**Ex6A:**

**1) Chia modulo-2**

Ta thực hiện chia nhị phân (mod-2) — mỗi lần gặp bit 1 ở vị trí hiện tại thì XOR khối bằng độ dài generator.

Bắt đầu với 1001100000. Generator = 1011 (4 bit).

Từng bước (ghi vị trí bắt đầu, khối 4 bit đang xét, kết quả XOR 4 bit, trạng thái dãy sau khi XOR):

1. Vị trí 0: khối = 1001
   1001 XOR 1011 = 0010
   Dãy trở thành: 0010100000
2. Vị trí 1: khối (vị trí 1..4) = 0101 — **bỏ qua** vì bit dẫn ở vị trí 1 là 0 (không chia)
   Dãy vẫn: 0010100000
3. Vị trí 2: khối = 1010
   1010 XOR 1011 = 0001
   Dãy trở thành: 0000010000
4. Vị trí 3: khối = 0010 — **bỏ qua** (bit dẫn = 0)
   Dãy vẫn: 0000010000
5. Vị trí 4: khối = 0100 — **bỏ qua** (bit dẫn = 0)
   Dãy vẫn: 0000010000
6. Vị trí 5: khối = 1000
   1000 XOR 1011 = 0011
   Dãy trở thành: 0000000110
7. Vị trí 6: khối = 0110 — **bỏ qua** (bit dẫn = 0)
   Kết thúc, dãy cuối cùng: 0000000110

Ở đây phần dư là **3 bit cuối cùng** (do generator độ 3): 110.

**2) Kết luận — CRC**

- **CRC (phần dư)** = 110.
- **Khung truyền (transmitted frame)** = dữ liệu gốc + CRC = 1001100 + 110 = **1001100110**.

**3) Kiểm tra ở phía nhận (receiver)**

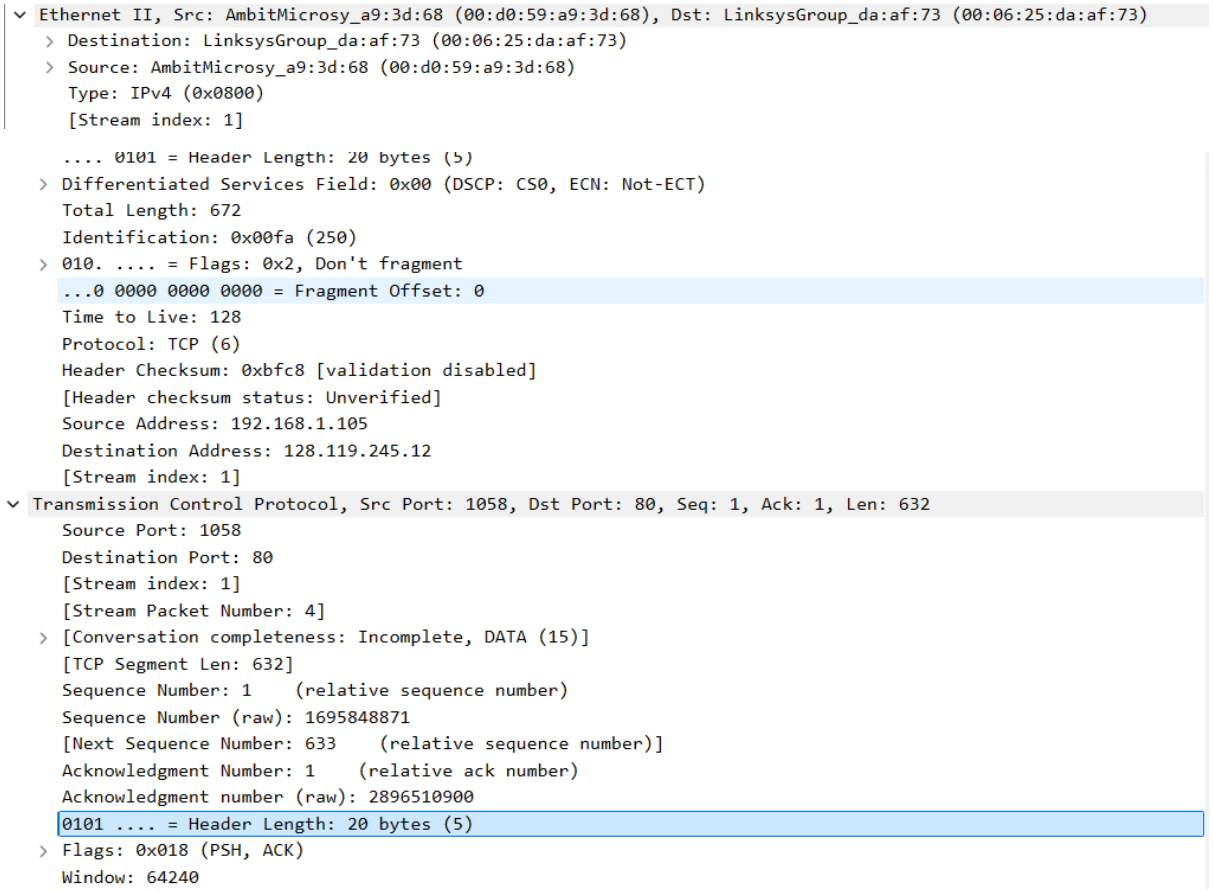
Bên nhận nhận chuỗi 1001100110. Ta thực hiện chia modulo-2 chuỗi này cho 1011. Nếu phần dư = 000 (tất cả bit 0) thì không có lỗi được phát hiện.

Thực hiện chia (tương tự như trên) cho 1001100110 cho 1011.

Bên nhận lấy khung 1001100110 và chia cho cùng generator 1011. Nếu phần dư = 000 thì hợp lệ.

**Lab6A:**

**1**



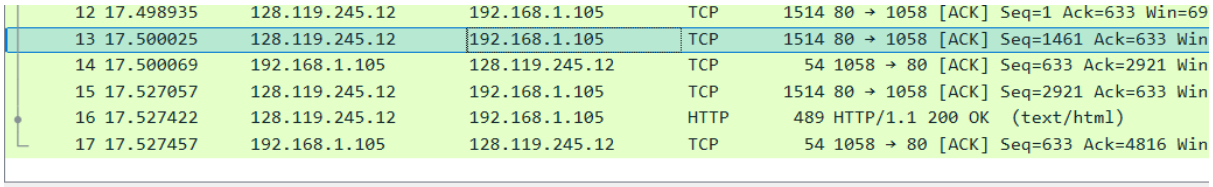

**A** Source: AmbitMicrosy_a9:3d:68 (00:d0:59:a9:3d:68) (A)

**B** Destination: 00:06:25:da:af:73 (R)

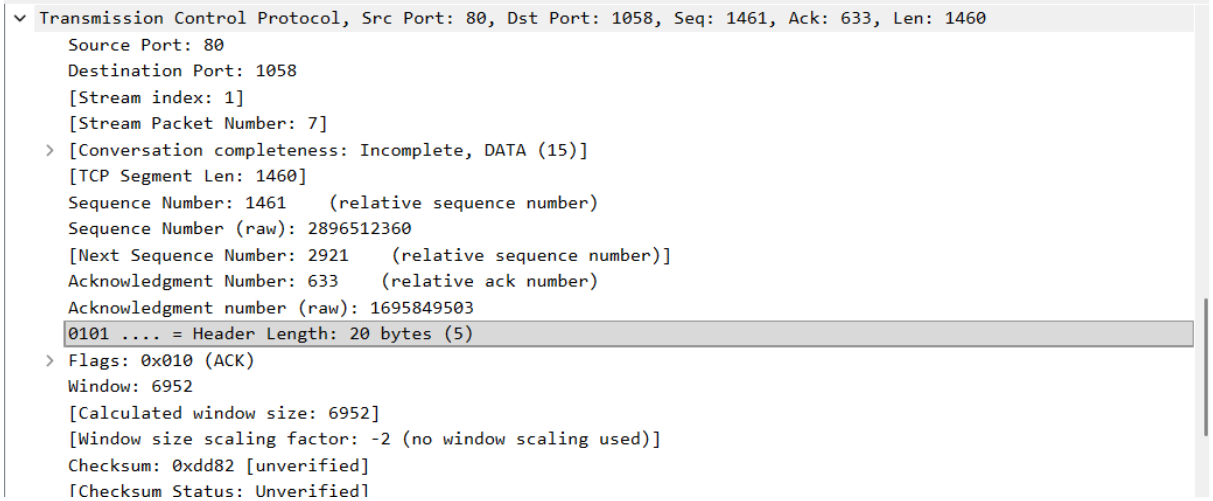
**C** Type: IPv4 (0x0800)

**D** 54 bytes: Ethernet 14 bytes, IP 20 bytes, TCP 20 bytes. (G)

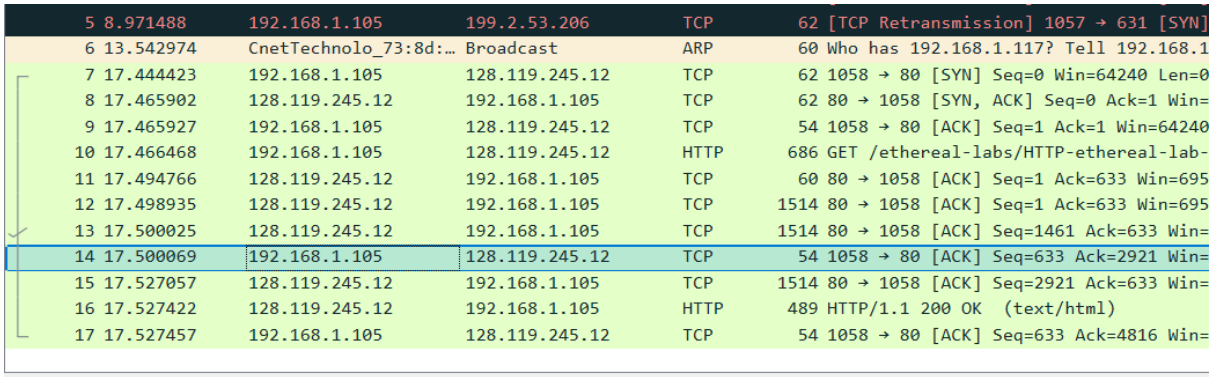
**2**



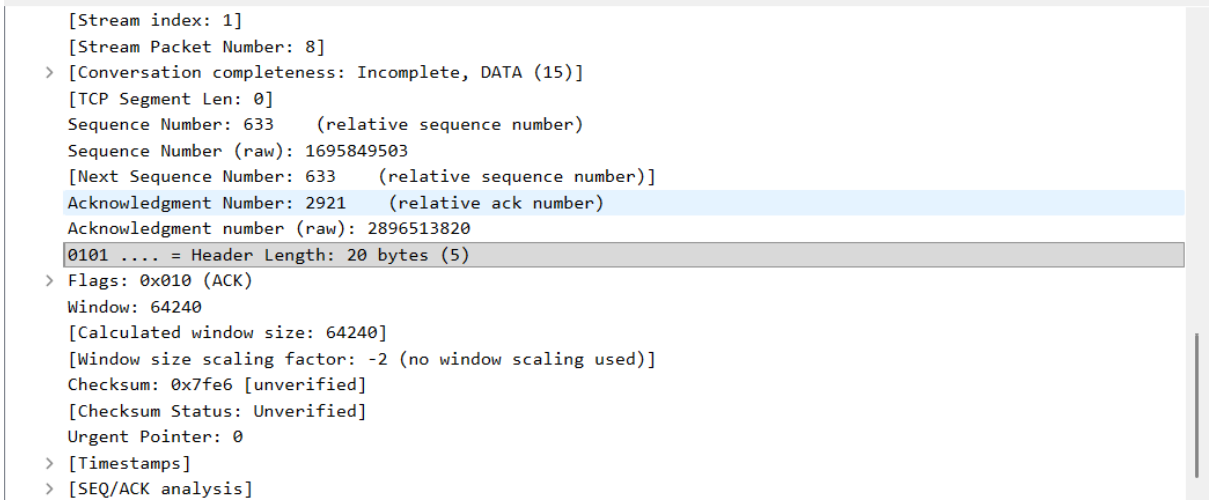
| | | | | | |
|---|---|---|---|---|---|
| 12 | 17.498935 | 128.119.245.12 | 192.168.1.105 | TCP | 1514 80 → 1058 [ACK] Seq=1 Ack=633 Win=69 |
| 13 | 17.500025 | 128.119.245.12 | 192.168.1.105 | TCP | 1514 80 → 1058 [ACK] Seq=1461 Ack=633 Win |
| 14 | 17.500069 | 192.168.1.105 | 128.119.245.12 | TCP | 54 1058 → 80 [ACK] Seq=633 Ack=2921 Win |
| 15 | 17.527057 | 128.119.245.12 | 192.168.1.105 | TCP | 1514 80 → 1058 [ACK] Seq=2921 Ack=633 Win |
| 16 | 17.527422 | 128.119.245.12 | 192.168.1.105 | HTTP | 489 HTTP/1.1 200 OK (text/html) |
| 17 | 17.527457 | 192.168.1.105 | 128.119.245.12 | TCP | 54 1058 → 80 [ACK] Seq=633 Ack=4816 Win |



```
Transmission Control Protocol, Src Port: 80, Dst Port: 1058, Seq: 1461, Ack: 633, Len: 1460
    Source Port: 80
    Destination Port: 1058
    [Stream index: 1]
    [Stream Packet Number: 7]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 1460]
    Sequence Number: 1461    (relative sequence number)
    Sequence Number (raw): 2896512360
    [Next Sequence Number: 2921    (relative sequence number)]
    Acknowledgment Number: 633    (relative ack number)
    Acknowledgment number (raw): 1695849503
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window: 6952
    [Calculated window size: 6952]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0xdd82 [unverified]
    [Checksum Status: Unverified]
```



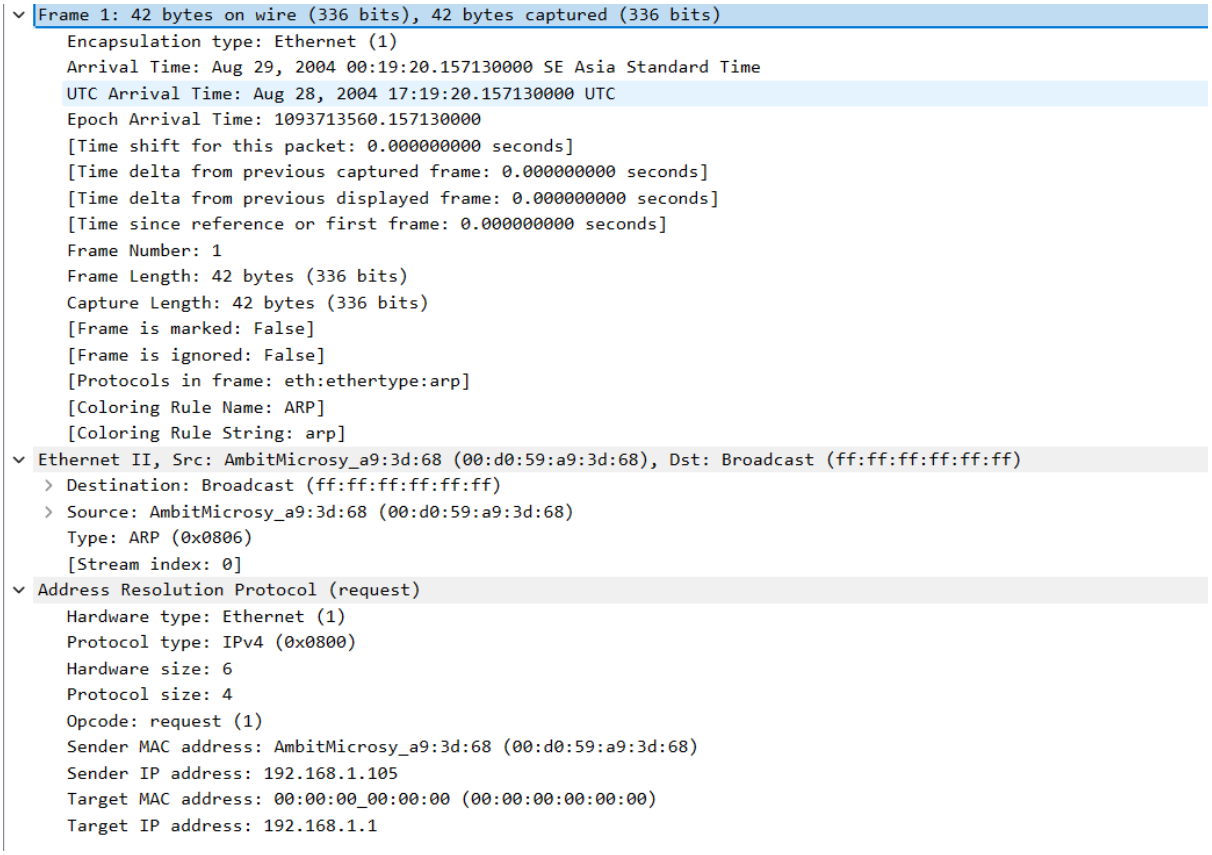
| | | | | | |
|---|---|---|---|---|---|
| 5 | 8.971488 | 192.168.1.105 | 199.2.53.206 | TCP | 62 [TCP Retransmission] 1057 → 631 [SYN] |
| 6 | 13.542974 | CnetTechnolo_73:8d:… | Broadcast | ARP | 60 Who has 192.168.1.117? Tell 192.168.1 |
| 7 | 17.444423 | 192.168.1.105 | 128.119.245.12 | TCP | 62 1058 → 80 [SYN] Seq=0 Win=64240 Len=0 |
| 8 | 17.465902 | 128.119.245.12 | 192.168.1.105 | TCP | 62 80 → 1058 [SYN, ACK] Seq=0 Ack=1 Win= |
| 9 | 17.465927 | 192.168.1.105 | 128.119.245.12 | TCP | 54 1058 → 80 [ACK] Seq=1 Ack=1 Win=64240 |
| 10 | 17.466468 | 192.168.1.105 | 128.119.245.12 | HTTP | 686 GET /ethereal-labs/HTTP-ethereal-lab- |
| 11 | 17.494766 | 128.119.245.12 | 192.168.1.105 | TCP | 60 80 → 1058 [ACK] Seq=1 Ack=633 Win=695 |
| 12 | 17.498935 | 128.119.245.12 | 192.168.1.105 | TCP | 1514 80 → 1058 [ACK] Seq=1 Ack=633 Win=695 |
| 13 | 17.500025 | 128.119.245.12 | 192.168.1.105 | TCP | 1514 80 → 1058 [ACK] Seq=1461 Ack=633 Win= |
| 14 | 17.500069 | 192.168.1.105 | 128.119.245.12 | TCP | 54 1058 → 80 [ACK] Seq=633 Ack=2921 Win= |
| 15 | 17.527057 | 128.119.245.12 | 192.168.1.105 | TCP | 1514 80 → 1058 [ACK] Seq=2921 Ack=633 Win= |
| 16 | 17.527422 | 128.119.245.12 | 192.168.1.105 | HTTP | 489 HTTP/1.1 200 OK (text/html) |
| 17 | 17.527457 | 192.168.1.105 | 128.119.245.12 | TCP | 54 1058 → 80 [ACK] Seq=633 Ack=4816 Win= |



```
    [Stream index: 1]
    [Stream Packet Number: 8]
  > [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 633    (relative sequence number)
    Sequence Number (raw): 1695849503
    [Next Sequence Number: 633    (relative sequence number)]
    Acknowledgment Number: 2921    (relative ack number)
    Acknowledgment number (raw): 2896513820
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window: 64240
    [Calculated window size: 64240]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x7fe6 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > [Timestamps]
  > [SEQ/ACK analysis]
```

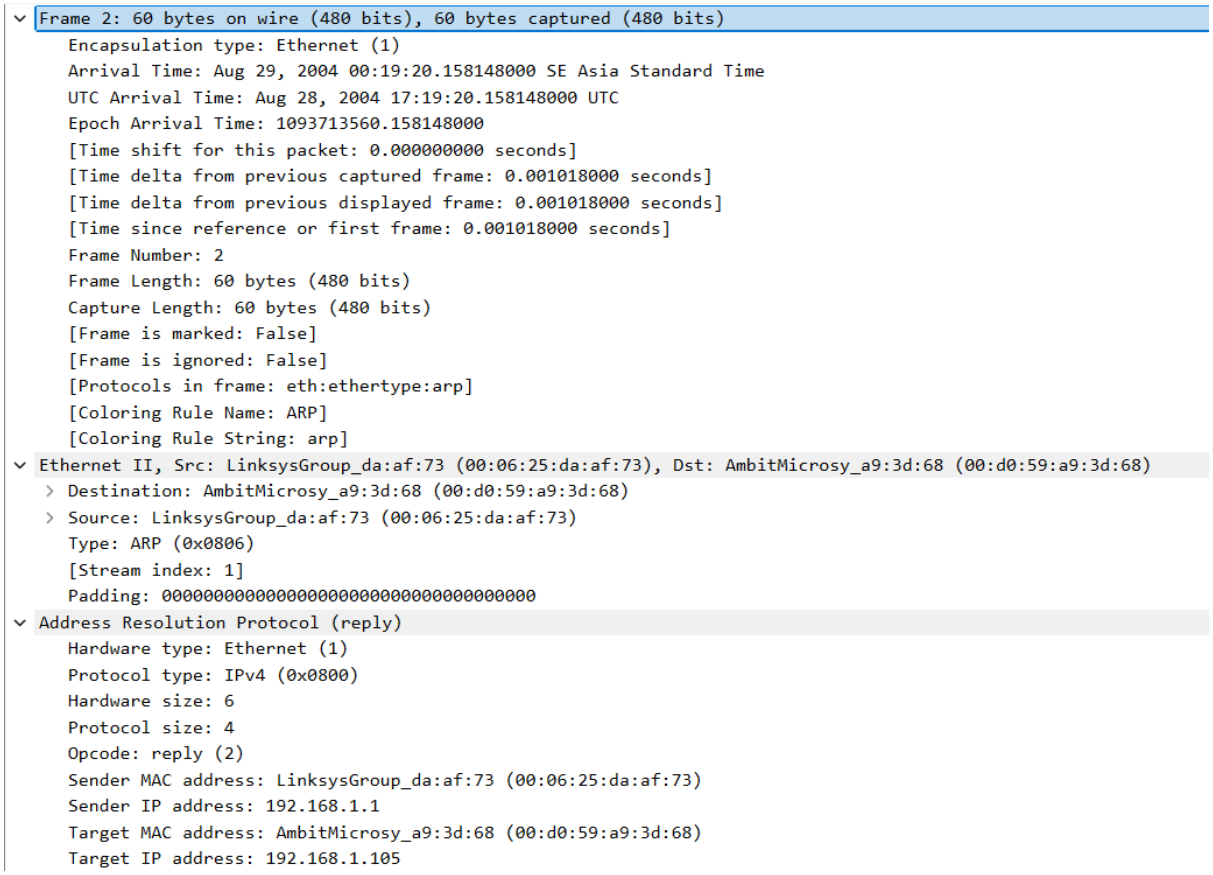
A. 00:06:25:da:af:73 — thuộc R
B. 00:d0:59:a9:3d:68 — thuộc A
C. 0x0800 — IPv4
D. Byte offset đến ký tự 'O' = 54 (bytes). Header trước payload: Ethernet 14 bytes, IP 20 bytes, TCP 20 bytes.

**3**




**A.** 00:d0:59:a9:3d:68 , ff:ff:ff:ff:ff:ff
**B.** 0x0806
**C.** 20
**D.** 0x0001
**E.** 192.168.1.105
**F.** 00:00:00:00:00:00 , 192.168.1.1

**4**




**A.** 00:06:25:da:af:73, 00:d0:59:a9:3d:68
**B.** 0x0806
**C.** 20
**D.** 0x0002
**E.** 192.168.1.1
**F.** 00:d0:59:a9:3d:68, 192.168.1.105