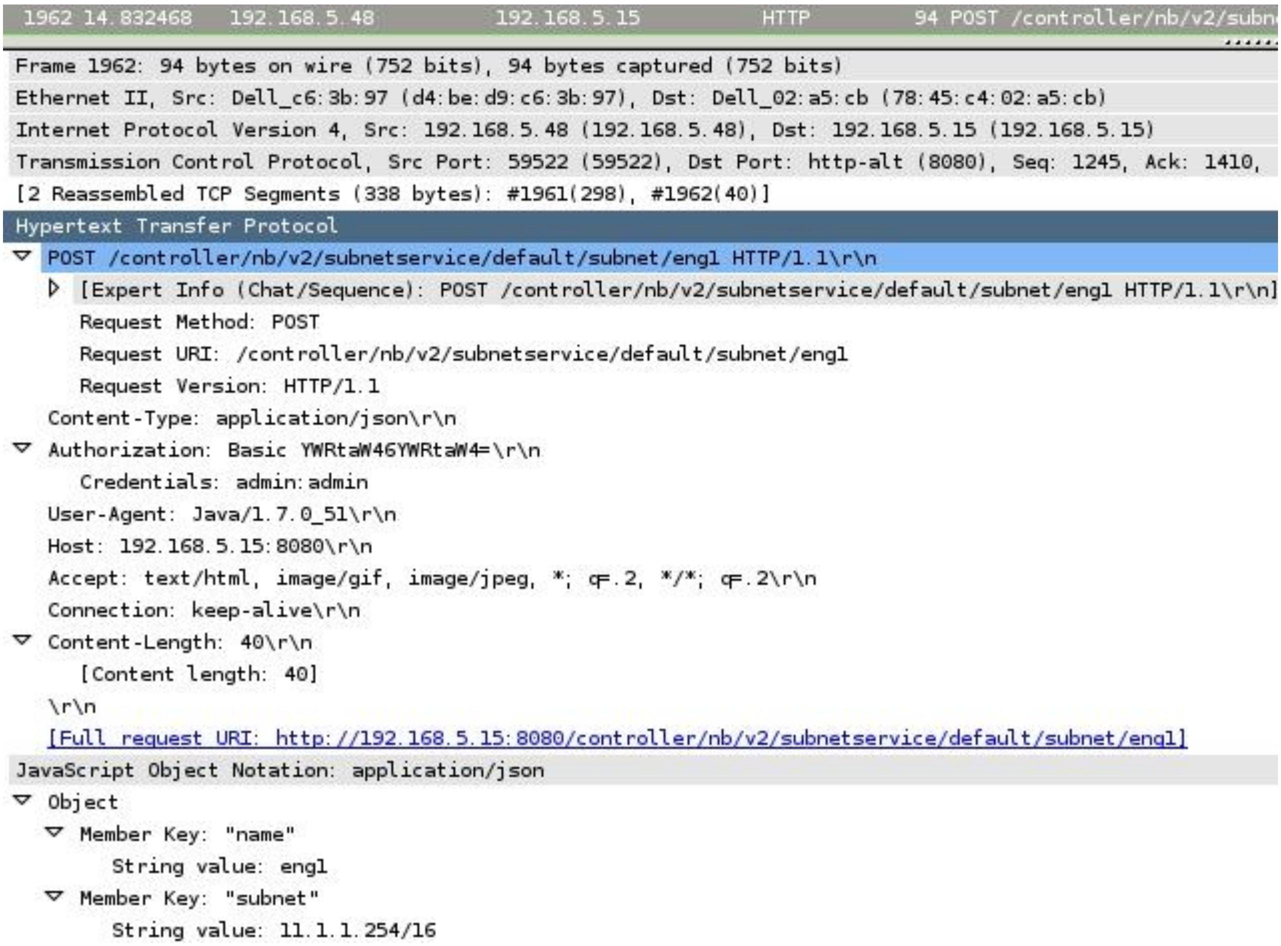
子网信息：

子网名：jiye2

地址：192.168.5.0/18

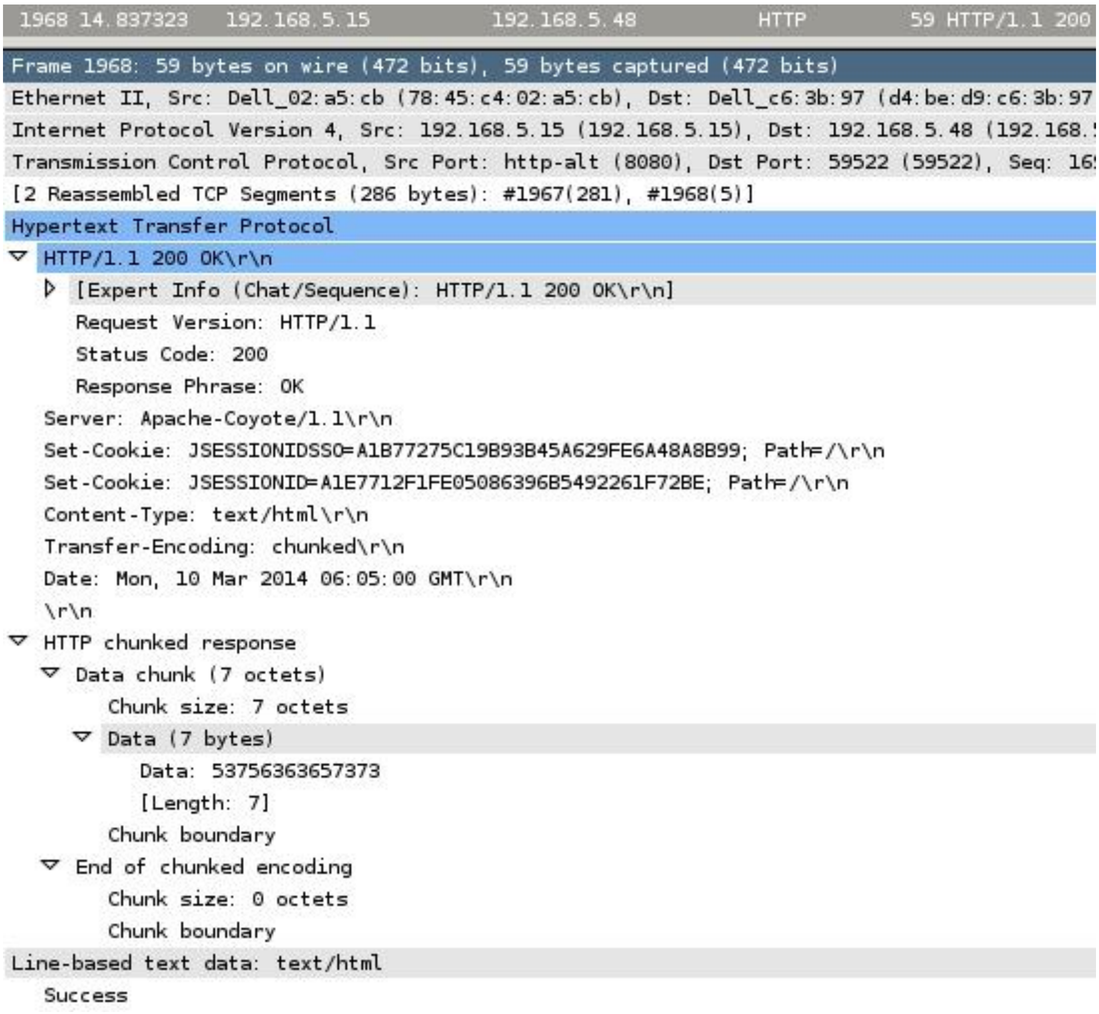
包含内容：无

4）修改指定子网信息



```
 1962 14.832468   192.168.5.48         192.168.5.15        HTTP       94 POST /controller/nb/v2/subn
Frame 1962: 94 bytes on wire (752 bits), 94 bytes captured (752 bits)
Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
Transmission Control Protocol, Src Port: 59522 (59522), Dst Port: http-alt (8080), Seq: 1245, Ack: 1410,
[2 Reassembled TCP Segments (338 bytes): #1961(298), #1962(40)]
Hypertext Transfer Protocol
▽ POST /controller/nb/v2/subnetservice/default/subnet/engl HTTP/1.1\r\n
  ▷ [Expert Info (Chat/Sequence): POST /controller/nb/v2/subnetservice/default/subnet/engl HTTP/1.1\r\n]
    Request Method: POST
    Request URI: /controller/nb/v2/subnetservice/default/subnet/engl
    Request Version: HTTP/1.1
  Content-Type: application/json\r\n
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
▽ Content-Length: 40\r\n
    [Content length: 40]
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/subnetservice/default/subnet/engl]
JavaScript Object Notation: application/json
▽ Object
  ▽ Member Key: "name"
      String value: engl
  ▽ Member Key: "subnet"
      String value: 11.1.1.254/16
```

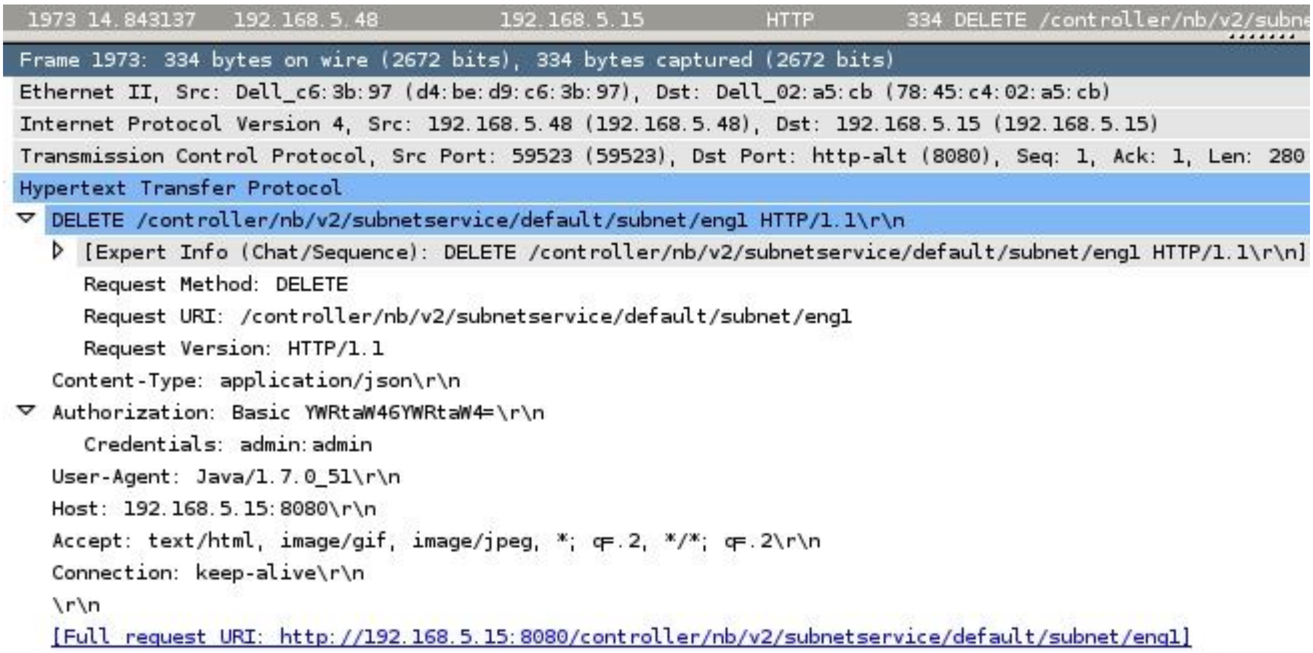
发出修改请求，求改子网的地址为 11.1.1.254/16。



```
 1968 14.837323   192.168.5.15          192.168.5.48        HTTP        59 HTTP/1.1 200
Frame 1968: 59 bytes on wire (472 bits), 59 bytes captured (472 bits)
Ethernet II, Src: Dell_02:a5:cb (78:45:c4:02:a5:cb), Dst: Dell_c6:3b:97 (d4:be:d9:c6:3b:97
Internet Protocol Version 4, Src: 192.168.5.15 (192.168.5.15), Dst: 192.168.5.48 (192.168.
Transmission Control Protocol, Src Port: http-alt (8080), Dst Port: 59522 (59522), Seq: 16
[2 Reassembled TCP Segments (286 bytes): #1967(281), #1968(5)]
Hypertext Transfer Protocol
▽ HTTP/1.1 200 OK\r\n
  ▷ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
     Request Version: HTTP/1.1
     Status Code: 200
     Response Phrase: OK
  Server: Apache-Coyote/1.1\r\n
  Set-Cookie: JSESSIONIDSSO=A1B77275C19B93B45A629FE6A48A8B99; Path=/\r\n
  Set-Cookie: JSESSIONID=A1E7712F1FE05086396B5492261F72BE; Path=/\r\n
  Content-Type: text/html\r\n
  Transfer-Encoding: chunked\r\n
  Date: Mon, 10 Mar 2014 06:05:00 GMT\r\n
  \r\n
▽ HTTP chunked response
  ▽ Data chunk (7 octets)
       Chunk size: 7 octets
     ▽ Data (7 bytes)
          Data: 53756363657373
          [Length: 7]
       Chunk boundary
  ▽ End of chunked encoding
       Chunk size: 0 octets
       Chunk boundary
Line-based text data: text/html
  Success
```

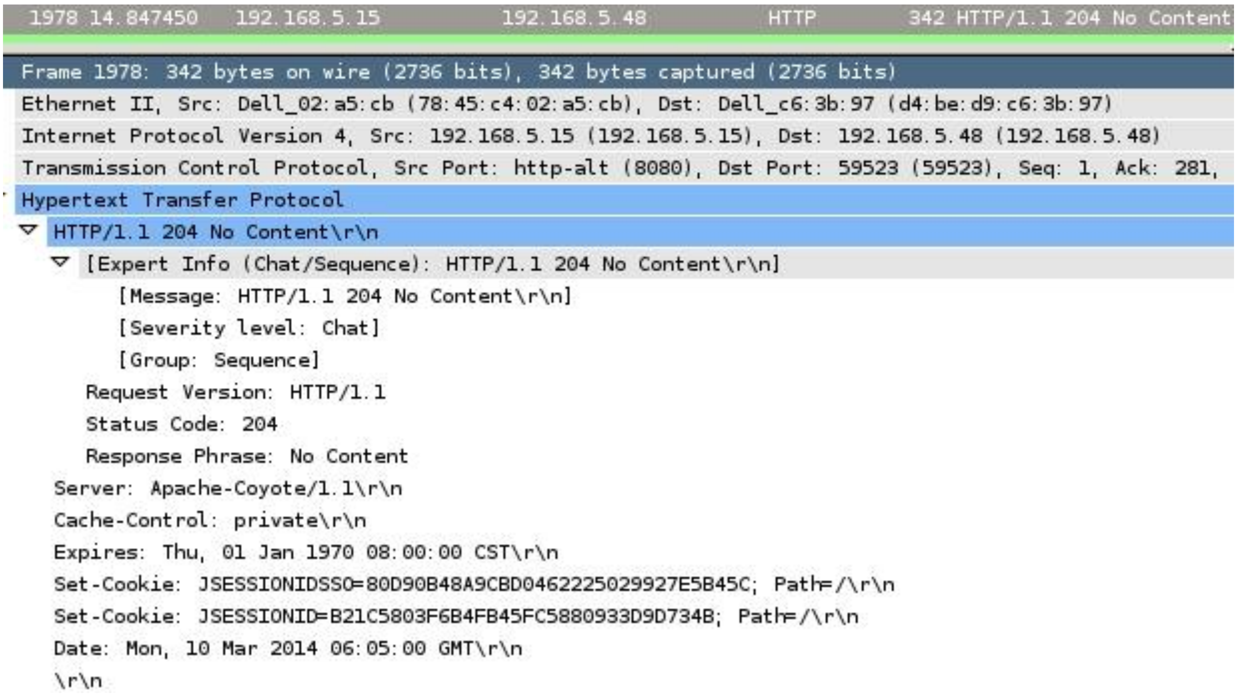
请求成功，返回 200。

5）删除指定子网



```
 1973 14.843137    192.168.5.48            192.168.5.15          HTTP        334 DELETE /controller/nb/v2/subne
Frame 1973: 334 bytes on wire (2672 bits), 334 bytes captured (2672 bits)
Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
Transmission Control Protocol, Src Port: 59523 (59523), Dst Port: http-alt (8080), Seq: 1, Ack: 1, Len: 280
Hypertext Transfer Protocol
▽ DELETE /controller/nb/v2/subnetservice/default/subnet/eng1 HTTP/1.1\r\n
  ▷ [Expert Info (Chat/Sequence): DELETE /controller/nb/v2/subnetservice/default/subnet/eng1 HTTP/1.1\r\n]
    Request Method: DELETE
    Request URI: /controller/nb/v2/subnetservice/default/subnet/eng1
    Request Version: HTTP/1.1
  Content-Type: application/json\r\n
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/subnetservice/default/subnet/eng1]
```

发出删除子网 eng1 的请求。



```
 1978 14.847450    192.168.5.15            192.168.5.48          HTTP        342 HTTP/1.1 204 No Content
Frame 1978: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
Ethernet II, Src: Dell_02:a5:cb (78:45:c4:02:a5:cb), Dst: Dell_c6:3b:97 (d4:be:d9:c6:3b:97)
Internet Protocol Version 4, Src: 192.168.5.15 (192.168.5.15), Dst: 192.168.5.48 (192.168.5.48)
Transmission Control Protocol, Src Port: http-alt (8080), Dst Port: 59523 (59523), Seq: 1, Ack: 281,
Hypertext Transfer Protocol
▽ HTTP/1.1 204 No Content\r\n
  ▽ [Expert Info (Chat/Sequence): HTTP/1.1 204 No Content\r\n]
      [Message: HTTP/1.1 204 No Content\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Version: HTTP/1.1
    Status Code: 204
    Response Phrase: No Content
  Server: Apache-Coyote/1.1\r\n
  Cache-Control: private\r\n
  Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
  Set-Cookie: JSESSIONIDSSO=80D90B48A9CBD0462225029927E5B45C; Path=/\r\n
  Set-Cookie: JSESSIONID=B21C5803F6B4FB45FC5880933D9D734B; Path=/\r\n
  Date: Mon, 10 Mar 2014 06:05:00 GMT\r\n
  \r\n
```

删除成功，返回 204.

7 Switch Manager REST APIs

本部分 APIs 的应用见 SwitchNorthbound

本部分 APIs 用来批准 node、node connector、property 的接入。

7.1 Switch Manager REST APIs 可提供的服务：

请求分类	客户端请求	服务器响应
nodes	GET:在网络上检索所有的节点和他们的属性	list
save	POST：保存最近的交换机设置	(custom)
node	GET:在网络上检索指定节点的节点连接器和他们的属性	list
node property	DELETE：删除指定节点的属性	(custom)
Node property value	PUT：增加一个转发层模式属性的描述给节点	(custom)
nodeConnector property	DELETE:删除节点连接器的属性	(custom)

7.2 Switch Manager REST APIs 数据模型

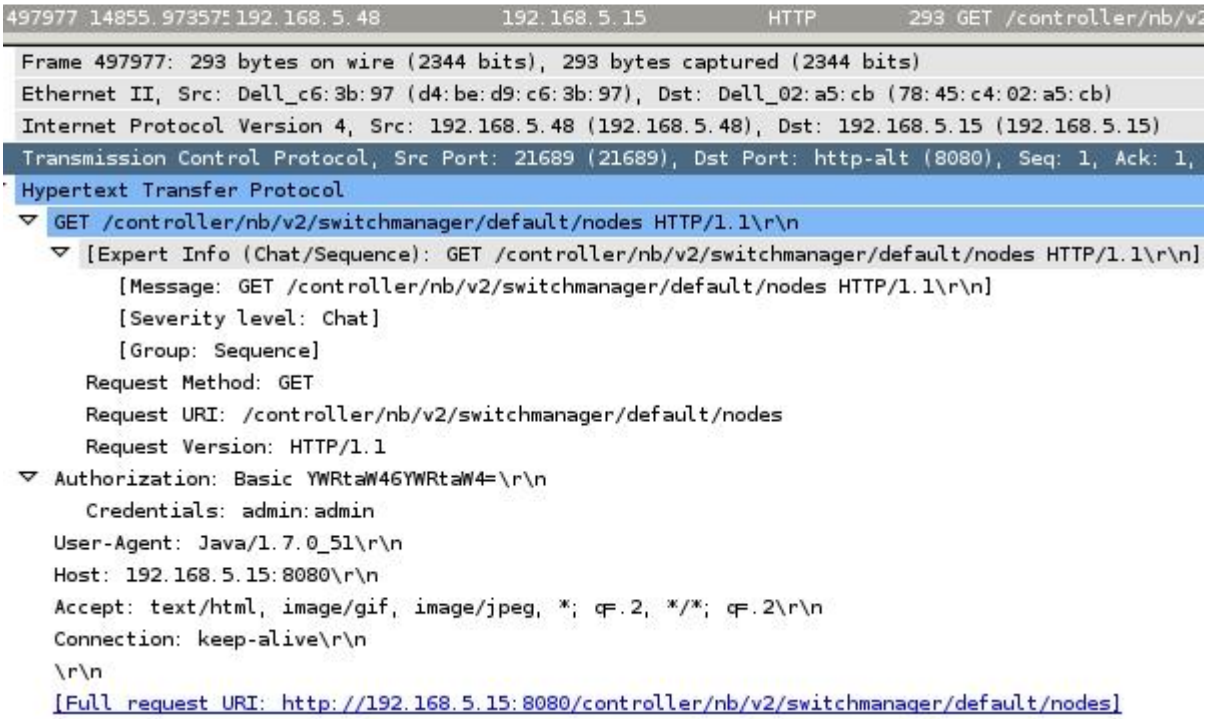
Data Elements	Data Types	XML Elements	
		名称（类型）	最大/最小 出现
list	nodes	nodeProperties (nodeProperties)	0/unbounded
node	node	type (string)	0/1
		id (string)	0/1
nodeConnector	nodeConnector	type (string)	0/1
		id (string)	0/1
		node (node)	0/1
nodeConnectorProperties	nodeConnectorProperties	nodeconnector (nodeConnector)	0/1
		properties/property	0/unbounded
nodeProperties	nodeProperties	node (node)	0/1
		properties/property	0/unbounded

7.3 实验分析

7.3.1 实验拓扑

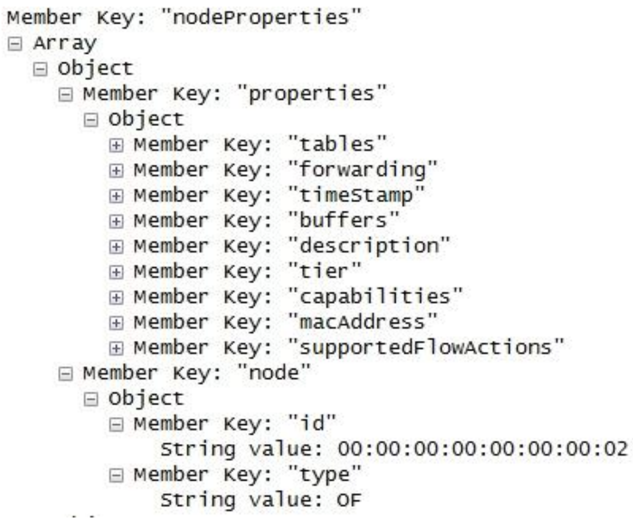
7.3.2 抓包情况分析

1）获取全部交换机信息

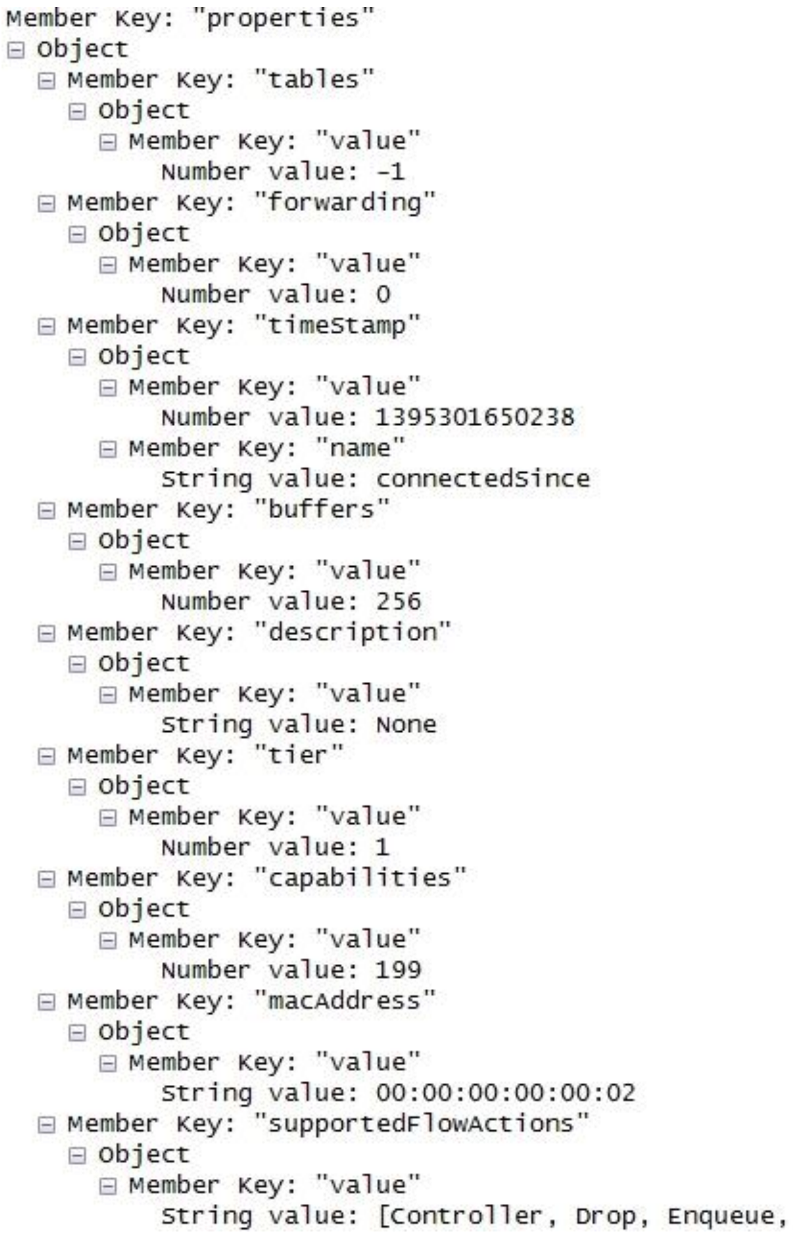


```
497977 14855.97357 192.168.5.48          192.168.5.15          HTTP          293 GET /controller/nb/v2
 Frame 497977: 293 bytes on wire (2344 bits), 293 bytes captured (2344 bits)
 Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
 Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
 Transmission Control Protocol, Src Port: 21689 (21689), Dst Port: http-alt (8080), Seq: 1, Ack: 1,
 Hypertext Transfer Protocol
  ▽ GET /controller/nb/v2/switchmanager/default/nodes HTTP/1.1\r\n
    ▽ [Expert Info (Chat/Sequence): GET /controller/nb/v2/switchmanager/default/nodes HTTP/1.1\r\n]
        [Message: GET /controller/nb/v2/switchmanager/default/nodes HTTP/1.1\r\n]
        [Severity level: Chat]
        [Group: Sequence]
      Request Method: GET
      Request URI: /controller/nb/v2/switchmanager/default/nodes
      Request Version: HTTP/1.1
  ▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
      Credentials: admin:admin
    User-Agent: Java/1.7.0_51\r\n
    Host: 192.168.5.15:8080\r\n
    Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://192.168.5.15:8080/controller/nb/v2/switchmanager/default/nodes]
```

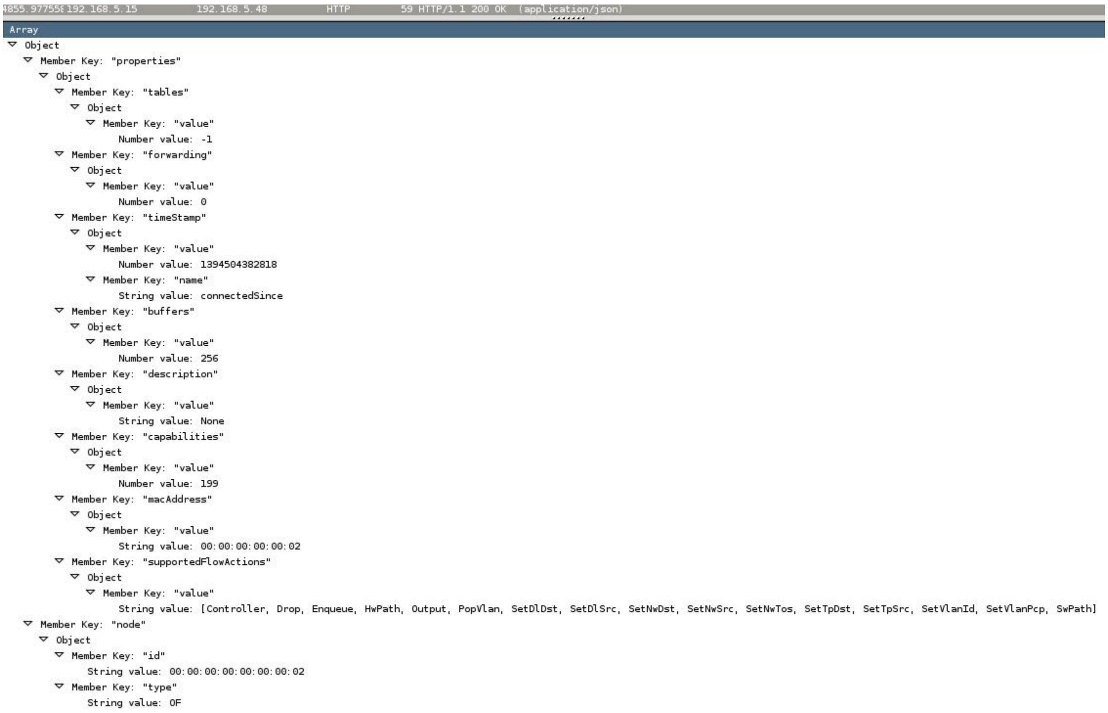
客户端请求交换机信息



```
Member Key: "nodeProperties"
⊟ Array
  ⊟ Object
    ⊟ Member Key: "properties"
      ⊟ Object
        ⊞ Member Key: "tables"
        ⊞ Member Key: "forwarding"
        ⊞ Member Key: "timeStamp"
        ⊞ Member Key: "buffers"
        ⊞ Member Key: "description"
        ⊞ Member Key: "tier"
        ⊞ Member Key: "capabilities"
        ⊞ Member Key: "macAddress"
        ⊞ Member Key: "supportedFlowActions"
    ⊟ Member Key: "node"
      ⊟ Object
        ⊟ Member Key: "id"
            String value: 00:00:00:00:00:00:00:02
        ⊟ Member Key: "type"
            String value: OF
```



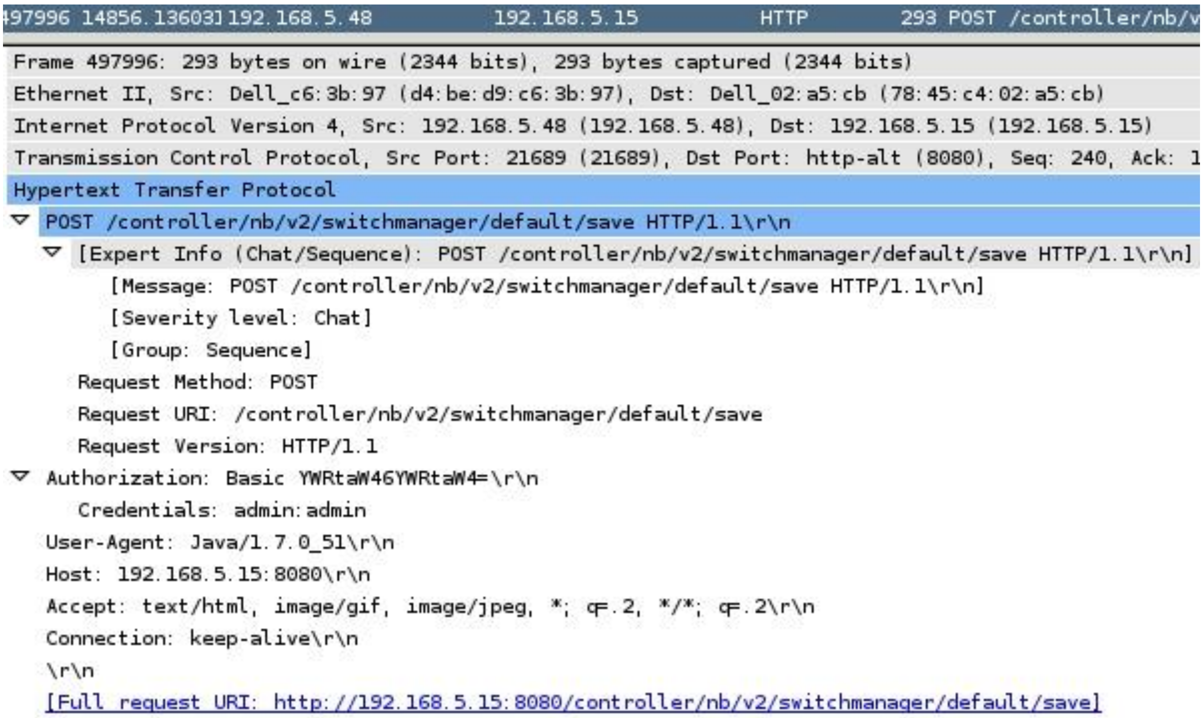
```
Member Key: "properties"
⊟ Object
  ⊟ Member Key: "tables"
    ⊟ Object
      ⊟ Member Key: "value"
          Number value: -1
  ⊟ Member Key: "forwarding"
    ⊟ Object
      ⊟ Member Key: "value"
          Number value: 0
  ⊟ Member Key: "timeStamp"
    ⊟ Object
      ⊟ Member Key: "value"
          Number value: 1395301650238
      ⊟ Member Key: "name"
          String value: connectedSince
  ⊟ Member Key: "buffers"
    ⊟ Object
      ⊟ Member Key: "value"
          Number value: 256
  ⊟ Member Key: "description"
    ⊟ Object
      ⊟ Member Key: "value"
          String value: None
  ⊟ Member Key: "tier"
    ⊟ Object
      ⊟ Member Key: "value"
          Number value: 1
  ⊟ Member Key: "capabilities"
    ⊟ Object
      ⊟ Member Key: "value"
          Number value: 199
  ⊟ Member Key: "macAddress"
    ⊟ Object
      ⊟ Member Key: "value"
          String value: 00:00:00:00:00:02
  ⊟ Member Key: "supportedFlowActions"
    ⊟ Object
      ⊟ Member Key: "value"
          String value: [Controller, Drop, Enqueue,
```



```
4855.97755E 192.168.5.15        192.168.5.48        HTTP        59 HTTP/1.1 200 OK  (application/json)
Array
▽ Object
  ▽ Member Key: "properties"
    ▽ Object
      ▽ Member Key: "tables"
        ▽ Object
          ▽ Member Key: "value"
              Number value: -1
      ▽ Member Key: "forwarding"
        ▽ Object
          ▽ Member Key: "value"
              Number value: 0
      ▽ Member Key: "timeStamp"
        ▽ Object
          ▽ Member Key: "value"
              Number value: 1394504382818
          ▽ Member Key: "name"
              String value: connectedSince
      ▽ Member Key: "buffers"
        ▽ Object
          ▽ Member Key: "value"
              Number value: 256
      ▽ Member Key: "description"
        ▽ Object
          ▽ Member Key: "value"
              String value: None
      ▽ Member Key: "capabilities"
        ▽ Object
          ▽ Member Key: "value"
              Number value: 199
      ▽ Member Key: "macAddress"
        ▽ Object
          ▽ Member Key: "value"
              String value: 00:00:00:00:00:02
      ▽ Member Key: "supportedFlowActions"
        ▽ Object
          ▽ Member Key: "value"
              String value: [Controller, Drop, Enqueue, HwPath, Output, PopVlan, SetDlDst, SetDlSrc, SetNwDst, SetNwSrc, SetNwTos, SetTpDst, SetTpSrc, SetVlanId, SetVlanPcp, SwPath]
  ▽ Member Key: "node"
    ▽ Object
      ▽ Member Key: "id"
          String value: 00:00:00:00:00:00:00:02
      ▽ Member Key: "type"
          String value: OF
```

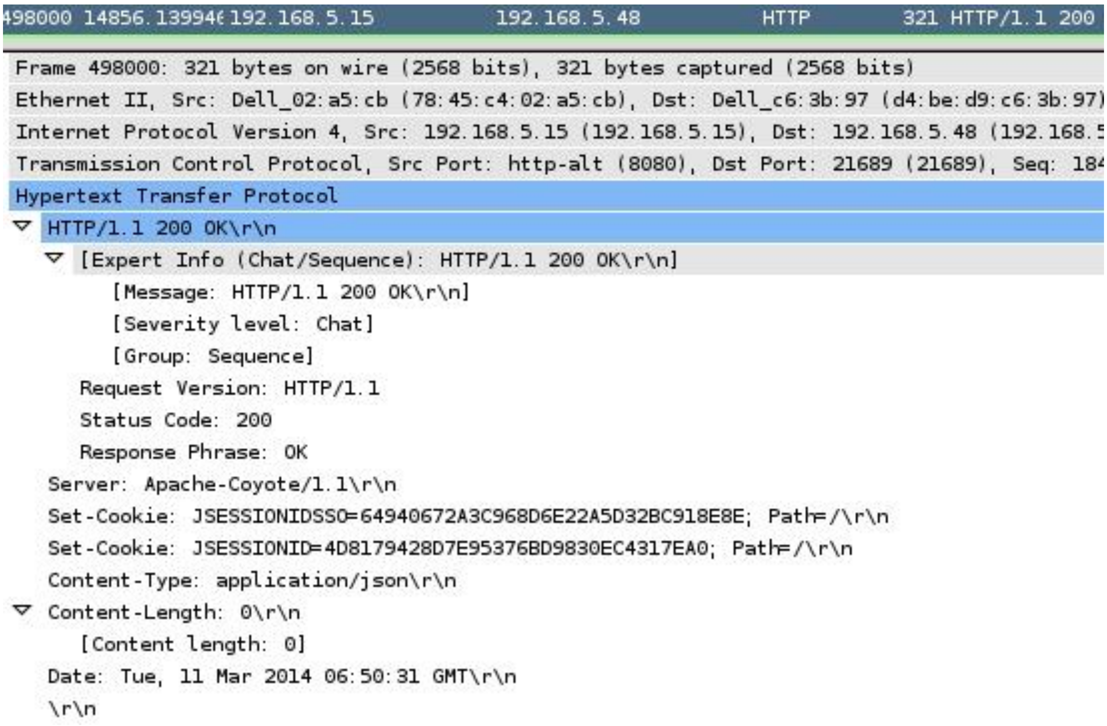
服务端响应。

2）存储交换机配置信息



```
497996 14856.13603] 192.168.5.48          192.168.5.15          HTTP          293 POST /controller/nb/v
Frame 497996: 293 bytes on wire (2344 bits), 293 bytes captured (2344 bits)
Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
Transmission Control Protocol, Src Port: 21689 (21689), Dst Port: http-alt (8080), Seq: 240, Ack: 1
Hypertext Transfer Protocol
▽ POST /controller/nb/v2/switchmanager/default/save HTTP/1.1\r\n
  ▽ [Expert Info (Chat/Sequence): POST /controller/nb/v2/switchmanager/default/save HTTP/1.1\r\n]
      [Message: POST /controller/nb/v2/switchmanager/default/save HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: POST
    Request URI: /controller/nb/v2/switchmanager/default/save
    Request Version: HTTP/1.1
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/switchmanager/default/save]
```

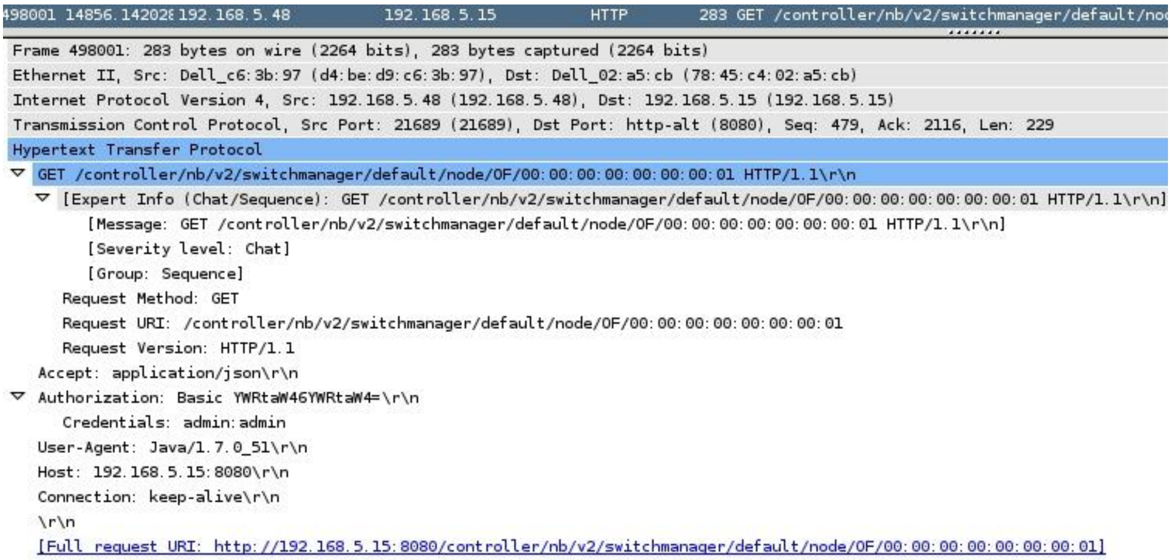
客户端发起 post 请求。



```
498000 14856.13994( 192.168.5.15          192.168.5.48          HTTP          321 HTTP/1.1 200
Frame 498000: 321 bytes on wire (2568 bits), 321 bytes captured (2568 bits)
Ethernet II, Src: Dell_02:a5:cb (78:45:c4:02:a5:cb), Dst: Dell_c6:3b:97 (d4:be:d9:c6:3b:97)
Internet Protocol Version 4, Src: 192.168.5.15 (192.168.5.15), Dst: 192.168.5.48 (192.168.5
Transmission Control Protocol, Src Port: http-alt (8080), Dst Port: 21689 (21689), Seq: 184
Hypertext Transfer Protocol
▽ HTTP/1.1 200 OK\r\n
  ▽ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [Message: HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Version: HTTP/1.1
    Status Code: 200
    Response Phrase: OK
  Server: Apache-Coyote/1.1\r\n
  Set-Cookie: JSESSIONIDSSO=64940672A3C968D6E22A5D32BC918E8E; Path=/\r\n
  Set-Cookie: JSESSIONID=4D8179428D7E95376BD9830EC4317EA0; Path=/\r\n
  Content-Type: application/json\r\n
▽ Content-Length: 0\r\n
    [Content length: 0]
  Date: Tue, 11 Mar 2014 06:50:31 GMT\r\n
  \r\n
```

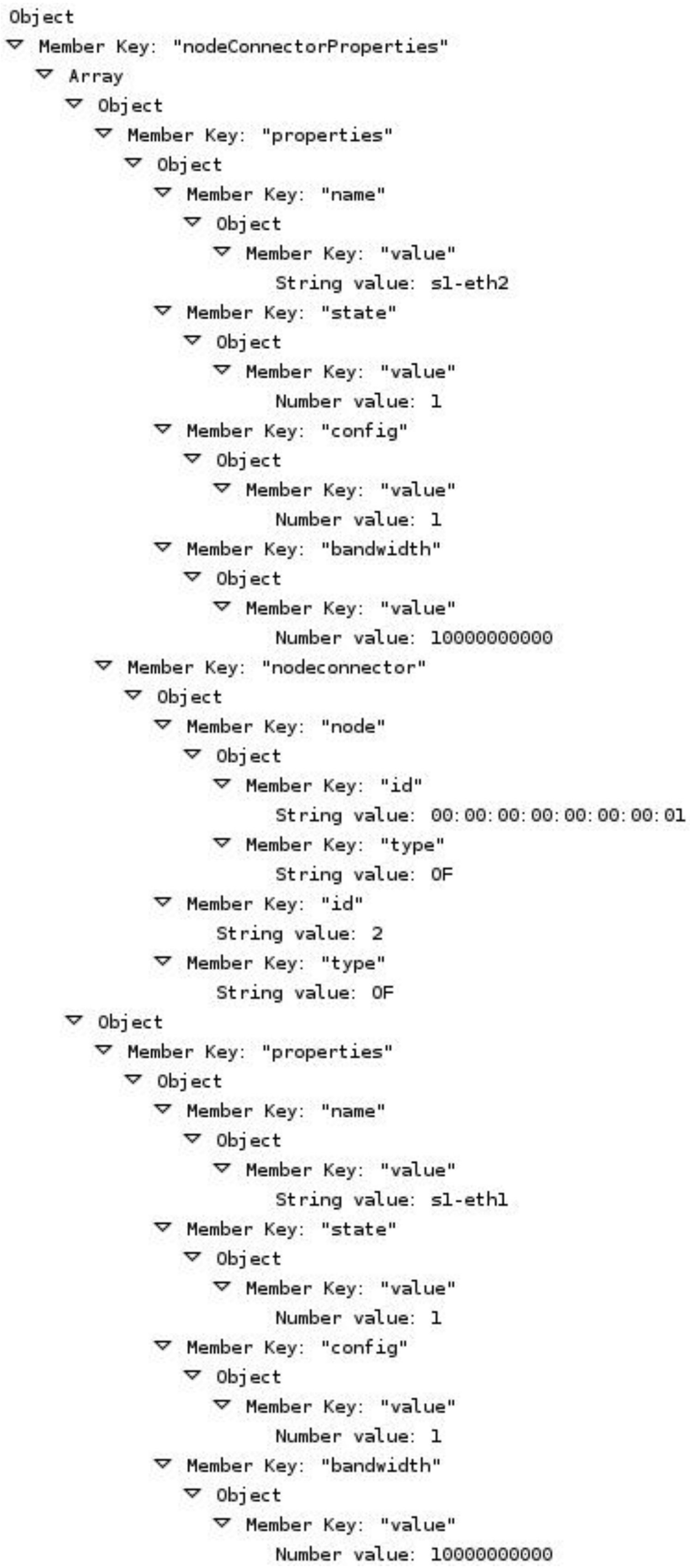
保存成功，返回 200。

3）获取单个交换机信息

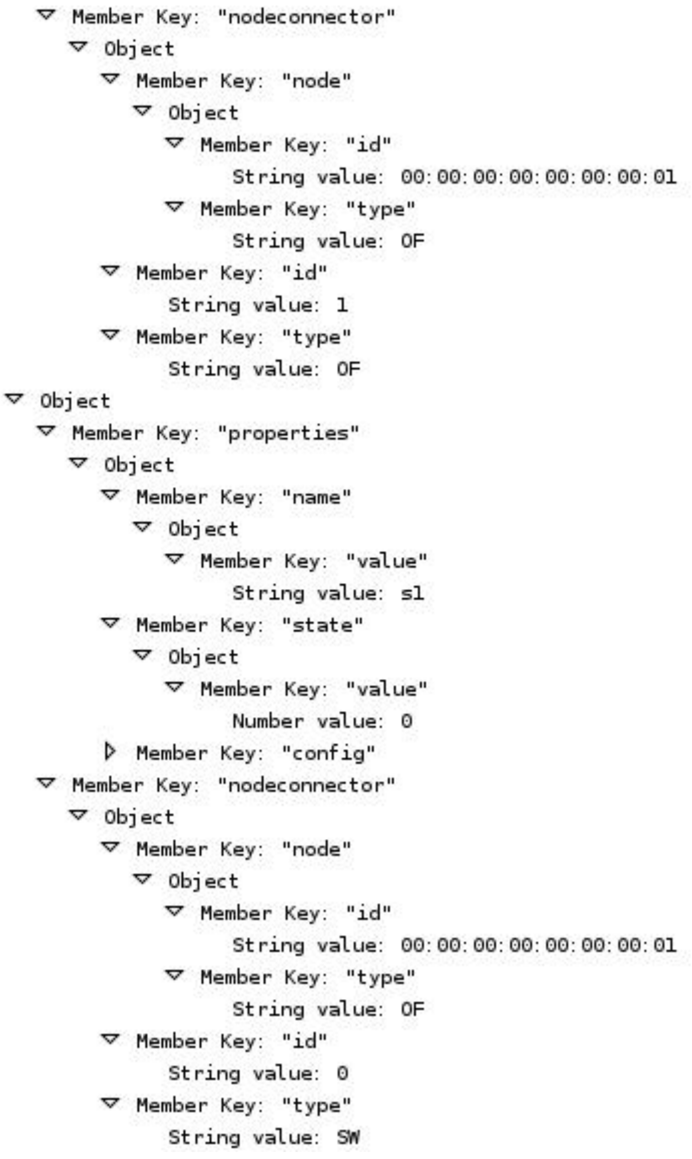


```
498001 14856.142028 192.168.5.48          192.168.5.15          HTTP        283 GET /controller/nb/v2/switchmanager/default/noc
 Frame 498001: 283 bytes on wire (2264 bits), 283 bytes captured (2264 bits)
 Ethernet II, Src: Dell_c6:3b:97 (d4:be:d9:c6:3b:97), Dst: Dell_02:a5:cb (78:45:c4:02:a5:cb)
 Internet Protocol Version 4, Src: 192.168.5.48 (192.168.5.48), Dst: 192.168.5.15 (192.168.5.15)
 Transmission Control Protocol, Src Port: 21689 (21689), Dst Port: http-alt (8080), Seq: 479, Ack: 2116, Len: 229
 Hypertext Transfer Protocol
▽ GET /controller/nb/v2/switchmanager/default/node/OF/00:00:00:00:00:00:00:01 HTTP/1.1\r\n
  ▽ [Expert Info (Chat/Sequence): GET /controller/nb/v2/switchmanager/default/node/OF/00:00:00:00:00:00:00:01 HTTP/1.1\r\n]
      [Message: GET /controller/nb/v2/switchmanager/default/node/OF/00:00:00:00:00:00:00:01 HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: GET
    Request URI: /controller/nb/v2/switchmanager/default/node/OF/00:00:00:00:00:00:00:01
    Request Version: HTTP/1.1
  Accept: application/json\r\n
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/switchmanager/default/node/OF/00:00:00:00:00:00:00:01]
```

客户端发出请求。



```
Object
▽ Member Key: "nodeConnectorProperties"
  ▽ Array
    ▽ Object
      ▽ Member Key: "properties"
        ▽ Object
          ▽ Member Key: "name"
            ▽ Object
              ▽ Member Key: "value"
                  String value: s1-eth2
          ▽ Member Key: "state"
            ▽ Object
              ▽ Member Key: "value"
                  Number value: 1
          ▽ Member Key: "config"
            ▽ Object
              ▽ Member Key: "value"
                  Number value: 1
          ▽ Member Key: "bandwidth"
            ▽ Object
              ▽ Member Key: "value"
                  Number value: 10000000000
      ▽ Member Key: "nodeconnector"
        ▽ Object
          ▽ Member Key: "node"
            ▽ Object
              ▽ Member Key: "id"
                  String value: 00:00:00:00:00:00:00:01
              ▽ Member Key: "type"
                  String value: OF
          ▽ Member Key: "id"
              String value: 2
          ▽ Member Key: "type"
              String value: OF
    ▽ Object
      ▽ Member Key: "properties"
        ▽ Object
          ▽ Member Key: "name"
            ▽ Object
              ▽ Member Key: "value"
                  String value: s1-eth1
          ▽ Member Key: "state"
            ▽ Object
              ▽ Member Key: "value"
                  Number value: 1
          ▽ Member Key: "config"
            ▽ Object
              ▽ Member Key: "value"
                  Number value: 1
          ▽ Member Key: "bandwidth"
            ▽ Object
              ▽ Member Key: "value"
                  Number value: 10000000000
```



```
   ▽ Member Key: "nodeconnector"
      ▽ Object
         ▽ Member Key: "node"
            ▽ Object
               ▽ Member Key: "id"
                    String value: 00:00:00:00:00:00:00:01
               ▽ Member Key: "type"
                    String value: OF
         ▽ Member Key: "id"
              String value: 1
         ▽ Member Key: "type"
              String value: OF
▽ Object
   ▽ Member Key: "properties"
      ▽ Object
         ▽ Member Key: "name"
            ▽ Object
               ▽ Member Key: "value"
                    String value: s1
         ▽ Member Key: "state"
            ▽ Object
               ▽ Member Key: "value"
                    Number value: 0
         ▷ Member Key: "config"
   ▽ Member Key: "nodeconnector"
      ▽ Object
         ▽ Member Key: "node"
            ▽ Object
               ▽ Member Key: "id"
                    String value: 00:00:00:00:00:00:00:01
               ▽ Member Key: "type"
                    String value: OF
         ▽ Member Key: "id"
              String value: 0
         ▽ Member Key: "type"
              String value: SW
```

8 User Manager REST APIs

本部分 APIs 的应用见 UserManagerNorthbound
本部分 APIs 用来管理用户，并且只能用于 HTTPs 协议。

8.1 User Manager REST APIs 可提供的服务：

请求分类	客户端请求	服务器响应
users	POST：增加一位新用户	userConfig
users	DELETE：删除一个用户	(custom)

8.2 User Manager REST APIs 数据模型

Data Elements	Data Types	XML Elements	
		名称（类型）	最大/最小 出现
userConfig	userConfig	roles (string)	0/unbounded
		password (string)	0/1
		user (string)	0/1
	configurationObject		

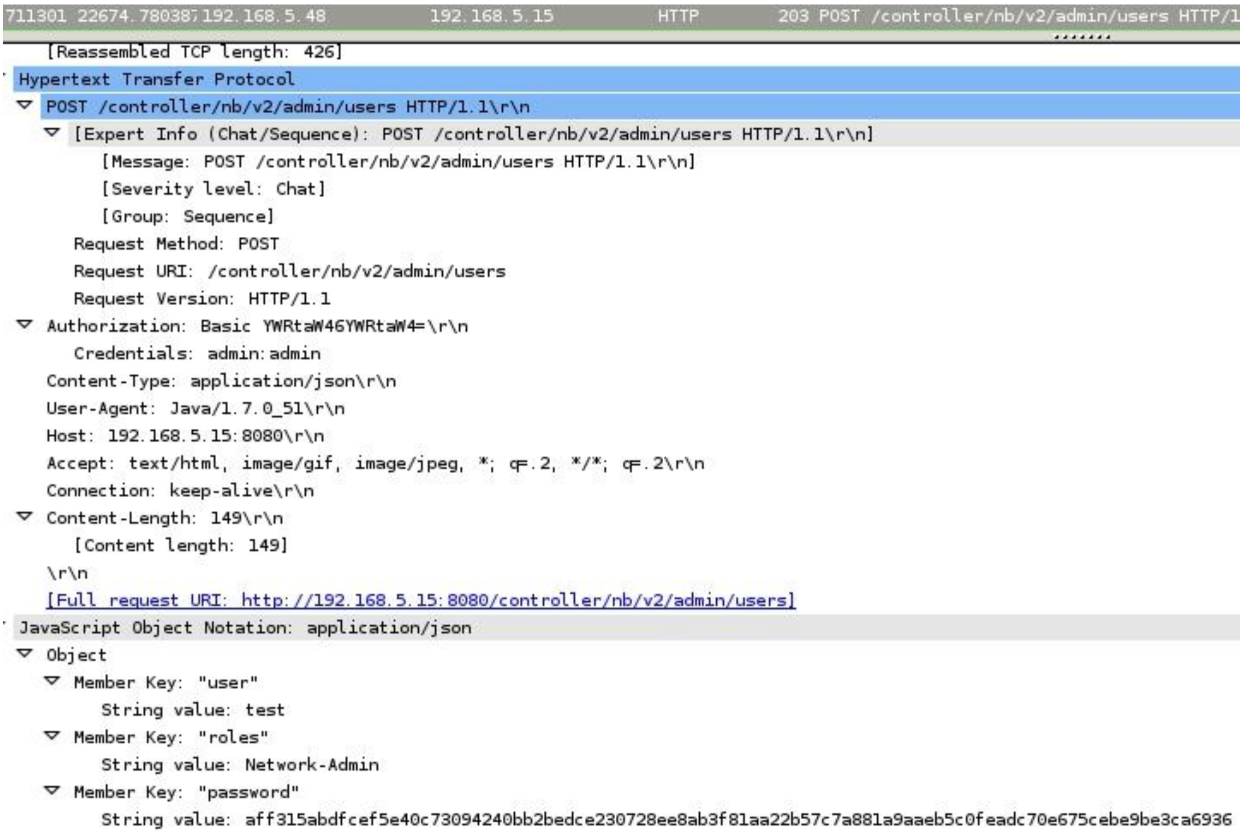
8.3 实验分析

8.3.1 抓包分析

1）添加新用户

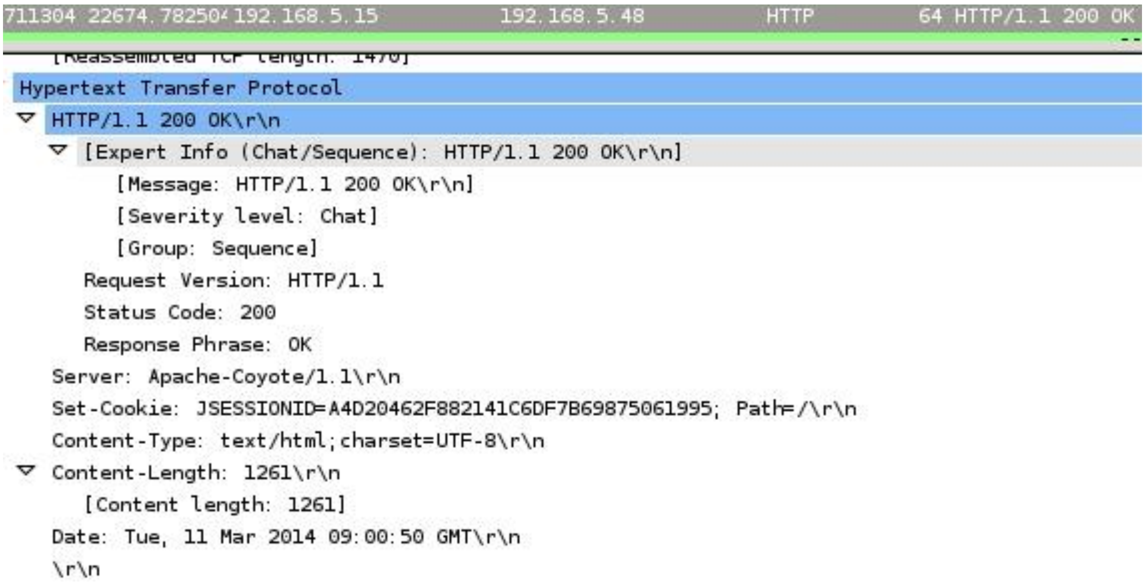
用户名：test、密码：testPass、角色（roles）：Network-Admin

这里要特别注意 url 的写法



```
711301 22674.780387 192.168.5.48          192.168.5.15          HTTP          203 POST /controller/nb/v2/admin/users HTTP/1
    [Reassembled TCP length: 426]
Hypertext Transfer Protocol
  POST /controller/nb/v2/admin/users HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): POST /controller/nb/v2/admin/users HTTP/1.1\r\n]
      [Message: POST /controller/nb/v2/admin/users HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: POST
    Request URI: /controller/nb/v2/admin/users
    Request Version: HTTP/1.1
  Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  Content-Type: application/json\r\n
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  Content-Length: 149\r\n
    [Content length: 149]
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/admin/users]
JavaScript Object Notation: application/json
  Object
    Member Key: "user"
      String value: test
    Member Key: "roles"
      String value: Network-Admin
    Member Key: "password"
      String value: aff315abdfcef5e40c73094240bb2bedce230728ee8ab3f81aa22b57c7a881a9aaeb5c0feadc70e675cebe9be3ca6936
```

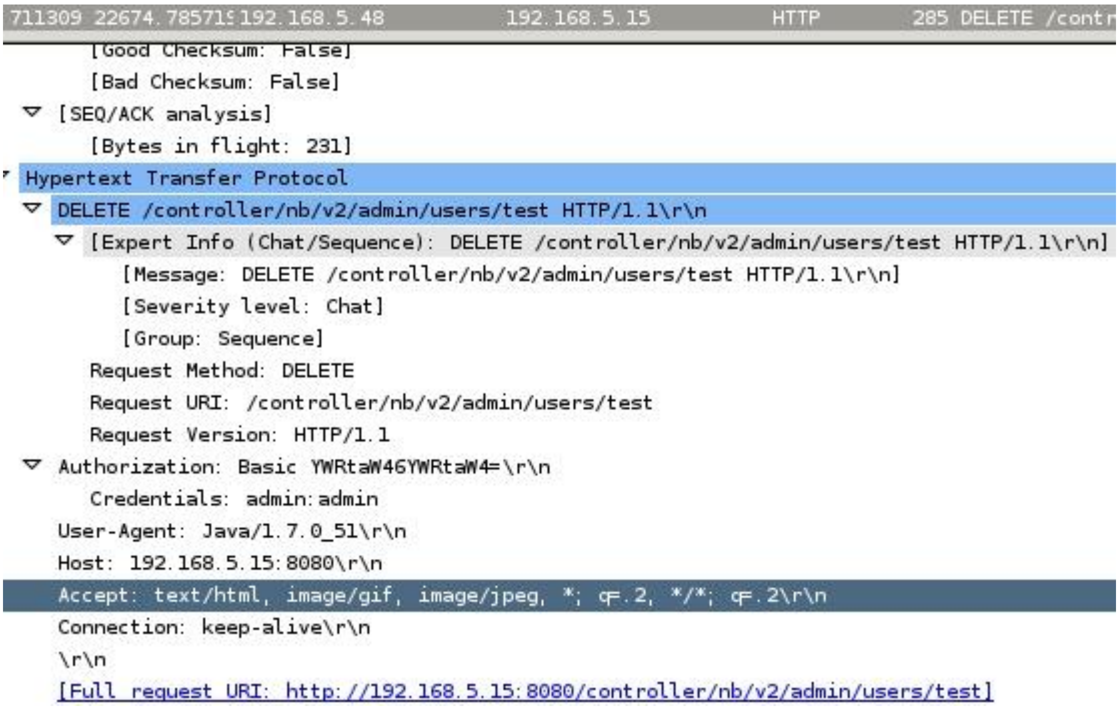
客户端向服务端发出 post 请求。（这里有一个疑问：POST 和 PUT 的区别究竟是什么）。



```
711304 22674.78250‹ 192.168.5.15        192.168.5.48        HTTP        64 HTTP/1.1 200 OK
    [Reassembled TCP Length: 1470]
Hypertext Transfer Protocol
▽ HTTP/1.1 200 OK\r\n
  ▽ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [Message: HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Version: HTTP/1.1
    Status Code: 200
    Response Phrase: OK
  Server: Apache-Coyote/1.1\r\n
  Set-Cookie: JSESSIONID=A4D20462F882141C6DF7B69875061995; Path=/\r\n
  Content-Type: text/html;charset=UTF-8\r\n
▽ Content-Length: 1261\r\n
    [Content length: 1261]
  Date: Tue, 11 Mar 2014 09:00:50 GMT\r\n
  \r\n
```

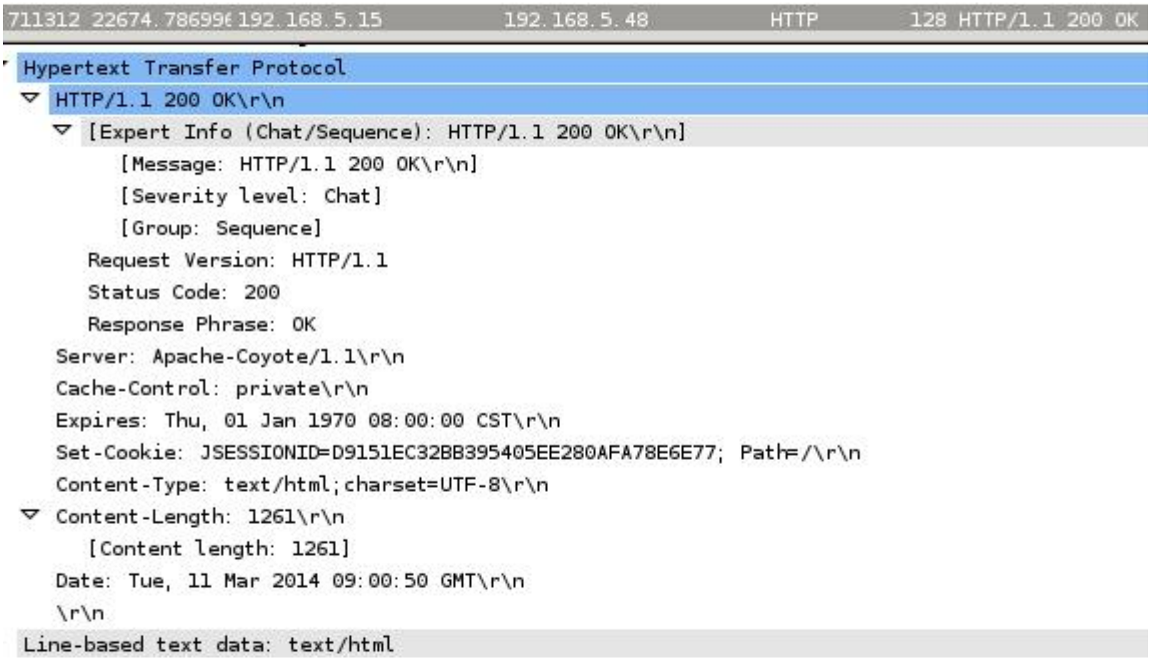
创建成功，返回 200。

2）删除用户



```
711309 22674.785715 192.168.5.48        192.168.5.15        HTTP        285 DELETE /contr
      [Good Checksum: False]
      [Bad Checksum: False]
  ▽ [SEQ/ACK analysis]
      [Bytes in flight: 231]
Hypertext Transfer Protocol
▽ DELETE /controller/nb/v2/admin/users/test HTTP/1.1\r\n
  ▽ [Expert Info (Chat/Sequence): DELETE /controller/nb/v2/admin/users/test HTTP/1.1\r\n]
      [Message: DELETE /controller/nb/v2/admin/users/test HTTP/1.1\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Method: DELETE
    Request URI: /controller/nb/v2/admin/users/test
    Request Version: HTTP/1.1
▽ Authorization: Basic YWRtaW46YWRtaW4=\r\n
    Credentials: admin:admin
  User-Agent: Java/1.7.0_51\r\n
  Host: 192.168.5.15:8080\r\n
  Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2\r\n
  Connection: keep-alive\r\n
  \r\n
  [Full request URI: http://192.168.5.15:8080/controller/nb/v2/admin/users/test]
```

客户端发出删除用户请求



```
711312 22674.786996 192.168.5.15          192.168.5.48          HTTP          128 HTTP/1.1 200 OK

Hypertext Transfer Protocol
▽ HTTP/1.1 200 OK\r\n
  ▽ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [Message: HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
    Request Version: HTTP/1.1
    Status Code: 200
    Response Phrase: OK
  Server: Apache-Coyote/1.1\r\n
  Cache-Control: private\r\n
  Expires: Thu, 01 Jan 1970 08:00:00 CST\r\n
  Set-Cookie: JSESSIONID=D9151EC32BB395405EE280AFA78E6E77; Path=/\r\n
  Content-Type: text/html;charset=UTF-8\r\n
▽ Content-Length: 1261\r\n
    [Content length: 1261]
  Date: Tue, 11 Mar 2014 09:00:50 GMT\r\n
  \r\n
Line-based text data: text/html
```

删除成功，返回 200。