Problem 1.
Consider if potential interleaving encountered in lab1 remains an issue.
As we use the same secrete key for all the client process. There is no way to tell each process apart. Interleaving would still happen for intensive process.

Problem 2.
My ping client only send msg once but the ping command on linux send msg never stop itself.

Myping.c

Received From: 128.10.3.58 Port Number 50001 ,time 270 ms .

LINUX ping:
64 bytes from xinu08.cs.purdue.edu (128.10.3.58): icmp_seq=1 ttl=64 time=0.170 ms
64 bytes from xinu08.cs.purdue.edu (128.10.3.58): icmp_seq=2 ttl=64 time=0.255 ms
64 bytes from xinu08.cs.purdue.edu (128.10.3.58): icmp_seq=3 ttl=64 time=0.236 ms
64 bytes from xinu08.cs.purdue.edu (128.10.3.58): icmp_seq=4 ttl=64 time=0.196 ms

The time interval is much bigger on my program. This could be the overhead of more byte. And the mypingd program.

Bonus
I compile the same tcpclient.c in problem 1 with GCC in both solaris UNIX and MacOS. As the compiler is the same one. The program can be compiled and ran successfully. The behavior of the client is the same as in LINUX.

PROBLEM 1 (20 pts)

1)
If the data frame does not suffer error:
reliable throughput (bps) = data bits / total time
Throughtput = framsize /RTT

If the data frame has error probability of p.
The total time needed to transfer the frame would be RTT(1+p);
So the throughput would be:
Throughtput = framsize /(RTT(1+p))

2)
If the frame can be transmitted after two consecutive trial
The total time spent would be $RTT(1+P+P^2)$
If the frame can be tranmitted after n trial:
The time spend on transmitting the frame would be
Realtime = $RTT(1+P+P^2+P^3+P^4+....+P^n) = RTT(1-P^{n+1})/(1-P) \approx RTT/(1-P)$

The throughput = framsize/Realtime $\approx(1-P)$*framsize/RTT

PROBLEM 2 (20 pts)
1)
(1,2,1,0)
(-2,1,0,0)
(1,2-5,0)
(0,0,0,2)

2) The receiver will dot product its own vector with the vector it received. It would be 0 if its bit is 0 and positive if the bit is positive, negative it the bit is negative.

This is because the msg is a linear combination of the four orthogonal vectors. According to the property of orthogonal vector set. Dot product with any other orthogonal vector will be 0. Dot product with its own will be positive.

The msg vector $z = a1x1 + a2x2 + \cdots + a4x4$
by orthogonality: $z \circ xi = ai(xi \circ xi)$
Thus, positive or negative depending on ai

3)
if the code vectors are not orthogonal (although still independent), the vector will not be able to get the right bit as the dot product property does not hold any more.

4)
when sending bits using amplitude modulation (AM), it can causes spreading. Because similar frequency AM can interference with each other, called inter-channel interference (ICI). For the worse case
signal bandwidths around will overlap with each other.
So the amplitude detected by receiver is distorted.
ICI from spreading and overlap causes weights from the
two signal spectra to be added. It will cause distortion of original weight values and may result in bit flips or outright failure.

5)
Two approaches

1. Put neighboring carrier frequencies far apart. By allocating sufficient spacing between adjacent carrier frequencies, it can directly prevent ICI. But it will waster expensive bandwidth.
2. Orthogonal FDM, use sinusoids that are mutually orthogonal. Transmit multiple bits on *N* orthogonal sinusoids with different frequencies.

PROBLEM 3 (20 pts)

1) The Amplitude Modulated (AM radio) carrier frequencies are in the frequency range 535-1605 kHz. Carrier frequencies of 540 to 1600 kHz are assigned at 10 kHz intervals.
2) Human auditory system is sensitive to 20 Hz–20 kHz. So Any frequency higher than 10 kHz will be missing. Human speech is around 30-3300 Hz. It is enough for talk radio. But because the higher frequency are lost, it is not suit for music.
3) AM radio does not require bidirectional communication. It is only from station to listener. There will be not time calibration like in FDMA. FDMA transmit with frquency change, AM translate with by amplitude modulation.

4) To digitize has two aspects:
• time: the radio with change from continous time to discrete time. A good quality radio require high sampling frequency.
• strength: amplitude is discretized. It will have 8 and 16 bits, which popular and sufficient. It has logarithmic scale due to human sensory sensitivity.

The signal should be able to be translated into wave again. So the AM receiver should be changed so as to decode the time and strength of the radio.

5) Fidelity can only be preserved when analog signal is bandlimited and recipe has finite number of ingredients. It can be only approximately like bandlimited.