CS468 – Homework Assignment 1 – Solution

Please submit your answer with an e-file to blackboard before Friday 2/28/14

1. An organization has a class C network 200.1.1 and wants to form subnets for four departments, which hosts as follows: A – 72 hosts, B – 35 hosts, C – 20 hosts, D – 18 hosts. There are 145 hosts in all.
   (a) Give a possible arrangement of subnet masks to make this possible. What is the network address, subnet mask, broadcast address, maximum number of hosts for each subnet. Please also show the available IP address range for each subnet.
   (b) Suggest what the organization might do if department D grows to 34 hosts.

Sol:
(a) Giving each department a single subnet, the nominal subnet sizes are 2^7,2^6,2^5,2^5 respectively; we obtain these by rounding up to the nearest power of 2. A possible arrangement of subnet numbers is as follows.

200.1.1.0 → 11001000.00000001.00000001.00000000
A: 72 hosts → require **2^7 = 128**, maximum number of hosts **2^7 – 2 = 126**
11001000.00000001.00000001.**0**0000000 → **200.1.1.0/25** (network address)
11111111.11111111.11111111.10000000 → **255.255.255.128/25** (subnet mask)
11001000.00000001.00000001.00000001 → **200.1.1.1** (the 1st host IP in A)
11001000.00000001.00000001.01111110 → **200.1.1.126** (the last host IP in A)
11001000.00000001.00000001.01111111 → **200.1.1.127** (broadcast address)

B: 35 hosts → require **2^6 = 64**, maximum number of hosts **2^6 – 2 = 62**
11001000.00000001.00000001.**10**000000 → **200.1.1.128/26** (network address)
11111111.11111111.11111111.11000000 → **255.255.255.192/26** (subnet mask)
11001000.00000001.00000001.10000001 → **200.1.1.129** (the 1st host IP in B)
11001000.00000001.00000001.10111110 → **200.1.1.190** (the last host IP in B)
11001000.00000001.00000001.10111111 → **200.1.1.191** (broadcast address)

C: 20 hosts → require **2^5 = 32**, maximum number of hosts **2^5 – 2 = 30**
11001000.00000001.00000001.**110**00000 → **200.1.1.192/27** (network address)
11111111.11111111.11111111.11100000 → **255.255.255.224/27** (subnet mask)
11001000.00000001.00000001.11000001 → **200.1.1.193** (the 1st host IP in C)
11001000.00000001.00000001.11011110 → **200.1.1.222** (the last host IP in C)
11001000.00000001.00000001.11011111 → **200.1.1.223** (broadcast address)

D: 18 hosts → require **2^5 = 32**, maximum number of hosts **2^5 – 2 = 30**
11001000.00000001.00000001.**111**00000 → **200.1.1.224/27** (network address)
11111111.11111111.11111111.11100000 → **255.255.255.224/27** (subnet mask)
11001000.00000001.00000001.11100001 → **200.1.1.225** (the 1st host IP in D)
11001000.00000001.00000001.11111110 → **200.1.1.254** (the last host IP in D)
11001000.00000001.00000001.11111111 → **200.1.1.255** (broadcast address)

(b) We have two choices: either assign multiple subnets to single departments, or abandon subnets and buy a bridge. Here is a solution giving A two subnets, of sizes 64 and 32; every other department gets a single subnet of size the next highest power of 2:

A1: 34 hosts → require **2^6 = 64**, maximum number of hosts **2^6 – 2 = 62**
11001000.00000001.00000001.01000000 → **200.1.1.64/26** (network address)
11111111.11111111.11111111.11000000 → **255.255.255.192/26** (subnet mask)
11001000.00000001.00000001.01000001 → **200.1.1.65** (the 1st host IP in A1)
11001000.00000001.00000001.01111110 → **200.1.1.126** (the last host IP in A1)
11001000.00000001.00000001.01111111 → **200.1.1.127** (broadcast address)
A2: 20 hosts → require **2^5 = 32**, maximum number of hosts **2^5 – 2 = 30**
11001000.00000001.00000001.00100000 → **200.1.1.32/27** (network address)
11111111.11111111.11111111.11100000 → **255.255.255.224/27** (subnet mask)
11001000.00000001.00000001.00100001 → **200.1.1.28** (the 1st host IP in A2)
11001000.00000001.00000001.00111110 → **200.1.1.62** (the last host IP in A2)
11001000.00000001.00000001.00111111 → **200.1.1.63** (broadcast address)

B is the same as in (a)

C: 20 hosts (require **2^5 = 32**, maximum number of hosts **2^5 – 2 = 30**
11001000.00000001.00000001.00000000 → **200.1.1.0/27** (network address)
11111111.11111111.11111111.11100000 → **255.255.255.224/27** (subnet mask)
11001000.00000001.00000001.00000001 → **200.1.1.1** (the 1st host IP in C)
11001000.00000001.00000001.00011110 → **200.1.1.30** (the last host IP in C)
11001000.00000001.00000001.00011111 → **200.1.1.31** (broadcast address)

D: 34 hosts → require **2^6 = 64**, maximum number of hosts **2^6 – 2 = 62**
11001000.00000001.00000001.11000000 → **200.1.1.192/26** (network address)
11111111.11111111.11111111.11000000 → **255.255.255.192/26** (subnet mask)
11001000.00000001.00000001.11000001 → **200.1.1.192** (the 1st host IP in D)
11001000.00000001.00000001.11111110 → **200.1.1.254** (the last host IP in D)
11001000.00000001.00000001.11111111 → **200.1.1.255** (broadcast address)

2. Calculate the effective throughput to transfer a 1,000KB file in the following case, assuming a round-trip time of 100ms, a packet size of 1KB data, and an initial 2*RTT of "handshaking" before data is sent.
   (a) The bandwidth is 1.5 Mbps, and data packets can be sent continuously.
   (b) The bandwidth is 1.5 Mbps, but after we finish sending each data packet, we must wait one RTT before sending the next.

Sol:
We will count the transfer as completed when the last data bit arrives at its destination. Analternative interpretation would be to count until the last ACK arrives back at the sender, in which case the time would be half an RTT (50ms) longer.

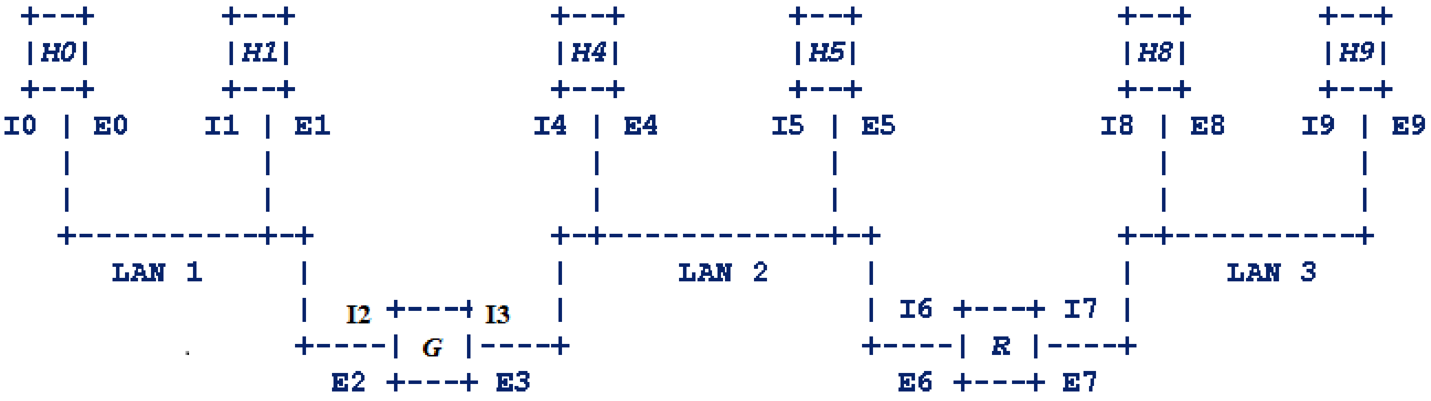
(a) 2 initial RTT's (200ms) + 1000KB/1.5Mbps (transmit) + RTT/2 (ACK reply)
≈ 0.25 + 8Mbit/1.5Mbps = 0.25+5.33 sec = 5.58 sec.

Throughput = 8Mbit/5.58sec = 1.43Mbps

If we pay more careful attention to when a mega is 10^6 versus 2^{20},
we get 8,192,000 bits/1,500,000 bits/sec = 5.46sec, for a total delay of 5.71 sec.
Throughput = 8Mbit/5.71sec = 1.4Mbps

(b) To the above we add the time for 999 RTTs (the number of RTTs between when packet1 arrives and packet 1000 arrives), for a total of 5.58 + 99.9 = 105.48 sec
Throughput = 8Mbit/105.48sec ≈ 75Kbps
or
5.71 + 99.9 = 105.61sec
Throughput = 8000Kbit/105.61sec ≈ 75.75Kbps

3. Consider the network topology below:



- *G* is a gateway**.**
- *R* is an *IP* router.
- *H0* , *H1* , *H4* , *H5* , *H8* & *H9* are hosts.
- I0 ~ I9 are 32-bit *IP* addresses, as shown.
- E0 ~ E9 are 48-bit *Ethernet* MAC addresses, as shown.

Suppose host *H4* sends an *IP* packet to host *H9*. This packet will, of course, be encapsulated in an *Ethernet* frame.

(a) What are source and destination *Ethernet* addresses in the Ethernet header of the frame when it traverses on LAN 2?

(b) What are source and destination *IP* addresses in the IP header of the packet when it traverses on LAN 2?

(c) What are source and destination *Ethernet* address in the Ethernet header of the frame when it traverses on LAN 3?

(d) What are source and destination *IP* address in the IP header of the encapsulated packet when it traverses on LAN 3?

Suppose host *H1* wants to talk to *H9* and *H1* has never connected to *H9* before.

(e) What *H1* will do and what protocol will be used?

(f) How many machines will get this message and who they are?

(g) What address information (source/destination MAC and source/destination IP) in the protocol header of the packet when it traverses on LAN 1?

(h) What address information (source/destination MAC and source/destination IP) in the protocol header of the packet when it traverses on LAN 2?

(i) What address information (source/destination MAC and source/destination IP) in the protocol header of the packet when it traverses on LAN 3?

(j) What *H9* will do when it receive the packet? What message it will reply and to whom? Please describe the details.

Sol:

(a) Source MAC address: E4, Destination MAC address: E6

(b) Source IP address: I4, Destination IP address: I9

(c) Source MAC address: E7, Destination MAC address: E9

(d) Source IP address: I4, Destination IP address: I9

(e) H1 will check it's ARP cache to find the MAC address for I9. Of course, MAC address for I9 is not there. H1 checks the APR cache again to find the MAC address for its default gateway. If the MAC address of the gateway is not here, H1 then broadcasts an ARP request packet to LAN1.
(More detail)
After H1 gets an ARP reply with the MAC address of the gateway (E2), H1 will send out the IP data packet to the gateway with the destination address I9.

(f) H0 and Gateway on LAN1 will receive the ARP request packet from H1.
(More detail)
Only H1 will receive the ARP reply packet from the gateway, and only the gateway will receive the IP data packet from H1 on LAN1.

(g) The address information in the ARP request packet from H1 is
Src MAC: E1, Dst MAC: all FF, Src IP: I1, Dst IP: I2
(More detail)
After the gateway gets the ARP request packet from H1, it replies with an ARP reply packet to H1. The address information in the ARP reply from the gateway to H1 is Src MAC: E2, Dst MAC: E1, Src IP: I2, Dst IP: I1
After H1 gets the ARP reply packet from the gateway, it replies with an IP data packet to the gateway. The address information in the IP data packet from H1 to the gateway is Src MAC: E1, Dst MAC: E2, Src IP: I1, Dst IP: I9

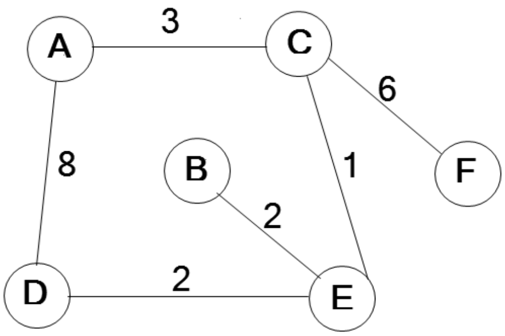
(h) After gateway gets the IP data packet from H1, it will check its ARP cache and sends an ARP request packet to LAN2 for the MAC address of the next hop with Src MAC: E3, Dst MAC: all FF, Src IP: I3, Dst IP: I6
(More detail)
After gateway gets an ARP reply with the MAC address of the Router (E6), the gateway will forward the IP data packet to the Router with the destination address of I9.

(i) After Router gets the IP data packet from the gateway, it will check its ARP cache and sends an ARP request packet to LAN3 for the MAC address of the destination with Src MAC: E7, Dst MAC: all FF, Src IP: I7, Dst IP: I9

(j) After H9 receives the ARP request packet from the router, it sends an ARP reply packet to the router with Src MAC: E9, Dst MAC: E7, Src IP: I9, Dst IP: I7, and

saves the router’s MAC and IP addresses in its ARP cache.
(More detail)
After router gets the ARP reply from H9, it forwards the IP data packet to H9.

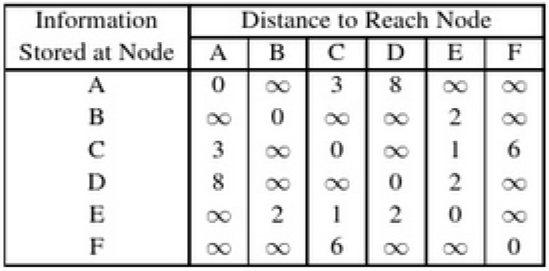
4. For the network given below, give global distance-vector tables when



(a) Each node knows only the distances to its immediate neighbors.
(b) Each node has reported the information it had in the preceding step to its immediate neighbors.
(c) Apply (b) again.

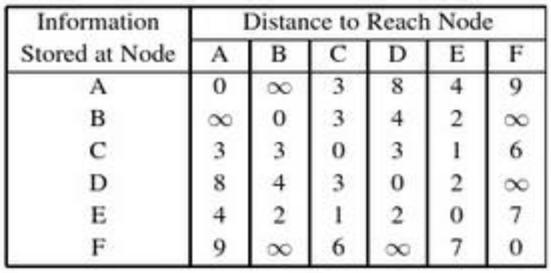
Sol:

(a)



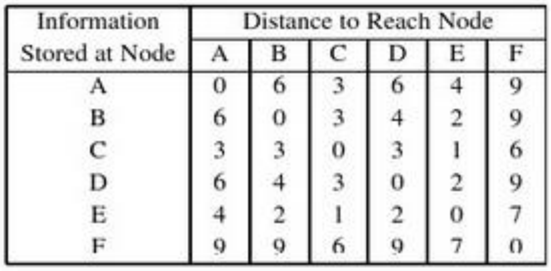
Information	Distance to Reach Node					
Stored at Node	A	B	C	D	E	F
A	0	∞	3	8	∞	∞
B	∞	0	∞	∞	2	∞
C	3	∞	0	∞	1	6
D	8	∞	∞	0	2	∞
E	∞	2	1	2	0	∞
F	∞	∞	6	∞	∞	0

(b)



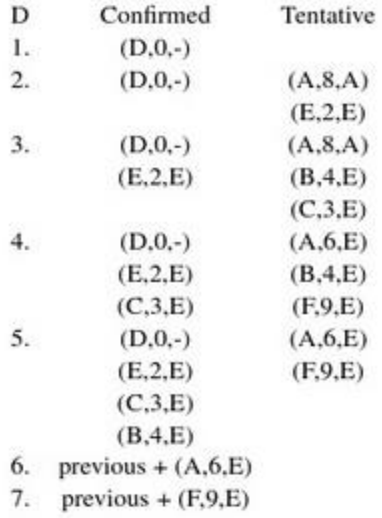
Information	Distance to Reach Node					
Stored at Node	A	B	C	D	E	F
A	0	∞	3	8	4	9
B	∞	0	3	4	2	∞
C	3	3	0	3	1	6
D	8	4	3	0	2	∞
E	4	2	1	2	0	7
F	9	∞	6	∞	7	0

(c)



Information	Distance to Reach Node					
Stored at Node	A	B	C	D	E	F
A	0	6	3	6	4	9
B	6	0	3	4	2	9
C	3	3	0	3	1	6
D	6	4	3	0	2	9
E	4	2	1	2	0	7
F	9	9	6	9	7	0

5. For the network given in question 4, show how the link-state algorithm builds the routing table for node D.
   Sol:



D	Confirmed	Tentative
1.	(D,0,-)	
2.	(D,0,-)	(A,8,A) (E,2,E)
3.	(D,0,-) (E,2,E)	(A,8,A) (B,4,E) (C,3,E)
4.	(D,0,-) (E,2,E) (C,3,E)	(A,6,E) (B,4,E) (F,9,E)
5.	(D,0,-) (E,2,E) (C,3,E) (B,4,E)	(A,6,E) (F,9,E)
6.	previous + (A,6,E)	
7.	previous + (F,9,E)	

6. Consider three common network commands: ping, traceroute and nslookup,
   (a) Give out when you want to use these three commands, what information you might get from these commands, and how they work. Please give your answer based on captured packets using Wireshark for each command.
   (b) I cannot access a remote machine (The machine's name is "boy", actually it is timeout when I use the command "ping boy"). Then you might derive the conclusions that the problems are 1) the name server is down, 2) the intermediate nodes is down, 3) the remote machine "boy" is down. Give your investigations that support your conclusion, specify clearly what command you use and what possible results that make you derive the conclusion.

Sol:

(a) Ping gives the information about whether a host address is reachable from your current network or not;

Traceroute gives information about the route from your current location to a host. It also measures the delay of packets;

Nslookup queries the DNS system to get a domain name or IP address mapping records.

(Require Wirshark screenshot to show these 3 protocols)

(b) 1) Use "nslookup boy" for boy to see if a correct IP address can be resolved from the name server. If not, then the result could be "connection timed out; no servers could be reached".

2) Use "traceroute boy" for boy to see if the current machine is able to reach the remote host or not. If not able to reach the host, it will print out some "*" to indicate the probe timed out.

3) Use "ping boy" for boy to see if the remote host can be reached or not. If not, the result would be "Destination Host Unreachable".

7. Mixed questions:
   (a) Give three similarities of Ethernet, fast Ethernet and Gigabit Ethernet, and three differences among them.
   (b) Why we need the hardware address, and why we need the IP address. Give three usages of these two addresses.
   (c) What ARP stands for? When I use ping, traceroute and nslookup commands, do these commands will invoke ARP? When they will and when they will not?
   (d) Can you use PING command only to simulate the function of TRACEROUTE? How? Please give a real example in your virtual machines.
   (e) Why you need DNS? If you cannot connect to any DNS server, what you need to know and to do to connect to remote web server?

Sol:

(a) Similarities:
1). They provide services up to and including data link layer.
2). They support bus topology which is the network topology for LAN.
3). They adopt twisted-pair type cable.
4). They will create similar Ethernet frame format.
5). They support CSMA/CD
Difference:
1). Speed: Ethernet (10Mbps), Fast Ethernet (100Mbps), Gigabit (1Gbps)
2). Media: Ethernet (10Base-T),Fast Ethernet (100Base-T), Gigabit (1000Base-T)
3). Standard: Ethernet (802.3), Fast Ethernet (802.3u), Gigabit (802.3z)

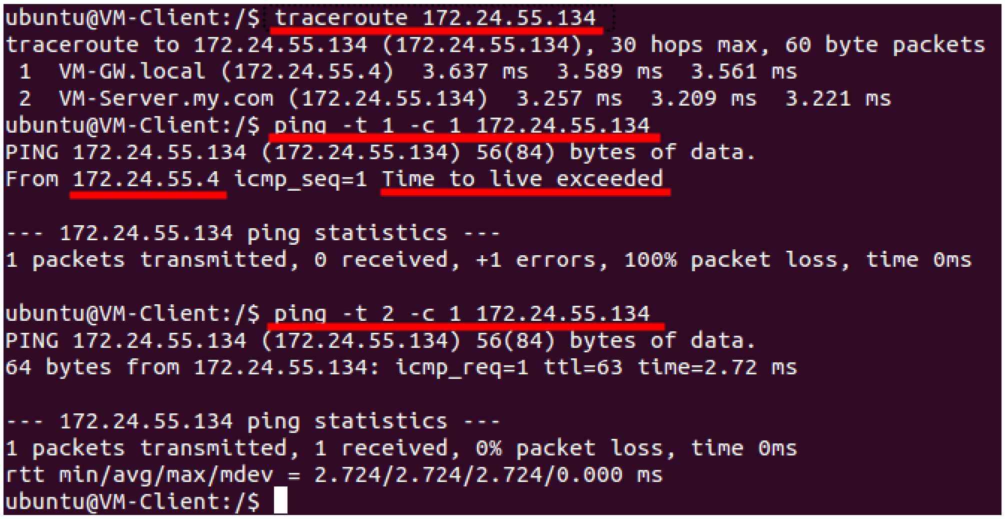
(b) Hardware address uniquely identifies a network adapter. It allows computers to uniquely identify themselves at a relatively low level (Layer 2 in the OSI model). Usages of hardware address: (1) at the link layer, the MAC address is used for communication within the same local network; (2) layer 2 switches use MAC addresses to restrict packet transmission to intended recipient; (3) Used for ARP protocol.

The IP address is used to globally identify a host in the Internet. It is used for global addressing. Usages of IP addresses are: (1) IP addressing; (2) Routing; (3) setting up firewall rules.

(c) ARP stands for Address Resolution Protocol. When no ARP record for an IP address can be found at the host's cached ARP table, the host needs to send out ARP request packets to revolve the MAC address.
In the case of ping and traceroute, the IP address can be the destination's IP or the gateway's IP depending on whether the destination resides in the same local network as the host or not.
In the case of nslookup, the IP address can be the DNS server's IP or the gateway's IP.

(d) Yes. For example, If I issue "traceroute 172.24.55.134" from VM-Client to VM-Server, the output shows that it pass through 1 hop VM-GW (172.24.24.55.4) to the destination VM-Server (172.24.55.134).

To get the similar output to the traceroute, “Ping –t 1 172.24.55.134” can be used to get the 1 hop (172.24.55.4) and Time to live exceeded error. With this error, it means that the ping packet didn’t reach to the destination yet. Then, the 2nd Ping command with ttl =2 can be issued: “Ping –t 2 172.24.55.134”. The output of the 2nd ping command return ICMP echo reply, that means the 2nd ping packet reach to the destination.



```
ubuntu@VM-Client:/$ traceroute 172.24.55.134
traceroute to 172.24.55.134 (172.24.55.134), 30 hops max, 60 byte packets
 1  VM-GW.local (172.24.55.4)  3.637 ms  3.589 ms  3.561 ms
 2  VM-Server.my.com (172.24.55.134)  3.257 ms  3.209 ms  3.221 ms
ubuntu@VM-Client:/$ ping -t 1 -c 1 172.24.55.134
PING 172.24.55.134 (172.24.55.134) 56(84) bytes of data.
From 172.24.55.4 icmp_seq=1 Time to live exceeded

--- 172.24.55.134 ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

ubuntu@VM-Client:/$ ping -t 2 -c 1 172.24.55.134
PING 172.24.55.134 (172.24.55.134) 56(84) bytes of data.
64 bytes from 172.24.55.134: icmp_req=1 ttl=63 time=2.72 ms

--- 172.24.55.134 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.724/2.724/2.724/0.000 ms
ubuntu@VM-Client:/$
```

(e) DNS is mainly used to translate human-friendly domain names to machine-friendly IP addresses. If you cannot connect to any DNS server, you need to know the IP address of the web server you want to visit and use the IP address instead of the domain name in your browser to access the web service.

8. Please decrypt the following ciphertext to plaintext, describe your approach in detail.

```
VA PELCGBTENCUL N PNRFNE PVCURE NYFB XABJA NF PNRFNEF
PVCURE GUR FUVSG PVCURE PNRFNEF PBQR BE PNRFNE FUVSG VF BAR
BS GUR FVZCYRFG NAQ ZBFG JVQRYL XABJA RAPELCGVBA GRPUAVDHRF
VG VF N GLCR BS FHOFGVGHGVBA PVCURE VA JUVPU RNPU YRGGRE VA
GUR CYNVAGRKG VF ERCYNPRQ OL N YRGGRE FBZR SVKRQ AHZORE BS
CBFVGVBAF QBJA GUR NYCUNORG GUR RAPELCGVBA FGRC CRESBEZRQ
OL N PNRFNE PVCURE VF BSGRA VAPBECBENGRQ NF CNEG BS ZBER
PBZCYRK FPURZRF FHPU NF GUR IVTRARER PVCURE NAQ FGVYY UNF
ZBQREA NCCYVPNGVBA VA GUR EBG13 FLFGRZ NF JVGU NYY FVATYR
NYCUNORG FHOFGVGHGVBA PVCUREF GUR PNRFNE PVCURE VF RNFVYL
OEBXRA NAQ VA ZBQREA CENPGVPR BSSREF RFFRAGVNYYL AB
PBZZHAVPNGVBA FRPHEVGL
```

Sol:
Count frequency of each letter:
A:36, B:35, C:27, D:1, E:37, F:46, G:46, H:9, I:1, J:6, K:3, L:12, M:0, N:42, O:8, P:32, Q:12, **R:67**, S:11, T:3, U:29, V:48, W:0, X:3, Y:19, Z:13

R is the most frequently used in the ciphertext, so let’s try R (17) →E (4)
Let’s choose the decryption function of Caeser cipher:

$$D(x)=(x-k) \bmod 26 \Rightarrow 4=(17-k) \bmod 26 \Rightarrow k = 13$$

After decrypting the first couple of ciphertexts with Caeser cipher and k=13, we found that there are some meaningful sentences. Therefore, we make sure the ciphertext is encrypted with Caeser cipher with k=13.

Plantext:

```
IN CRYPTOGRAPHY A CAESAR CIPHER ALSO KNOWN AS CAESARS
CIPHER THE SHIFT CIPHER CAESARS CODE OR CAESAR SHIFT IS ONE
OF THE SIMPLEST AND MOST WIDELY KNOWN ENCRYPTION TECHNIQUES
IT IS A TYPE OF SUBSTITUTION CIPHER IN WHICH EACH LETTER IN
THE PLAINTEXT IS REPLACED BY A LETTER SOME FIXED NUMBER OF
POSITIONS DOWN THE ALPHABET THE ENCRYPTION STEP PERFORMED
BY A CAESAR CIPHER IS OFTEN INCORPORATED AS PART OF MORE
COMPLEX SCHEMES SUCH AS THE VIGENERE CIPHER AND STILL HAS
MODERN APPLICATION IN THE ROT13 SYSTEM AS WITH ALL SINGLE
ALPHABET SUBSTITUTION CIPHERS THE CAESAR CIPHER IS EASILY
BROKEN AND IN MODERN PRACTICE OFFERS ESSENTIALLY NO
COMMUNICATION SECURITY
```