

南京大学本科生实验报告

课程名称： 计算机网络

任课教师： 田臣/李文中

助教：

学院	计算机科学与技术系	专业	计算机科学与技术
学号	185220001	姓名	磯田智明
Email	185220001@smail.nju.edu.cn	开始/完成日期	2021.05.23/2021.05.25

1. 实验名称

Reliable Communication

2. 实验目的

- 设计可靠通信机制

3. 实验内容

Task 1: Preparation

配置实验环境

Task 2: Middlebox

实现Middlebox功能，其会按照一个概率丢弃包，从而模拟网络中丢包的情况

Task 3: Blastee

实现Blastee功能，其作为接收端

Task4: Blaster

实现Blaster功能，其作为发送端

Task5: Running your code

测试Middlebox，Blastee和Blaster实际运行情况

4. 实验结果

Task 2: Middlebox

The features of middlebox

判断发来的包是哪个端口传来的，如果是从 `middlebox-eth0` 则说明该包是从 `blaster` 发来的，将包的源mac地址改为 `middlebox-eth1` 的mac地址，目的地址改为 `blastee` 的mac地址即可；如果该包是从 `middlebox-eth1` 传来的话，与前一种情况类似。

在收到包之后，还会按照一定的丢包。其中丢包的具体实现是会在1到100产生一个随机数，因为丢包的概率为0.19。如果这个数大于20则会被转发，小于20则会被丢弃

Task 3: Blastee

The features of Blastee

Blastee 会收到来自 Blaster 的包，该包的结构如下：

```
# <----- Switchyard headers -----> <----- Your packet header(raw bytes) -----> <-- Payload in raw bytes --->
# -----
# |  ETH Hdr |  IP Hdr  |  UDP Hdr  | Sequence number(32 bits) | Length(16 bits) |  Variable length payload  |
# -----
```

Blastee 回复的包格式应为，如下结构：

```
# <----- Switchyard headers -----> <----- Your packet header(raw bytes) -----> <-- Payload in raw bytes --->
# -----
# |  ETH Hdr |  IP Hdr  |  UDP Hdr  |           Sequence number(32 bits)           |           Payload   (8 bytes)   |
# -----
```

所以要构造一个 Blastee 发送给 Blaster 的包，首先设置好ETH，IP和UDP包头。其中ETH和IP包头的源地址都为 Blastee 的mac地址和ip地址，目的地址为 Blaster 的mac地址和ip地址。

由 Blaster 发来包的结构可知， packet[3] 中的第0到4字节存放着 Sequence number ；第4到6字节存放着 Length ；第6字节开始存放着 payload 。

所以在构造 Blastee 包的时候就要将 Sequence number 设置为 packet[3] 中的第0到4字节；Payload 设置为 packet[3] 从第6字节开始的八个字节即可

Task4: Blaster

The features of Blaster

本节逻辑主要体现在是两个函数模块中，分别是 handle_packet 和 handle_no_packet 。

- 其中 handle_packet 主要处理从 Blastee 发往 Blaster 的ACK包
- 其中 handle_no_packet 主要处理内容是向 Blastee 发送数据包，并且重新发送没有收到ACK的数据包

handle_packet 逻辑

读出收到包的 Sequence number 并将该序号对应的数据做好标记，表示为该包已经被收到不需要被重传。并且还要及时的更新 LHS

handle_no_packet 逻辑

首先判断 LHS 序号对应的包是否超时，如果超时需要进行一次重传；否则需要判断目前 RHS 和 LHS 的位置判断是否超过发送窗口的大小，如果没有超过则可以发送新的包，并且更新 RHS 的值

Task5: Running your code

Running in the Mininet

在终端中输入以下指令启动mininet

```
1 | $ sudo python start_mininet.py
```

在mininet中启动 middlebox , blastee 和 blaster

```
1 | mininet> xterm middlebox
2 | mininet> xterm blastee
3 | mininet> xterm blaster
```

并且在各自的xterm中运行如下指令：

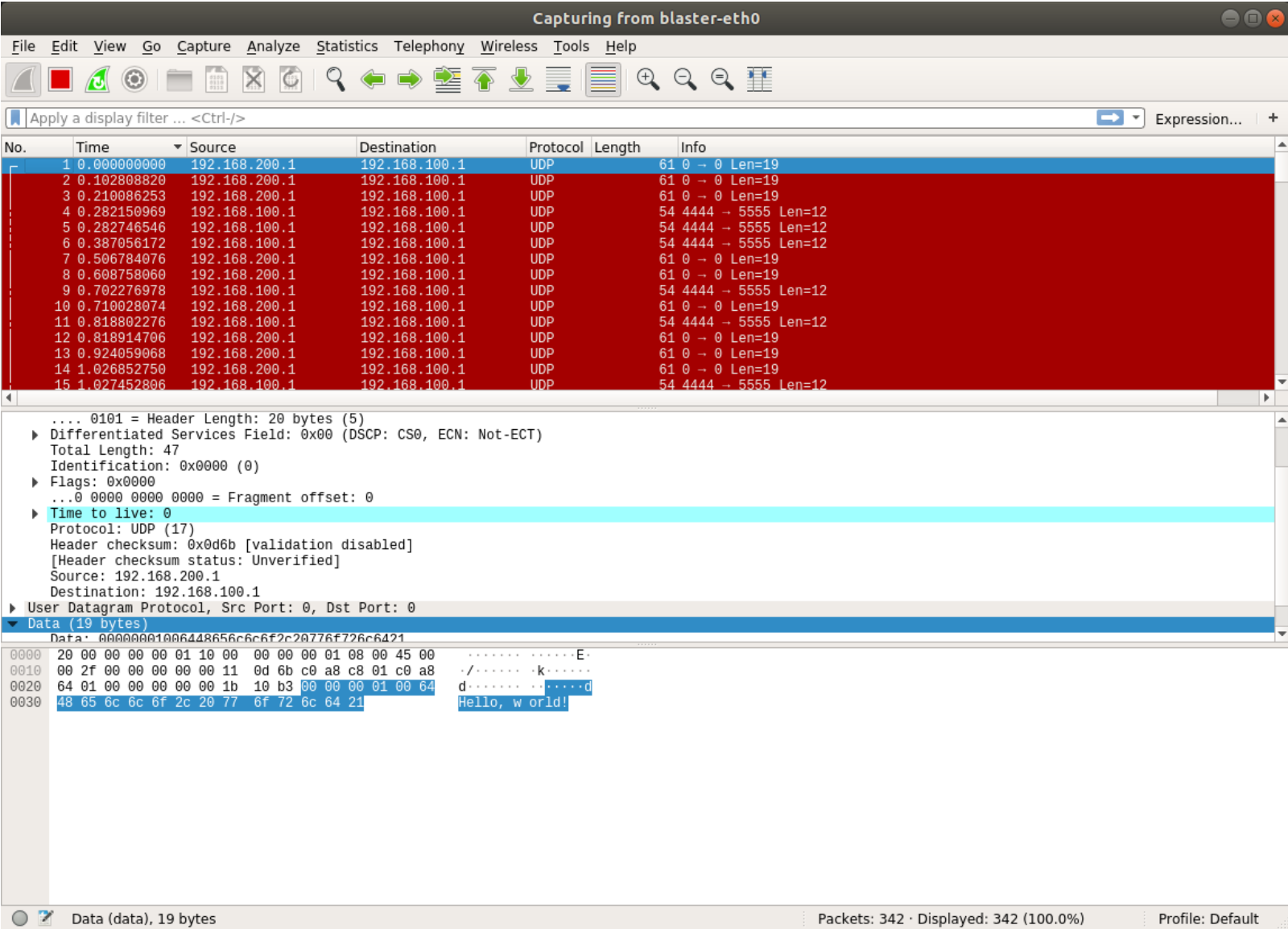
```
1 | middlebox# swyard middlebox.py -g 'dropRate=0.19'
2 | blastee# swyard blastee.py -g 'blasterIp=192.168.100.1 num=100'
3 | blaster# swyard blaster.py -g 'blasteeIp=192.168.200.1 num=100 length=100
   | senderWindow=5 timeout=300 recvTimeout=100'
```

利用wireshark抓 middlebox , blastee 和 blaster 的包

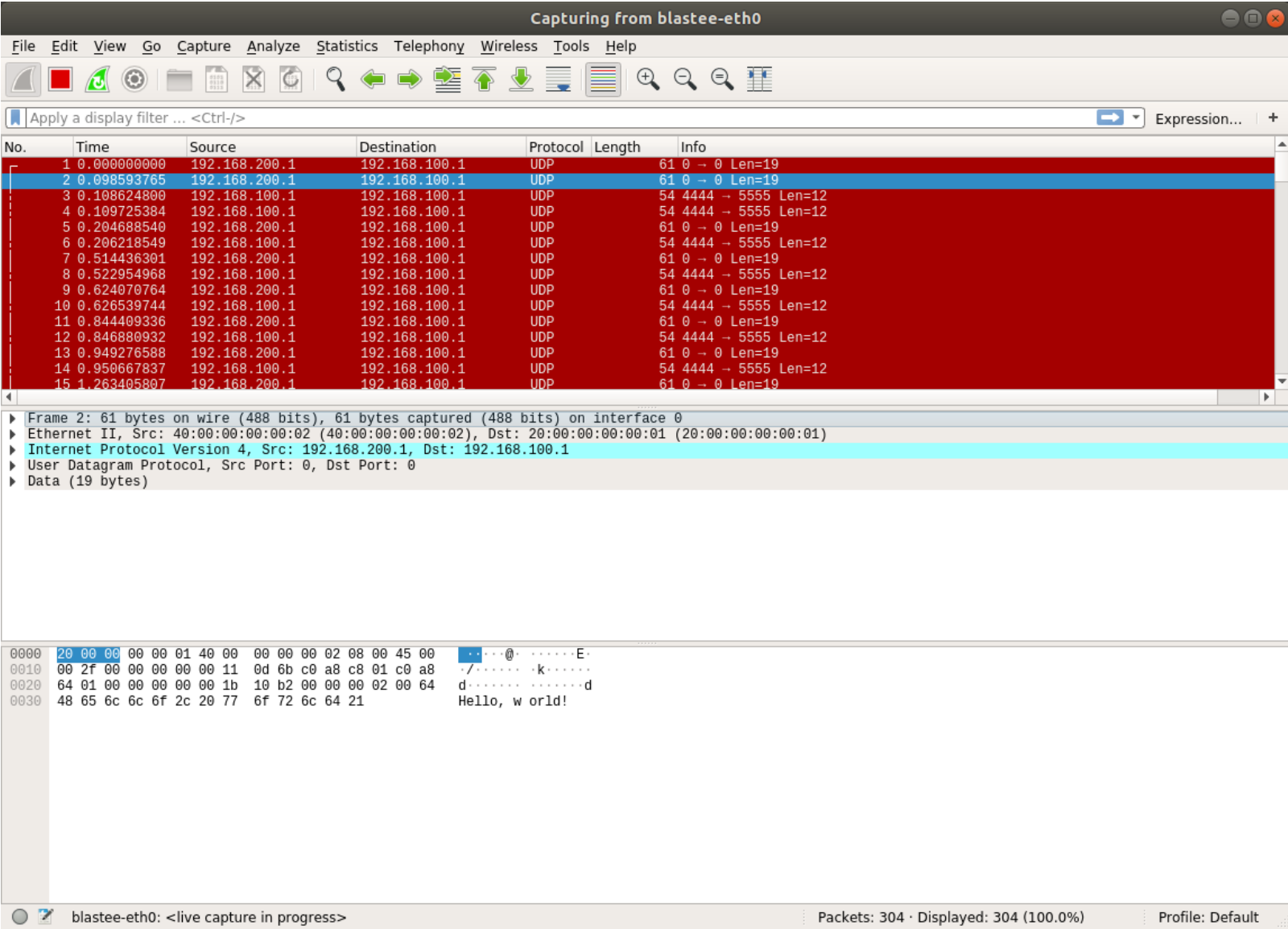
```
1 middlebox# wireshark -i middlebox
2 blastee# wireshark -i blastee
3 blaster# wireshark -i blaster
```

wireshark中的结果

blaster的抓包情况



blastee的抓包情况



middlebox的抓包情况


```
1 sequence = struct.pack(">4s", packet[3].to_bytes()[0:4])
2 payload = struct.pack(">8s", packet[3].to_bytes()[6:14])
```

Task 4: Blaster

```
1 if (time.time() - self.LHS_timer) > self.timeout:
2     sequence_number = self.LHS.to_bytes(4, "big")
3     Length = self.length.to_bytes(2, "big")
4     variable_length_payload = struct.pack(">13s", bytes("hello,
world!".encode('utf-8')))
5     log_info (f"Retransmitting pakcet from blaster to blastee, packet info
{pkt}")
6     self.net.send_packet("blaster-eth0", pkt)
7 elif (self.RHS - self.LHS + 1 <= self.senderwindow) and
(self.sent_pkt_flag[self.num-1] == 0):
8     sequence_number = self.RHS.to_bytes(4, "big")
9     Length = self.length.to_bytes(2, "big")
10    variable_length_payload = struct.pack(">13s", bytes("Hello,
world!".encode('utf-8')))
11    log_info(f"Sending packet from blaster to blastee, pkt info {pkt}")
12    self.net.send_packet("blaster-eth0", pkt)
13    self.sent_pkt_flag[self.RHS] = 1
14    if (self.RHS - self.LHS + 1 < self.senderwindow) and (self.RHS <
self.num):
15        self.RHS += 1
```

通过判断 LHS 序号对应的包是否超时，如果超时需要进行一次重传；否则需要判断目前 RHS 和 LHS 的位置判断是否超过发送窗口的大小，如果没有超过则可以发送新的包

6. 总结与感想

在阅读框架代码时发现头文件引用了struct包，由此进一步学习了struct的 pack 和 unpack 方法