

1. /26 implica 2^6 subnets = 64, pc1 e pc2 não comunicam porque o pc2 tem o address errado ( da network inferior). A comunicação de PC1 será enviada para R1 onde será redirecionada para a network inferior na forma de um arp request, este não obterá resposta logo PC1 não irá aprender o port associado a PC2.
2. X : 192.168.2.1 -> 192.168.2.62 exceto os do PC1 e PC2

Y: 192.168.2.65 -> 192.168.2.126 exceto os do PC3

1. e0 & e1-> mac do PC1

e2 -> mac de f0/0 de R1

1. mais users -> mais trafico -> mais colisões -> mais lento. Implementar um sistema request to send/ clear to send
2. reduce the leasing time, apos saída dos users não há renewal, criando espaço livre mais cedo para os próximos.
3. Sim, porque é para fora. Table: private addresses and ports associados à pool de adresses públicos
4. 2 partes: Uma através do mac address e a outra através do router
5. Low timeout faz com que dê expire antes do ack chegar ao data sender.
6. O udp precisa de 40% logo o tcp não consegue passar de 60%
7. Iterativo, apesar de perder tempo em round trip times, como está perto do server continua a ser mais rápido que o recursivo que iria percorrer todos os servers

11)Sim porque têm addresses/ports diferentes atribuídos pelo nat/pat

12)