

计算机网络系统设计

卢雨轩 19071125

孙天天 19071110

刘 阳 19071127

2022 年 6 月 19 日

一、 总体设计

采用 Rust 编写 ping 程序。在解析用户传入的命令行参数后，使用 netlink 系统调用获取系统路由表（也可以选择绕过路由表），使用 pnet 库提供的 raw 网络接口构造 ICMP 和 IP 数据包并发送。发送一个数据包的典型流程见 图 1。

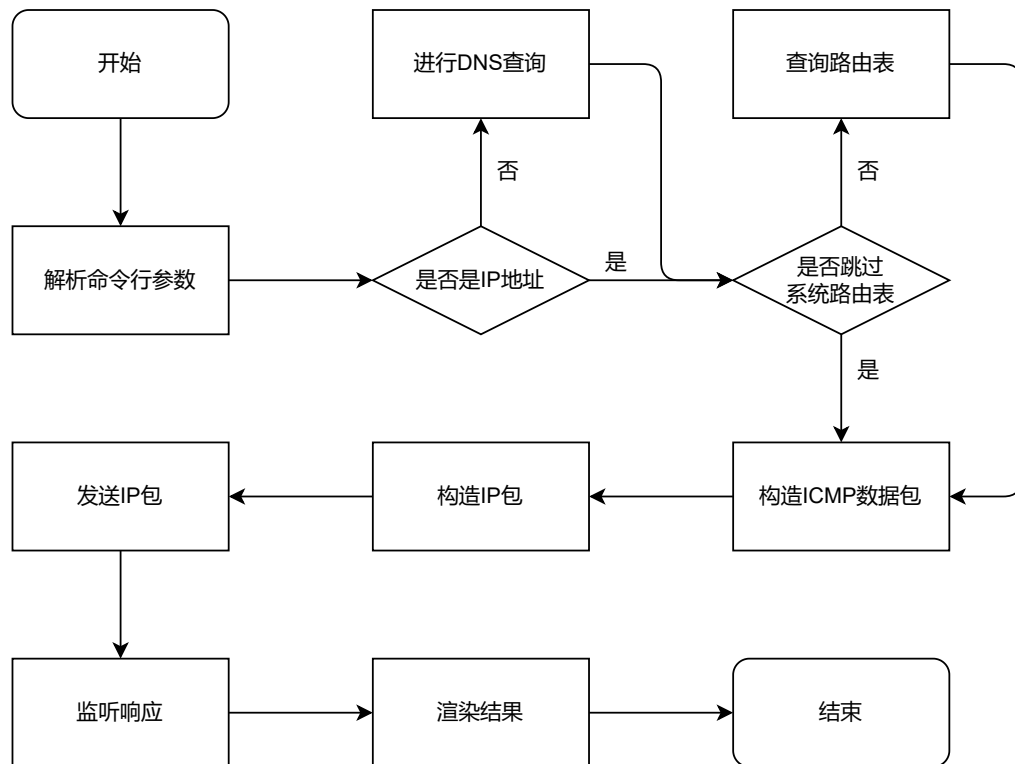


图 1: 发送一个数据包的典型流程

预计实现以下功能：

- 基础 ping 的网络测试功能，包含 Linux 的 ping 软件中较为具有代表性的功能。
 - 如：广播地址、设置 TTL、安静模式、包数量、包大小等
- traceroute：通过不断增加发的包的 TTL，根据返回的 TimeExceeded 包的源地址来跟踪从当前机器到目标机器的路由链路。通过 geoip 服务获取 ip 地址的地理位置信息并展示。
- CLI 图像绘制功能：绘制拓扑意义上的网络连接图和统计意义上的延迟折线图，来更形象地展示网络测试结果。

二、 ICMP 部分

本功能块负责构造 ICMP Echo Request 包。相关代码如下：

```
let mut vec: Vec<u8> = vec![0; 16]; // 包长度
// Use echo_request so we can set the identifier and sequence number
let mut echo_packet = echo_request::MutableEchoRequestPacket::new(&mut
    ↪ vec[..]).unwrap();
echo_packet.set_sequence_number(20); // SEQ 字段
echo_packet.set_identifier(2);      // Ident 字段
echo_packet.set_icmp_type(IcmpTypes::EchoRequest);

let csum = util::checksum(echo_packet.packet(), 1);
echo_packet.set_checksum(csum);
```

三、 TCP 部分

本模块负责构造 IP 数据包。相关代码如下：

```
let mut ip_vec: Vec<u8> = vec![0; Ipv4Packet::minimum_packet_size() + 16];
let mut ip_packet = MutableIpv4Packet::new(&mut ip_vec[..]).unwrap();

let total_len = (20 + 16) as u16;

ip_packet.set_version(4);
ip_packet.set_header_length(5);
ip_packet.set_total_length(total_len); // 总长度
ip_packet.set_ttl(128);               // TTL
ip_packet.set_next_level_protocol(IpNextHeaderProtocols::Icmp); // ICMP 协议
ip_packet.set_source(Ipv4Addr::new(172, 31, 135, 147)); // 源地址
ip_packet.set_destination(Ipv4Addr::new(172, 31, 143, 255)); // 目标地址

let checksum = ipv4::checksum(&ip_packet.to_immutable());
ip_packet.set_checksum(checksum); // 计算校验码
ip_packet.set_payload(echo_packet.packet());
```

四、 CLI 图像绘制部分

本课设支持的 CLI 图像绘制模块旨在直观地展示网络环境状况。具体地说，系统支持绘制网络连接的拓扑结构图和延迟测试折线统计图。

4.1 网络连接拓扑结构图

我们利用来自于多次 traceroute 和 ping 得到的数据绘制网络中各设备的拓扑关系图。用户可以开关是否要将接下来的测试结果放入拓扑结构图的绘制中。

使用类似 git 提交结点可视化的方式，效果大致如下所示。

```
* 10.0.0.2 (内网)
|
* 10.0.0.1 (内网)
|\
* | 114.5.14.19 (北京市 北京 联通)
| * 19.8.1.0 (北京市 北京 联通)
|\ \
* | | 218.197.48.78 (湖北省 十堰市 教育网)
  * | 11.45.1.4 (美国 俄亥俄)
    * 191.9.8.10 (巴西)
```

4.2 延迟折线统计图

我们计划用折线统计图的方式实时显示最近几次 ping 的延迟数据，这样可以直观地观察延迟变化。

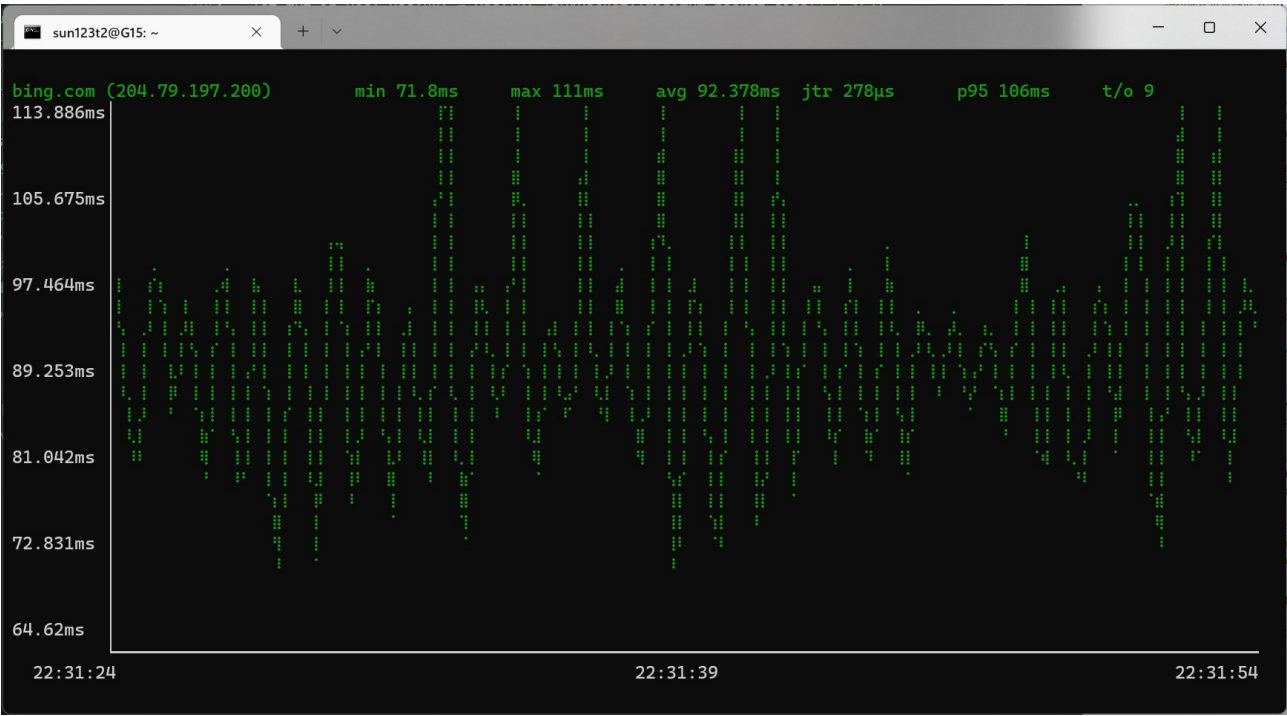


图 2: 延迟折线统计图设想图示

效果大致如图2所示，当统计图未填满终端时从左边向右生长，填满时则整体向左平移滚动显示。