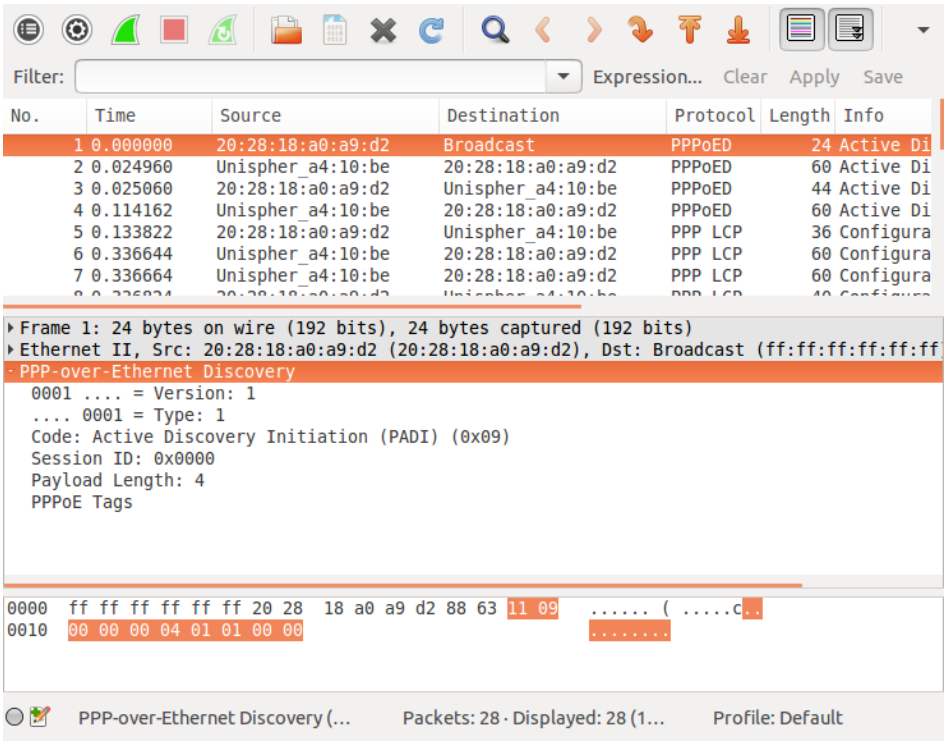
Computer Networks Assignment #3 Zhang Zhexian (0545080) zhangzhexian@outlook.com

1. [Chapter 3 Hands-on 4] Use Sniffer or similar software to find out how many kinds of “protocol types” in the “Type” field of the Ethernet frames you capture. What transport/application layer protocols, if any, do they belong to?

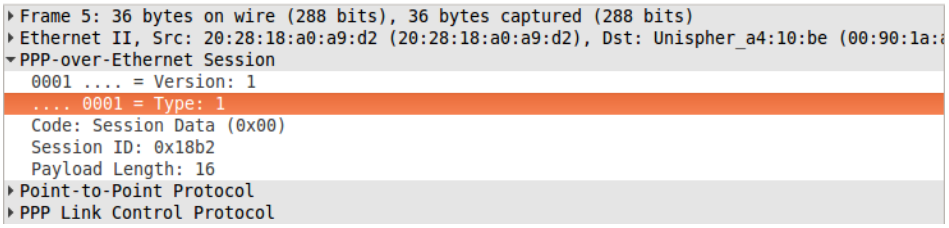
   The Sniffer I used is Wireshark in Linux Ubuntu, and the capture file is obtained via https://wiki.wireshark.org/SampleCaptures#Point-To-Point_over_Ethernet

   There are several protocol types, for example:

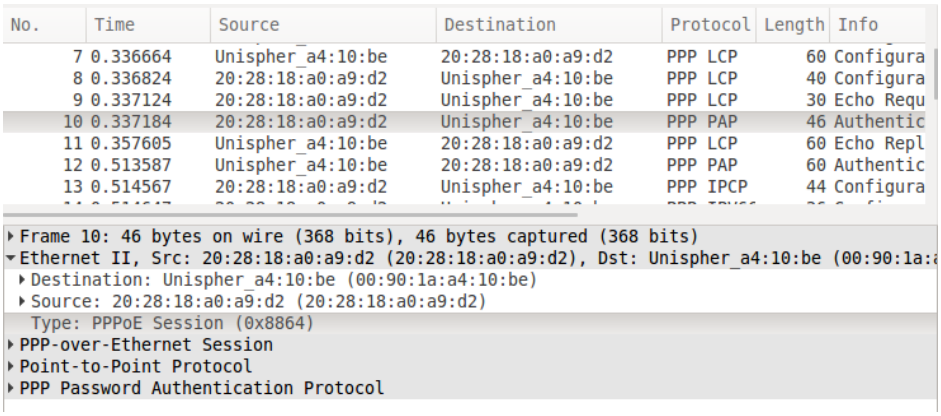
   PPPoED is the Point-to-point protocol over Ethernet, a link layer protocol



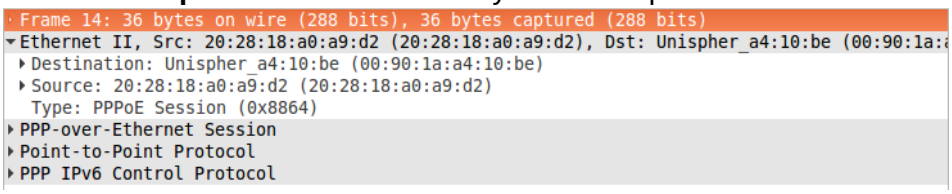
   PPP link control protocol is another protocol that’s in the link layer



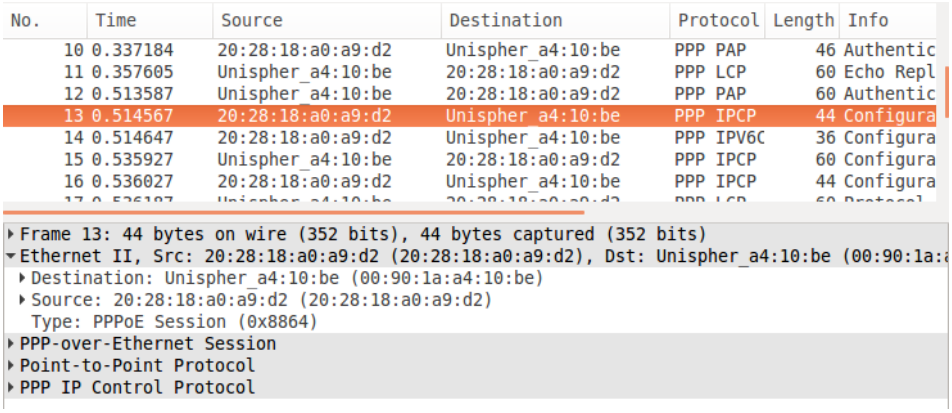
PPP password authentication protocol



IPv6 control protocol is a network layer Internet protocol



Similarly for **PPP IP control protocol**



2. [Chapter 3 Hands-on 7] After making the kernel and choosing some drivers to be modularized, how do we compile the driver, install the driver, and run these modules? Please also compose one small module to validate your answer. Show what commands are needed to compile and install your module. How do you verify whether your module has been installed successfully? (Hint: Read insmod(8), rmmod(8), and lsmod(8).)

   We compile the driver using gcc, for example, gcc –c hello.c

   We then install the module using insmod(8), in this case the command is insmod hello.o

   To verify that the module has been installed correctly, we use lsmod(8) that prints the contents of the /proc/modules file. It shows which loadable kernel modules are currently loaded. It should match the driver module we installed.

   When the module is no longer needed, we may remove it by rmmod(8).

3. [Chapter 3 Written 2] Read RFC 1071 and RFC 1624 to see how IP checksum is computed. Then practice with the following trivial blocks of words by hand:
   0x36f7 0xf670 0x2148 0x8912 0x2345 0x7863 0x0076
   What if the first word above is changed to 0x36f6? RFCs are available at http://www.ietf.org/rfc.html.

Number	Files	Title	Authors	Date	More Info	Status
RFC 1071	ASCII, PDF	**Computing the Internet checksum**	R.T. Braden, D.A. Borman, C. Partridge	September 1988	Updated by RFC 1141, Errata	Informational (changed from Unknown April 2016)

Number	Files	Title	Authors	Date	More Info	Status
RFC 1624	ASCII, PDF	**Computation of the Internet Checksum via Incremental Update**	A. Rijsinghani, Ed.	May 1994	Updates RFC 1141, Errata	Informational

By reading the documents, I learnt that IP checksum is computed by calculating the 16-bit 1's complement sum.

To check a checksum, the 1's complement sum is computed over the same set of octets, including the checksum field. If the result is all 1 bits, the check succeeds.

Word (Hex)	Word (Binary)	Checksum (Binary)	Checksum (Hex)
36f7	0011 0110 1111 0111	1100 1001 0000 1000	C908
f670	1111 0110 0111 0000	0000 1001 1000 1111	098F
2148	0010 0001 0100 1000	1101 1110 1011 0111	DEB7
8912	1000 1001 0001 0010	0111 0110 1110 1101	76ED
2345	0010 0011 0100 0101	1101 1100 1011 1010	DCBA
7863	0111 1000 0110 0011	1000 0111 1001 1100	879C
0076	0000 0000 0111 0110	1111 1111 1000 1001	FF89

If the first word above is changed to 0x36f6, the binary word will defer by 1 bit, i.e. it is (0011 0110 1111 0110), so the checksum is only different by 1 bit, 1100 1001 0000 1001, and the checksum in hex is C909.

4. [Chapter 3 Written 5] What are the advantages and disadvantages if we make the minimum Ethernet frame larger?

   The major advantage is efficiency. A longer frame length means fewer total overhead in frame header, so effectively more data may be transmitted over the same amount of time.

   There are a few disadvantages though:
   a. Longer delay caused by transmission of longer package;
   b. More bit error because more bits lead to the probability of more erroneous bits;
   c. Higher memory requirement for buffering and handling longer frame.

5. [Chapter 3 Written 7] Should a switch re-compute a new FCS of an incoming frame before it is forwarded?

   Yes.

If the switch is a store-and-forward switch, it will be able to check FCS in time. When the switch detects an inconsistency in FCS, in addition to notify the upstream of the error and perform other error control steps, it should also re-compute the correct FCS and update the FCS field in the frame. In this way, it will not propagate the FCS errors further.

6. [Chapter 3 Written 13] Suppose bit stuffing with 0 is used after 5 consecutive 1's. Assuming the probabilities of 0's and 1's in the bit stream are equal and the occurrences are at random, what is the transmission overhead of the bit stuffing scheme? (Hint: Formulate a recursive formula f(n) to find the expected number of overhead bits in an n-bit string first.)

   Overhead bit is the non-data bit necessary for transmission, in this case the 0 for bit stuffing is the overhead bit.

   In an n-bit string (n>=6), expected number of overhead bits = [Probability of having 5 consecutive 1's]*[number of possible 5 consecutive bits sequence]

   In an 6-bit string (6>=6), expected number of overhead bits: f(6)={(0.5)^5}*2=0.3125*(6-4)
   In an 7-bit string (7>=6), expected number of overhead bits: f(7)={(0.5)^5}*3=0.3125*(7-4)
   …
   In an n-bit string (n>=6), expected number of overhead bits: f(n)=0.3125*(n-4)

   Thus, transmission overhead = f(n)/n = 0.3125*(n-4)/n

7. [Chapter 3 Written 15] In 1000BASE-X, a frame of 64 bytes is first block coded with 8B/10B before transmitting. Suppose the propagation speed is 2 * 10^8. What is the frame "length" in "meters"? (Suppose the cable is 500m long.)

   Number of bytes after 8B/10B coding = 64*(10/8) = 80

   Propagation speed is 2 * 10^8

   Total number of bytes transmitted per unit time = 80 * 2 * 10^8 = 1.6 * 10^10

   1 byte = 8 bits

   Total number of bits transmitted per unit time = 1.6 * 10^10 * 8 = 1.28*10^11

   1000BASE-X is the initial standard for Gigabit Ethernet, so frame "width" = 1/(10^9)

   Total frame "length" = 1.28*10^11 / (10^9) = 128 m

8. [Chapter 3 Written 18] CPU executes instructions at 800 MIPS. Data can be copied 64 bits at a time, with each 64-bit word copied costing six instructions. If an incoming frame has to be copied twice, how much bit rate, at most, of a line can the system handle? (Assume that all instructions run at the full 800-MIPS rate.)

MIPS stands for million instructions per second

Number of bits in an incoming frame, A = 64 bits
Number of instructions (copied twice) in an incoming frame, B = 6*2 = 12
Bit per instruction ratio, A/B = 16/3

Maximum bit rate = 800*10^6 * 16/3 = 4.27*10^9 bits per second

9. [Chapter 3 Written 20] What is the probability that one out of 100 frames of 1000 bytes suffers from an error on average if the bit error rate is 10^−8?

1 byte = 8 bits
1 frame: 1000 bytes = 8000 bits

Probability of having an error in 1 frame = 10^-8 * 8000 = 8*10^-5
Probability of having the erroneous frame out of 100 frames = (1/100) * 8*10^-5 = 8*10^-7