**Part 2. Textbook questions**

**Chapter 7 [25 points]**

7.1 What are the services that can be provided using IEEE 802.11?

The emergence of the IEEE 802.11 standard defined two categories for sending MAC service data units between two entities on a network. These are recognized in two categories of station services and distribution system services. Service station services include wireless elements on the network such as a handheld device or a smart phone. These stations provide services such as authentication, deauthentication, MAC service data unit delivery, and privacy such as encryption of traffic. Distribution system services provides services to outside your network. A wireless access point would use distribution system services. This would include services such as association, disassociation, and reassociation between access points and stations. Distribution is another service provided in the distribution system service, it allows for a MAC frames to be sent across a system. Another service provided is integration. The integration service allows for communication between 802.11 and non 802.11 standards. So in conclusion IEEE 802.11 services include services for within your network as station services, and services for outside your network as distribution system services.

7.2 What are the differences between the infrastructure mode and the ad-hoc mode in wireless networks?

There are many differences between ad-hoc and infrastructure networks. To begin, ad-hoc allows for each device on a network to communicate directly with each other. All devices can talk to each other and there is not one central entity controlling the communication. They can only communicate with other ad-hoc devices. Typically this has made security less sophisticared in contrast to infrastructure mode networks. Infrastructure mode allows for the use of an access point. The access point is what controls data and communication. This has some potentially added benefits such as the possibility for faster speeds, integration with other networks, and stronger security. So basically ad-hoc it peer to peer while infrastructure has a central entity.

7.3 Why are acknowledgements used in 802.11 but not in a wired Ethernet?

Acknowledgments are used in 802.11 standards because they help convey and protect against packet loss. In a traditional wired Ethernet, packets are dispatched properly without any packet loss or collision. The use of an ACK in 802.11 wireless networks allows for the detection of dropped packets and collisions. If the sender receives an acknowledgment then they know their packet has been delivered correctly. In wired Ethernet they have collision detection through CSMA/CA so the acknowledgment is not needed. That is why acknowledgments are used in 802.11 and not in wired Ethernet.

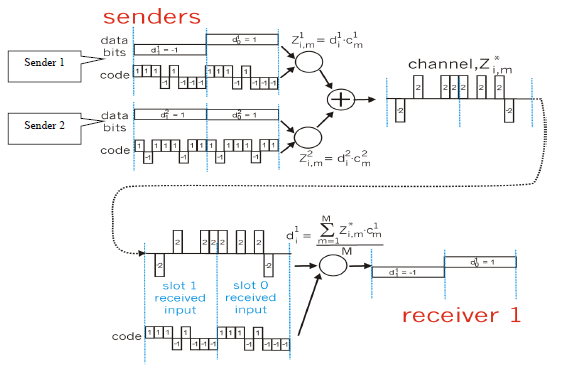
7.4 Suppose the IEEE 802.11 RTS and CTS frames were as long as the standard DATA and ACK frames. Would there be any advantages to using the CTS and RTS frames? Why or why not?

There would not be any advantages for using CTS and RTS frames is they were as long as standard DATA frames. If two devices transmit data at the same time using RTS/CTS and the RTS frame is as long as a DATA frame then the channel will be wasted as if it would have been for two colliding DATA frames. In this instance an ACK will be waiting for a NAK that will never come. This is why using RTS/CTS is only useful when frames are smaller than the DATA frame.

7.5 What is the difference between a permanent address and a care-of address? Who assigns a care-of address?

For mobile devices there are two types of addresses permanent and care-of addresses. Your permanent address is given to mobile device and this is where your mobile device lives on its home network. A care-of-address is an address is gets when visiting a foreign network. When a device is roaming outside of its home network the permanent address will receive data and then forward on to the other network via the care-of address. The care-of address is given to the device by the foreign network via a foreign agent.

7.6 The following picture is a copy of a Figure in the textbook. It shows a CDMA example that supports two senders. Suppose that the receiver wanted to receive the data being sent by sender 2. Show (by calculation) that the receiver is indeed able to recover sender 2’s data from the aggregate channel signal by using sender 2’s code.\



d21= (((-2\*-1) + (2\*1) + (2\*1) +(2\*1)) / 8) = 1

d22= (((2\*1) + (2\*1) +(2\*1) + (-2\*-1)) / 8) = 1

Because the calculations above both equal one, this proves that the receiver is able to recover sender number 2’s data from the aggregate channel by using sender number 2’s code.

**Chapter 8 [25 points]**

8.1 What is the most important difference between a symmetric key system and a public key system?

Although symmetric and public key systems both use keys, they do use them in a different fashion. To begin, in a symmetric key system both the sender and receiver know the same secret key to decipher an algorithm. That is the most important difference between the two. In a public key system there is a public key and private key. The sender can encrypt the message with the receiver’s public key, then when the message is received the receiver decrypts the message with their own private key. This is asymmetric encryption and attempts to block the security vulnerability if one of the keys were compromised. In symmetric encryption if you get the private key from one side you can use it on the other side. This is not the case with public key encryption because the private key is different than the public key.

8.2 In what way does a hash function provide a better message integrity check than a checksum (such as Internet Checksum)?

Although a hash function and a checksum both return values that cannot be reversed a hash function provides better message integrity. This because internet checksums are designed to detect common errors quickly and not for message integrity. Internet checksums can have collisions, meaning that two answers can equal the same outcome from the checksum. A good hashing algorithm aims to eliminate collisions where every input will have its own output. So because hash functions are designed to not have collisions they are better for message integrity.

8.3 Can you “decrypt” a hash of a message to get the original message? Explain.

No you cannot decrypt a hash message to get its original message. This is because a hashing function is a one way function that can be reversed. What you can do to get the original message is compare your messages hash with the hash of the original message. If the two hashes match then they are the same message. This can be taken advantage of by brute forcing all messages or a dictionary attack. To mitigate this salt was added to hash functions so that two inputs of the same value don’t have the same hashed values outputted, protecting against the dictionary attack.

8.4 Suppose that Bob receives a PGP message from Alice. How does Bob know for sure that Alice created the message?

Because PGP uses private and public keys Bob can verify he received the message from Alice. This is because Alice can digitally sign message with her private key. When Bob receives the message he can verify the message by using Alice’s public key. This is because only Alice’s public key can verify the digital signature is from Alice’s private key. That is how Bob is able to verify the message is from Alice via her private and public keys.

8.5 Consider WEP for 802.11. Suppose that the data is 10010100 and the keystream is 10110010. What is the resulting ciphertext?

10010100

+10110010

=00100110

Resulting cipher text = 00100110

8.6. Using the monoalphabetic cipher in the textbook, encode the message “This is a server.” Decode the message “rmij’u uamu xyj.”

Plain text = abcdefghijklmnopqrstuvwxyz

Cipher in book page 598 = mnbvcxzasdfghjklpoiuytrewq

“This is a server” = “uasi si m icotco”

“Rmij’u uamu xyj” = “wasn’t that fun”

8.7 Consider the RSA algorithm with *p=5* and *q=17*.

a. What are *n* and *z*?

n = (5\*17) = 85

z = (4\*16) = 64

b. Let *e* be 7. Is this an acceptable choice? Why? If not, can you suggest another option?

Yes, this is an acceptable choice because it is less than 85 and shares no other common factors besides 1 with 64.

c. Based on your answer for part b), find *d* such that *de*=1 (mod *z*) and *d*<85.

D= 65 / 7 == **9.28571429**

(9.28571429 \* 7)= 65,

65 mod 64 = 1

**Chapter 9 [25 points]**

9.1 Generalize the basic approaches we used for making the best out of best effort service for real-time interactive multimedia applications.

General basic approaches used for making best effort service for real-time interactive media applications go as follows. To begin, removal of Jitter is preformed using timestamps, playout delays, and sequence numbers. Lost recovery schemes should be used such as interleaving and forward error correction. It is best practice to bring content closer to clients by using content distribution networks. Another good practice is to use UDP when possible for more real time audio / video and negates against TCP congestion. Another good practice is to have the client side bandwidth to match the server side bandwidth. Keeping in mind bandwidth provisioning and network dimensioning to help accomplish this. These are some best practices for real-time multimedia applications.

9.2 There are two types of redundancy in video. Describe them and discuss how they can be exploited for efficient compression.

There are two types of video redundancy, spatial and temporal. These both can be exploited through video compression. Spatial redundancy is the redundancy within and image. For instance an image with many white pixel spaces can be compressed and the white pixels be reused without giving up image quality. Temporal redundancy compares two images as a whole, if the two images are the same then the compression can encode both images using only one of the images while still referencing both of them. Effectively eliminating the need to encode the second redundant image.

9.3 Assume an Internet phone application generates packets only during talk spurts. During a talk spurt the sender generates bytes at a rate of 1000 bytes per second, and every 50 msecs the sender gathers bytes into chunks. Assume that RTP is used that will add a header to each chunk. In addition UDP and IP will be used. Suppose all headers (including RTP, UDP and IP) have a total length of h and an IP datagram is emitted every 40 msecs. Find the transmission rate in bits per second for the datagram generated by one side of the application.

1000\*.05 + h == (50 + h)

So that 50 +h bytes are sent every 50 msec.

The transmission rate is

(50+h) \*8\* 1000/50 bps =(400 + 8h)\*1000/50 = (400,000 +8000h)/50 bps

== **(8000+ 160h) bps**

9.4 Consider the procedure described in “Adaptive Playout Delay” for estimating average delay *di*. Let be the most recent sample delay, let be the next most recent sample delay, and so on. For a given audio application, suppose three packets have arrived at the receiver with sample delays , , and . Express the estimate of delay *d* in terms of *u* and the three samples. 3 3 t r − 2 2 t r − 1 1 t r − 2 2 t r − 3 3 t r –

n-1

d(n) = u ∑ (1-u)j-1(rj – tj) + (1-u)n-1(rn– tn)

j=1

d(1) = (1-u)1-1(r3 – t3) = (r3 – t3)

2-1

d(2) = u ∑ (1-u)j-1(rj – tj) + (1-u)2-1(r3– t3) = (1-u) (r2 – t2) + (1-u)(r3– t3)

j=1

d(3) = u [ (1-u)(r2 – t2) + (1-u)2(r3– t3)]

9.5 This chapter describes several FEC schemes. Briefly summarize them. Both schemes increase the transmission rate of the stream by adding overhead. Does interleaving also increase the transmission rate?

In both forward error correction schemes, they add overhead increasing transmission rate. This is done in the first scheme by creating a duplicate chunk after every N number of chunks. That way if the first chunk were to fail one could recover the data from the redundant chunk. This is done by using an exclusive OR with the N original chunks. The second scheme uses a duplicate low resolution scheme with a low bit rate with the original. This can be used as fall back if packets are lost. Interleaving does not increase the transmission rate, this is because it does not add any overhead but only rearranges the data units.

9.6 Compare the procedure described in “Adaptive Playout Delay” for estimating average delay with the procedure in Chapter 3 (“Estimating the Round-Trip Time”) for estimating round-trip time. What do the procedures have in common? How are they different?

Average delay and round-trip time have similar features in common in the fact that the both place weighted values on both their sample size and their estimated pieces of data. For round-trip time this is expressed when the estimated round-trip time is multiplied by a larger number and the sample size is multiplied by a smaller number and then both are added together to get the new estimated round-trip time. An example of this would be newRoundTripTime = (estimatedRoundTripTime \* 7/8) + (sampleRoundTripTime \* 1/8). Taking note of how the weighted values multiplicities add up to one. In the estimating average delay the weighting is opposite. The higher weight is put on the sample newer delays while a lower weight is put on the delays of the past. This is opposite of round-trip time putting higher weight on the past values and lower weight on the newer values. Having weighted values for past and present values makes these procedures have something in common although they are fundamentally backwards from each other’s logic.

9.7 Is it possible for a CDN to provide worse performance to a host requesting a multimedia object than if the host has requested the object directly from the distant origin server? Please explain.

Yes it is possible for a CDN to provide worse performance than from an original distant server. To begin if the CDN is acting up that entity has to wait on the CDN provider to fix the issue and can’t be solved in house where the in house server can be fixed onsite immediately when it’s down. So if the CDN provider has not fixed the issue then the onsite server would be better. Next if the CDN is outside of your geographic region it might be better to go with the in house server. Since CDN’s are developed with a target audience in mind, if you are not that target audience then you are not designed for optimal viewing of content.

9.8 What is the difference between end-to-end delay and packet jitter? What are the causes of packet jitter?

End-to-end delay effects packet jitter. To begin end-to-end delay is the time it takes a packet to make a round trip from source to destination. When that end-to-end delay is different between packets that is packet jitter. This packet jitter can be caused by network congestion slowing down how fast packets are sent out. It can also be caused by change of paths where the new path takes more hops and a longer time to complete. Queuing can cause a packet to be sent out at different intervals also creating jitter. Really any reason to slow down or speed up the transmission of the next packets end-to-end delay will cause jitter.

9.9 Summarize how the token buckets and WFQs can be used together to provide policing mechanisms.

Token buckets be used together to create a type of traffic policing called the leaky bucket policing. This is used to make sure traffic meets a certain criteria, such as voice traffic not being faster than 1mbps. When adding weighted fair queuing to the leaky bucket you can provide a provable maximum delay in a queue. Because packets are coming into the queue and coming out of the queue at different finite speeds one can calculate the max delay possible in the weighted fair queue. This allows for a better estimation of allowed delayed time in the policy mechanism.