# ECE/CS 439
# MP 2: Wireless Tools
# Due 24 November 2017

## Doppler sensing

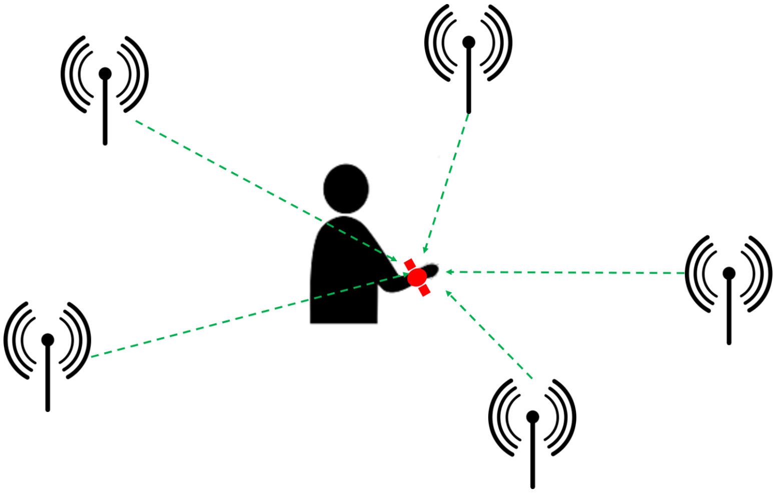

Figure 1: Doppler sensing of user velocity

Wireless networks have a number of sensing applications in addition to communication. In this problem, you would design a velocity sensor using wireless signals. Consider Figure.1, showing five access points communicating with a smart-watch worn by a user. Our goal here would be to detect the velocity of hand motion of the user. To this end, we would use doppler shifts of the AP signals detected at the smartwatch. Assume that the access points are transmitting pure sin waves (no modulation) at a carrier frequency of 5.800 GHz, and there is no multipath in the environment (channel $h = 1$). Determine the velocity of the hand, based on doppler shifts observed at the smartwatch (the observations are given to you as data files as described next)

**Input format:** Your program would take the 3D locations of the access points as well as the user as inputs. In addition, the program would also take 5 second long downconverted and downsampled baseband signals (sampled at $1Mhz$) received on the smart-watch (from each of these access points). Using these inputs, compute the doppler shifts relative to each access point, and use the information to compute the 3D velocity of hand motion. (Hint: A doppler shift at $5.8GHz$ would appear as a frequency offset in baseband. Hence, a zero doppler shift would only result in a DC component at baseband)

Your program needs to take an "input.txt" file with following parameters. First line would indicate the location of the client. The second line would indicate the number of APs $N$. The next $N$ lines would indicate 3D co-ordinates of each AP. Then, the program would also need to input the baseband signals from each of the $N$ APs in the order they were listed in "input.txt". An example "input.txt" for a 4 AP case is given in [5]. The baseband signals for each AP must be input as separate files. Input files for the 4 APs in the example are given in [1–4].

**Output format:** Your program needs to output a single line containing the 3D velocity of the user hand motion. Your submission must include the program as well as a "readme.txt" file with instructions to run it, so that it can be evaluated for other inputs.

## Network utilization

Design a simulator to analyze the utilization of a wireless network performing CSMA/CA with random backoff. Consider $N$ nodes all of which can hear each other (No hidden or exposed terminals). The pseudo-code below provides an incomplete high level algorithm to compute the utilization.

**Algorithm 1** MAC Pseudo-code

```
choose $N$ random numbers $r[i]\ \forall\ i \in \{1..N\}$ for $N$ nodes
while $total\_time < T$ do
    find the node with the minimum counter (multiple nodes with minimum counter would cause collision)
    if there is no collision then
        $good\_time = good\_time + packet\_time$
        for the node $j$ that transmitted, first update the contention window and then choose a new counter $r[j]$
    else
        $\forall$ nodes $k$ that collided, first update the contention window and then choose a new counter $r[k]$
    end if
    $total\_time = total\_time + packet\_time$
    update counters of all nodes $i$ that did not collide:
    $r[i] = r[i]$ - minimum backoff, $\forall\ \ i$
end while
$utility = \frac{good\_time}{total\_time}$
```

Compute the utility as a function of the number of nodes (N) for the following backoff strategies (recall that initial backoff is a random number in the range of [0, $CW\_min$], and the backoff grows with collisions).

1. 802.11 (double the contention window (CW) for every collision, and cut-down CW to $CW\_min$ upon successful transmission)

2. Additive increase multiplicative decrease (increment CW by 2 for every collision, and cut-down by half upon successful transmission)

3. Multiplicative increase multiplicative decrease (double CW for every collision, and cut-down by half upon successful transmission)

4. Additive increase additive decrease (increment CW by 2 for every collision, and decrement by 2 for successful transmission)

5. Multiplicative increase additive decrease (double CW for every collision, and decrement by 2 for successful transmission)

Assume $CW\_min = 15$, i.e., initial backoff = rand [0, 15]. Also, for all strategies, let the transmitter drop the packet after successive 10 collisions (a new packet contention always starts with $CW\_min$).

Perform the above analysis for three different packet sizes 1000, 2000, and 10000 bytes transmitted at a data rate of $6Mbps$. Assume a slot size of $9us$ for backing off. Ignore any headers, and there is no

RTS/CTS exchange.

**Input format:** One line with four numbers - Nodes $N$, packet size $P$ in bytes, simulation length $T$ in milliseconds, and the backoff strategy (between 1-5 above)

**Output format:** One number which is the network utilization. Your submission must include the program as well as a "readme.txt" file with instructions to run it so that it can be evaluated for other inputs. Your submission must also include three graphs. The first graph would use a packet size of 1000 bytes and plot the utilization vs number of nodes (1 to 200) for all five backoff strategies – five plots in the same graph for comparison. The second and third graph would repeat the same for 2000 and 10000 byte packets respectively.

## References

[1] Ap1 baseband signal. `https://courses.engr.illinois.edu/ece439/fa2017/mp2/AP1.txt`.

[2] Ap2 baseband signal. `https://courses.engr.illinois.edu/ece439/fa2017/mp2/AP2.txt`.

[3] Ap3 baseband signal. `https://courses.engr.illinois.edu/ece439/fa2017/mp2/AP3.txt`.

[4] Ap4 baseband signal. `https://courses.engr.illinois.edu/ece439/fa2017/mp2/AP4.txt`.

[5] input configuration file. `https://courses.engr.illinois.edu/ece439/fa2017/mp2/input.txt`.