1.[Mrinfo](https://www.freebsd.org/cgi/man.cgi?query=mrinfo&sektion=8)

Mrinfo使用组播协议(IGMP)，对于实现组播协议的路由器，能够获取其本地接口以及其邻居路由器邻接关系的边表。

1. **工具来源**

Mrinfo来源于[mrouted](https://github.com/troglobit/mrouted/releases)工具包。获取mrouted源代码后，进行编译生成mrinfo可执行文件。

注：编译前 apt install byacc 安装此依赖项

1. **测量依赖环境**

测量源主机需要配置公网ip。

1. **命令**

./mrinfo -r 0 -t 1 ip

-r: 设置ask neighbor查询重试次数限制。

-t: 设置等待查询响应的超时秒数

ip: 待查询的目标

1. **输出结果**

./mrinfo -r0 -t1 189.23.123.210

189.23.123.210 (189.23.123.210) [version 12.0]:

200.244.41.158 -> 0.0.0.0 (local) [1/0/pim/querier/leaf]

200.244.22.13 -> 200.244.22.12 (200.244.22.12) [1/0/pim/querier]

200.244.22.15 -> 200.244.22.14 (200.244.22.14) [1/0/pim/querier]

200.244.22.141 -> 200.244.22.140 (200.244.22.140) [1/0/pim/querier]

200.244.22.143 -> 200.244.22.142 (200.244.22.142) [1/0/pim/querier]

目标路由器的接口ip –> neighbor路由器的ip

探测主机：188.166.144.143，密码： b8ff9cf02f2bf1c56a051e877d

自动探测脚本目录：/home/ly/mrouted-3.9.8/auto\_mrinfo\_c.py

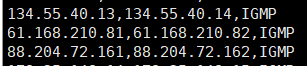
使用方法：python auto\_mrinfo\_c.py ip列表文件 输出原始文件

自动处理脚本目录：/home/ly/mrouted-3.9.8/extract.mrinfo.py

使用方法：python extract.mrinfo.py 输入原始文件 输出边表文件

输出边表文件格式：每行分别是：相邻路由器接口ip对

如图



1. **导入mysql**

处理脚本根目录：10.10.11.130:/home/ly/n4j

1.将预处理后边表文件生成一份顶点文件：

python /home/ly/n4j/mrinfo/extract.mrinfo.ip.py 边文件名 顶点文件名

参数1：边文件名即原始数据处理后的边

参数2：要生成的节点文件名

2.将mrinfo导入mysql命令：

mysql -h 10.10.11.140 -u root -proot < <(cat mrinfo.sql

|sed "s|"create\_date"|2019\_01\_11\_16\_11|g"

|sed "s|"edge\_path"|/home/ly/n4j/mrinfo/mrinfo.edge|g"

|sed "s|"node\_path"|/home/ly/n4j/mrinfo.node|g")

mrinfo.sql位置: /home/ly/n4j/mrinfo/

create\_date:创建表日期，要与后面导入n4j名字一致

edge\_path:边文件路径

node\_path:节点文件路径

2.[pchar](http://www.kitchenlab.org/www/bmah/Software/pchar/README)带宽测量文档

pchar将不同大小的探测数据包发送到网络，并分析由中间路由器或目标主机生成的ICMP消息。通过测量不同大小的数据包的响应时间，pchar可以估计沿路径的带宽和固定的往返延迟。

1. **工具来源**

获取[源码](http://www.kitchenlab.org/www/bmah/Software/pchar/)，进行编译生成pchar可执行文件。

1. **测量依赖环境**

网络状况良好的机器

1. **命令**

**./pchar -n -I 400 -R 3 -t 2 -p ipv4icmp +ip**

-n: 不进行反向域名解析

-I: 数据包大小的增量

-R: 每一跳重复的次数

-t: ICMP响应超时时间

-p: 选择网络协议类型

ip: 测量目标

1. **输出结果**

**./pchar -n -I 400 -R 3 -t 2 -p ipv4icmp 89.211.6.241**

3: 138.197.244.89 (138.197.244.89)

Partial loss: 0 / 9 (0%)

Partial char: rtt = 6.117830 ms, (b = 0.000039 ms/B), r2 = 0.470157

stddev rtt = 0.035213, stddev b = 0.000041

Partial queueing: avg = 0.000034 ms (121737 bytes)

**Hop char: rtt = --.--- ms, bw = 48854.961832 Kbps**

Hop queueing: avg = -0.000046 ms (0 bytes)

其中3为第3跳路由器，Hop char中bw表示上一跳到该路由器接口的链路带宽。

探测主机：188.166.144.143，密码： b8ff9cf02f2bf1c56a051e877d

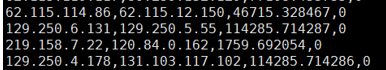
自动探测脚本目录：/home/ly /pchar-1.5/ auto\_pchar.py

使用方法：auto\_pchar.py ip列表文件 输出原始文件

自动处理脚本目录：/home/ly/mrouted-3.9.8/extract.pchar.py

使用方法：python extract.pchar.py 输入原始文件 输出预处理边表文件

预处理边表文件格式: 每行逗号分开，每行分别是：入ip，出ip，带宽



1. **pchar数据导入mysql**

处理脚本根目录：10.10.11.130:/home/ly/n4j

1.将预处理后边表文件生成一份顶点文件：

python /home/ly/n4j/pchar/extract.mrinfo.ip.py 边文件名 节点文件名

参数1：边文件名即原始数据处理后的边

参数2：要生成的节点文件名

2．将pchar导入mysql命令：

mysql -h 10.10.11.140 -u root -proot < <(cat pchar.sql

|sed "s|"create\_date"|2019\_01\_11\_16\_11|g"

|sed "s|"edge\_path"|/home/ly/n4j/pchar/pchar.edge|g"

|sed "s|"node\_path"|/home/ly/n4j/pchar/pchar.node|g")

其中：

pchar.sql位置: /home/ly/n4j/pchar/pchar.sql

create\_date:创建表日期，要与后面导入n4j名字一致

edge\_path:边文件路径

node\_path:节点文件路径

**3.将mrinfo/pchar导入n4j(以mrinfo为例)**

处理脚本根目录：10.10.11.130:/home/ly/n4j

所需文件：mrinfo.node，mrinfo.edge,两个节点、

边表文件在目录/home/ly/n4j/mrinfo下

第一步：在/home/ly/n4j/mrinfo 下建立simpleHttpSever服务，端口34567（随意）

第二步：边、节点文件mrinfo.node，mrinfo.edge导入n4j，命令如下

./import.sh /home/ly/n4j/mrinfo mrinfo 2019\_01\_11\_16\_35\_0 http://10.10.11.130:34567/mrinfo.node http://10.10.11.130:34567/mrinfo.edge

./import.sh位置: /home/ly/n4j/

参数1: /home/ly/n4j/mrinfo ：mrinfo或者pchar数据目录（建立simpleHttpserver，端口34567（随意））

参数2: mrinfo：mrinfo或者pchar，数据表名中的数据类型部分

参数3：数据表名中的日期部分

参数4：http://10.10.11.130:34567/mrinfo.node mrinfo/pchar 节点文件名

参数5：http://10.10.11.130:34567/mrinfo.edge mrinfo/pchar 边表文件名

参数2和参数3确定了表的具体名称

注意：由于建立simpleHttpSever，mrinfo/pchar边和节点文件需要在同一目录，参数1就是在这个目录下建立simpleHttpserver服务