Lecture Notes on Wifi

Goals:

(1) Discuss general architecture of wifi networks

(2) Discuss MAC protocols associated with wifi networks

→ CSMA/CA

↳ Need because of the hidden node problem

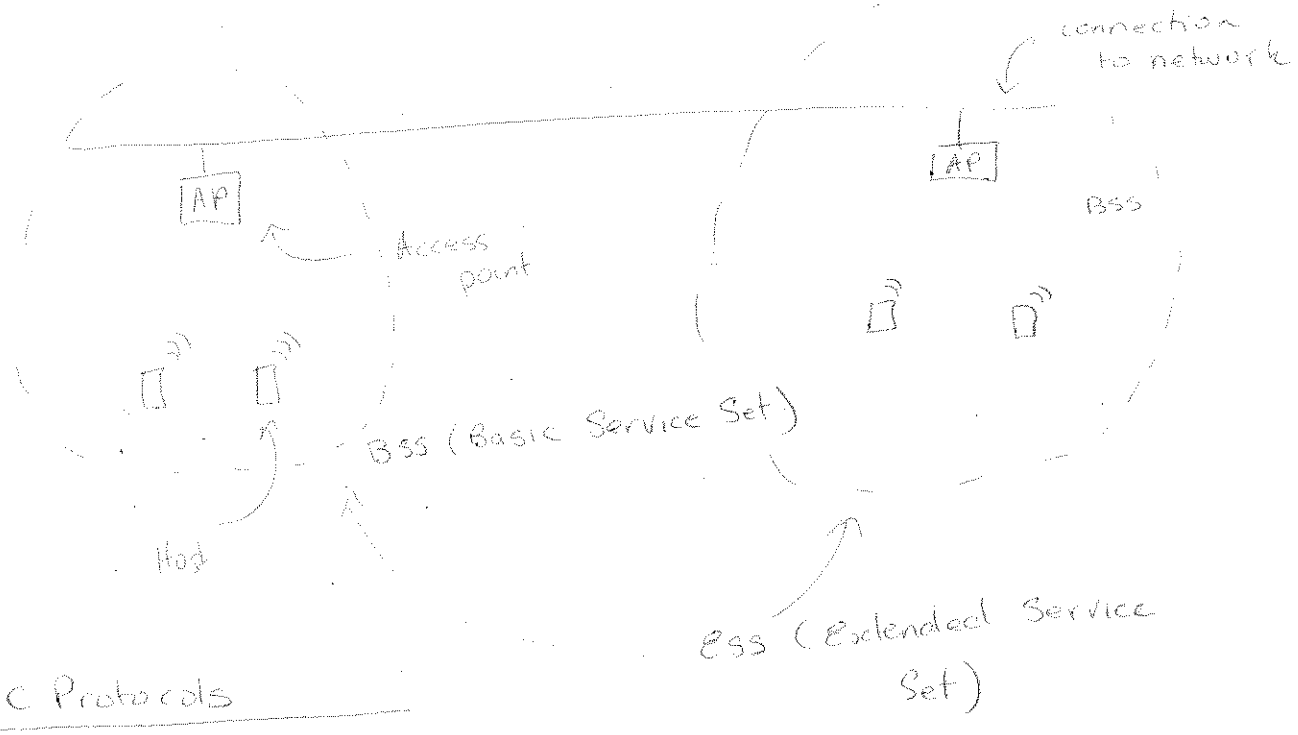
→ Discuss fragmentation (Motivation & Issues)

(3) Discuss how node connect / join a wifi network.

(4) Discuss authentication & Encryption of Wep network

* ~~discuss~~ Include discussion of attacks.

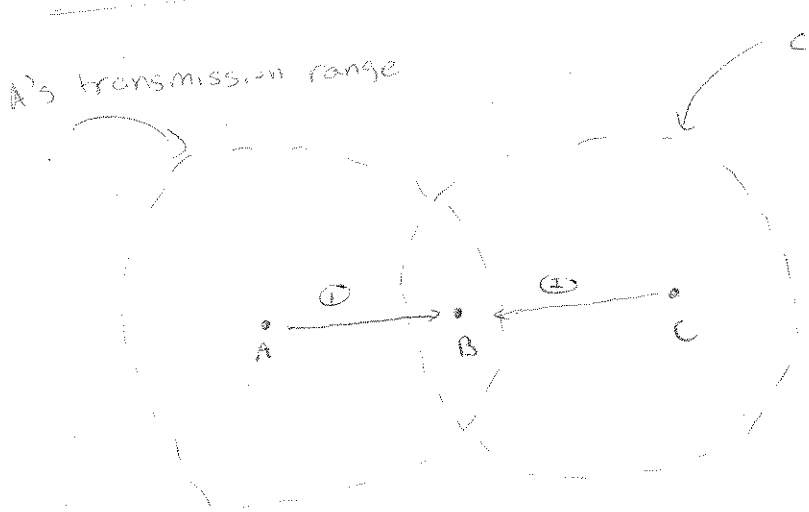
Architecture & Vocabulary



MAC Protocols

We can't use the same MAC protocols we used in wire networks. Why?

- Because of the hidden node problem.

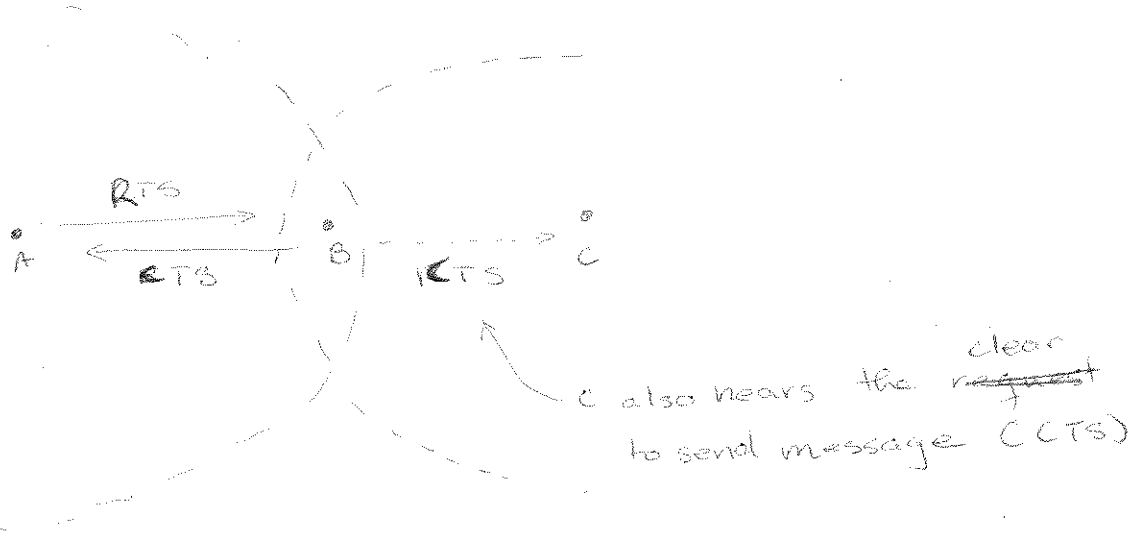


① A transmits because it can't hear C's transmission

② C transmits because it can't hear A's transmission

This results in a collision.

How do we deal with the hidden node problem CSMA/CA



- The request to send (RTS) message from A is heard by B but not by C

However B's CTS is heard by both C & A

Stations record the nodes associated with RTS & CTS in a Network allocation vector (NAV)

So C NAVs:

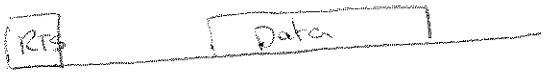
C's NAV (RTS) C's NAV (CTS)

[ ] [A]

empty

example transmission

Scr



Ack lets the transmitter know that the packet wasn't lost due to a collision

DST

Fragmentation

(1) Data isn't transmitted in an continous chuck instead it is fragmented. Why?

- wifi radio networks are more prone to errors
  - → Higher BER (Bit Error Rate) so if we make the packets smaller we can detect these errors.
  - → ~~Collision~~ ~~Smaller packets all~~.

Fragmentation is done at the MAC protocol level so the Data link layer is not aware of the change. (This means that the wifi MAC does this the assembly and disassembly of the fragments)

> Recall fragmentation causes issues remember IPV4

① Stop and wait is used for each fragment

② Possible to inject fragments

See www.fragattacks.com

Now that we seen are architecture and the MAC protocol. What type of encoding does Wifi use? Wifi uses the OFDM encoding scheme. [See wide linked on course site]

Data link
Physical ←

How do we Join a WiFi Network?

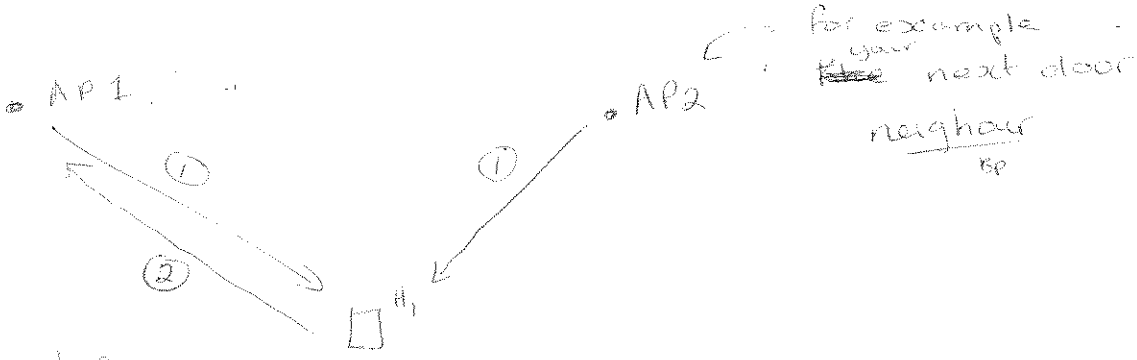
① first you need to scan and Identify the availble networks.

② Then need associate and authenicate with the network

① Scanning

There are two types of scanning
→ passive
→ active

Passive scan



① transmit Beacon frame contain SSID ← identifying wifi

② Association Request frame

③ Association Response frame.

Active Scanning

Search for an AP given the SSID

① Probe Request.

② Probe Response

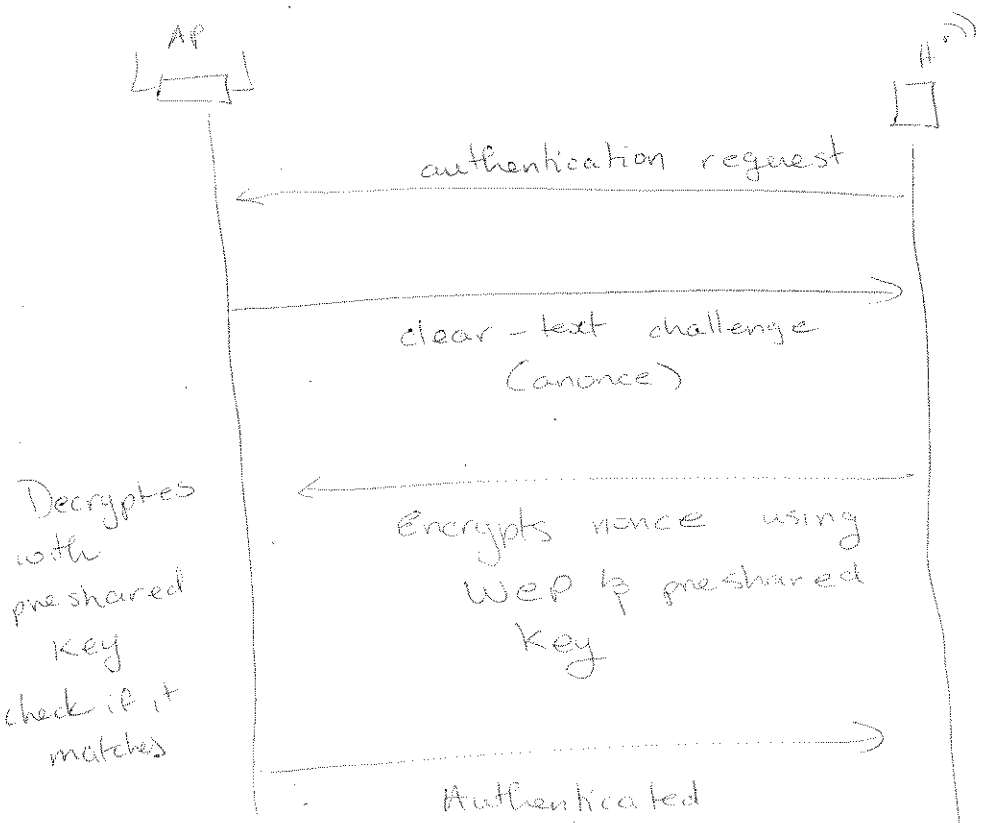
③ Association Request frame

④ Association Response frame

How do we ensure that only authorize people have access to the network

① Pre shared key

@ nonce
→ random number used once



If we capture messages from the exchange it is possible to extract the pre shared key

Intercepting Mobile Communications the Insecurity of 802.11
Borisov, Goldberg, Wagner 2001

How do we crack this exchange

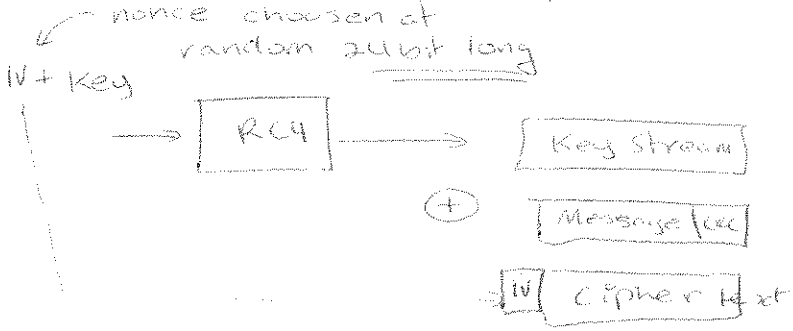
64 wep key is only 10 hexadecimal character
(0-9) & (A-F) not that big of a key
Space

so if we ~~che~~ capture the exchange

K (nonce) ⟶ we can brute force it
(Dictionary attack it)
until we ~~be~~ find the key.

① Remember the nonce was sent plain-text.
So we keep try key combination until we get the key
→ we can do this offline
no need to interact with the AP

So lets do a deep dive into WEP
So we can see this in action



Turn out that 24 bits of an IV is not enough if you listen for long enough on a network the IV repeat after about 5000 messages (Don't have to listen for long 30s - 60s)

So what if capture the duplicate IVs

$P = \langle M, c(M) \rangle$ ← plaintext is message plus the crc of the message

$C = P \oplus RC4(v,k)$

v we can see from the packet

Decryption Process

$P' = C \oplus RC4(v,k)$

$= (P \oplus RC4(v,k)) \oplus RC4(v,k)$ ← since the same IV was used the key is the same

$P' = P$

↑ same ~~pseudo-rng~~ properties used as with the one time pad reduction

Still so what if the IV get repeated.

Let's consider the case of IV reuse

$$C_1 = P_1 \oplus RC4(v, K)$$

$$C_2 = P_2 \oplus RC4(v, K)$$

xor two ciphers

$$C_1 \oplus C_2 = (P_1 \oplus RC4(v,K)) \oplus (P_2 \oplus RC4(v,K))$$

$$= P_1 \oplus P_2$$

↑ no secret material

Know attack for recovering P_1 & P_2

So far we been looking @ RC4 ~~abstract~~ abstractly what are the details of the algorithm?

RC4 invented by Ron Rivest 1987 same guy as RSA

goal generate a predictable random stream

① Pick a key "VVIA"
↳ ASCII → 55, 56, 41

② Initialize an S array from which we will sample from
S[256] = {0, 1, 2, 3, ... 255}

③ Extend the key to fill a new array K sometime denoted by T
K[256] → repeat the key 55, 56, 41, 55, 56, 41

Two steps

1. Key Scheduling
2. Pseudo Random generation

Key Scheduling

```
J = 0
for i -> 0 .. 255
    J = J + S[i] + K[i] mod 256
    swap(S[i], S[J])
```

use key to determine new index in S array

deterministic mixing

Now it time to generate a Pseudo random sequence from our S array

```
I = 0, J = 0
keyStream[] = []
while true
```

keep going so we have key long enough for our message

```
    J = S[i] mod 256
    swap(S[i], S[J])
    t = S[i] + S[J] mod 256
    keystream.push(st)
```

contain our Pseudo random sequence

1. WPA also not secure build on top of WEP
2. WPA3 is currently considered the best standard.