

计算机网络实验报告-04

阮星程

2015K8009929047

一、 实验题目

广播网络实验。

二、 实验内容

- 实现 main.c 中的 broadcast 部分，完成 three_nodes_bw.py 中拓扑结构的各个终端相互 ping 通。
- 利用 iperf 测试网络链路的利用效率。
- 自己书写新的拓扑结构，观察数据包在环路中的不断转发。

三、 实验过程

本次实验要书写的代码很少，绝大多数的内容老师已经搭建完成，重在理解老师以搭建好的框架。

花费约三小时进行代码部分的阅读以及理解，对各个函数的功能进行了简单的标注

```
//from iface_list find corresponding iface
static iface_info_t *fd_to_iface(int fd)
{
    iface_info_t *iface = NULL;
    list_for_each_entry(iface, &instance->iface_list, list) {
        if (iface->fd == fd)
            return iface;
    }

    log(ERROR, "Could not find the desired interface according to fd %d", fd);
    return NULL;
}

//send packet to iface
void iface_send_packet(iface_info_t *iface, const char *packet, int len)
{
    struct sockaddr_ll addr;
    memset(&addr, 0, sizeof(struct sockaddr_ll));
    addr.sll_family = AF_PACKET;
    addr.sll_ifindex = iface->index;
    addr.sll_halen = ETH_ALEN;
    addr.sll_protocol = htons(ETH_P_ARP);
    struct ether_header *eh = (struct ether_header *)packet; //assign packet address to eh
    memcpy(addr.sll_addr, eh->ether_dhost, ETH_ALEN);

    if (sendto(iface->fd, packet, len, 0, (const struct sockaddr *)&addr,
               sizeof(struct sockaddr_ll)) < 0) {
        perror("Send raw packet failed");
    }
}
```

之后开始书写，由于具体框架在 PPT 上已经给出，其余部分的代码大多已经了解，所以花了 3 分钟完成了代码。

```
void broadcast_packet(iface_info_t *iface, const char *packet, int len)
{
    // TODO: broadcast packet
    iface_info_t *iface_ptr;
    static int i = 0;
    list_for_each_entry(iface_ptr, &instance->iface_list, list) {
        if (iface_ptr != iface) {
            iface_send_packet(iface_ptr, packet, len);
            printf("%d: packet %s send! dest iface: fd %d index %d name %s\n", i++, packet, iface_ptr->fd, iface_ptr->index, iface_ptr->name);
        }
    }
    //fprintf(stdout, "TODO: broadcast packet here.\n");
}
```

后面实验的拓扑框架也很容易构建，增加对应的终端即连线即可

```
class BroadcastTopo(Topo):
    def build(self):
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        h3 = self.addHost('h3')
        b1 = self.addHost('b1')
        b2 = self.addHost('b2')
        b3 = self.addHost('b3')

        self.addLink(h1, b1, bw=20)
        self.addLink(h2, b2, bw=10)
        self.addLink(h3, b3, bw=10)
        self.addLink(b1, b2, bw=10)
        self.addLink(b1, b3, bw=10)
        self.addLink(b2, b3, bw=10)

if __name__ == '__main__':
    topo = BroadcastTopo()
    net = Mininet(topo = topo, link = TCLink, controller = None)

    h1, h2, h3, b1, b2, b3 = net.get('h1', 'h2', 'h3', 'b1', 'b2', 'b3')
```

至此实验的框架代码已经构建完毕。

四、实验结果

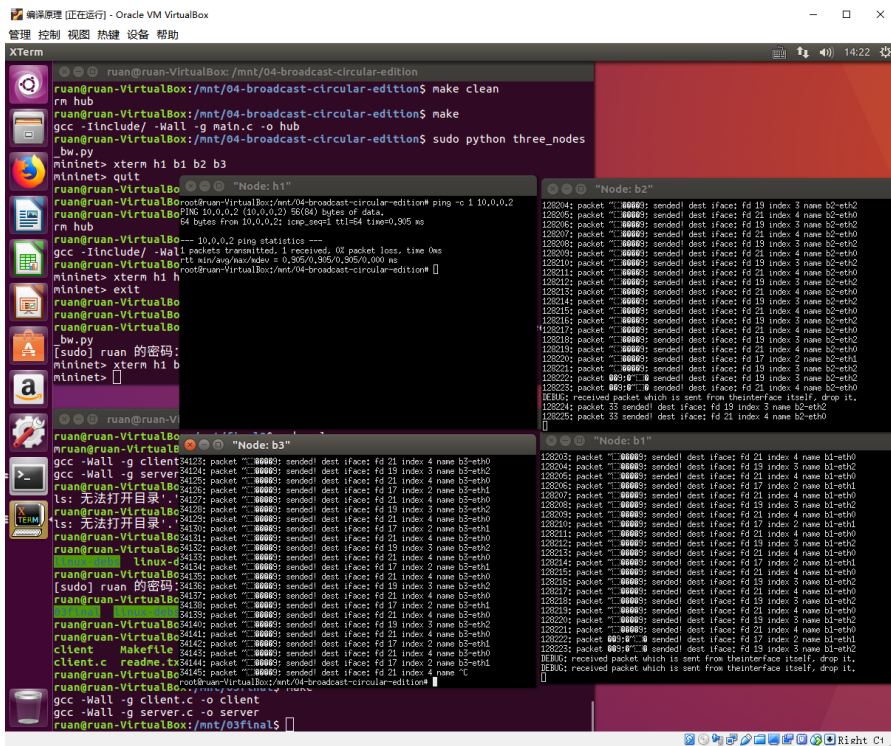
本节直接书写运行结果，具体的运行代码请见 `readme.txt`

构建拓扑网络，测试 3 台 host 之间是否能够 ping 通，以此测试转发效果：

[illegible]

通过 iperf 指令观察网络的利用效率:

以 h1 节点中的内容为例，上面两个部分为 h2,h3 作为 servers，h1 作为 client 去连接；下面的结果为 h2,h3 作为 clients 连接 server h1。



五、 结果分析

从结果来看，3 个目标我们都已经很好地实现：

- 实现 broadcast 部分确实能够很好地实现转发，各个连接到 hub 的节点也能够相互 ping 通。
- 测试链路效率时，由 h1 去连接其它终端，由于 hub 到 h2,h3 的带宽限制，及时 hub 到 h1 的带宽有 20Mb/s，实际能够发挥的带宽也只接近 hub 到 h2,h3 的带宽 10Mb/s；反过来，h2,h3 同时连接 h1，与带宽的分布相符，几乎三条链路都接近了最大的带宽；但是也能注意到，在同时连接时的速度是要比单独连接要慢的。
- 在环路中，一个包发出后被 hub 循环多次转发，在环路中一直旋转，浪费了很多的网络资源，这样做在今后的实现中需要避免。