# Mobile Networks Report

Yanbin He 5037751

June 21, 2020

## 1 Derivation of bit rate

### 1.1 Procedure

In this part, I will show the process of deriving the SINR values of the corresponding 50 PRB-specific traces.

First, let's consider the first user case, user0. The definition of SINR is shown in the following:

$$SINR = \frac{Power_{signal}}{Power_{Interference} + Power_{Noise}} \tag{1}$$

The SINR values are defined on different PRBs. Thus, we can calculate them one by one. The signal power in one PRB is the power divided by the number of PRBs, since the power is equally distributed. Then, we can calculate the path gain between user and base station which serves this user, based on the following equation:

$$G = -(115 + 35\log_{10}(distance_{km}))(dB) \tag{2}$$

Taking the multipath fading into consideration, we can know the signal power of user0 in each PRB chaging with transmission time interval.

$$Power_{signal}(dB) = Power_{Transmitted}(dB) + G(dB) + MF(dB) \tag{3}$$

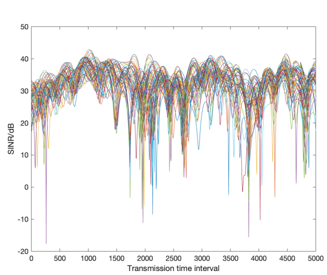
The calculation of interference power is more complicated than that of signal power. First, we have to be clear that all the other cells are serving their own users. Thus each of them can transmitted corresponding PRB signals. Due to the fact that the reuse factor is 1, we have to consider all of these cells. For each PRB, the interference is from the same PRB of other cells. Therefore, we can derive this one by one as well. First, we can calculate the power of each PRB in different cells. Then, we can derive the distances between different base stations to user0 and calculate the path gain. After that, we can use the simplification introduced in the assignment material to derive the multipath fading from each cell to the user0, which is only related to the index of the cell. We take the first $(5000 - 135 * b)$ values if the interference is from cell b and then add the rest to its end to make sure that each trace has 5000-TTI. Finally, we just add up all the interference power from each cell and derive the total interference. We have to be careful that all of these calculations are in dB (or dBm), if we want to sum the powers up, we have to first translate them into mW, which is linear unit. The power of noise is straightforward.

$$N = kTB \tag{4}$$

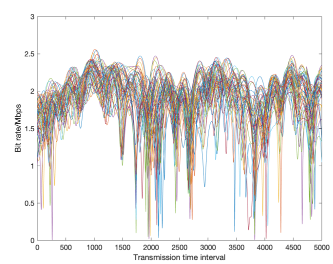
where k is Boltzmann constant $1.38064852 * 10^{-23}$, T is the temperature $290K$ and B is bandwidth $180000Hz$.

## 1.2 Result

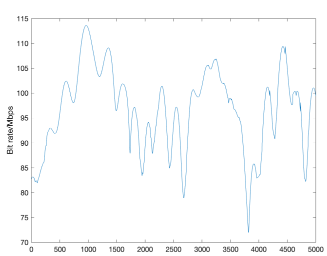
The results of the first and second questions are shown below. The first picture illustrates the corresponding 50 PRB-specific traces of SINR values of user0.



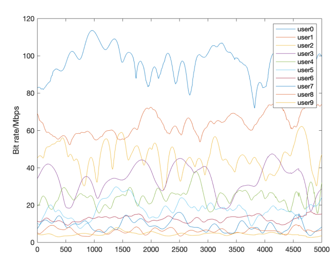

(a) 50 PRB-specific traces of SINR of user0




(b) Attainable bit rates of SINR traces of user0




(c) Aggregate bit rate for user0




(d) Aggregate bit rate for all the users

Figure 1: Solutions to Question 1 and 2

The fourth picture is the answer to the second question. In this figure ,we are going to show the ten use-specific traces of aggregate (over the 50 PRBs) bit rates. The following table shows the time-averaged aggregate bits rate for 10 users. The unit is Mbps. As we can see, with the distance between user and antenna gets larger, the throughput experienced by the user decreases. This is because the path gain gets lower. As a consequence, the SINR is lower, thus the bit rate gets lower. The cell edge user experiences the lowest bit rate.

| User0 | User1 | User2 | User3 | User4 | User5 | User6 | User7 | User8 | User9 |
|---|---|---|---|---|---|---|---|---|---|
| 96.74 | 63.37 | 45.65 | 33.87 | 24.28 | 17.74 | 12.63 | 9.08 | 6.17 | 4.46 |

## 2 Packet Scheduling

In this section, we are going to calculate the average throughput experienced by each user, aggregate cell throughput and Jain's fairness index to show the throughput fairness and resource fairness of different packet scheduling strategy. All of the results will be organized as table in appendix. The Jain's fairness index is a value between 0 and 1 to show the consistency of all the input entries. The formula is:

$$\mathcal{J} = \frac{(\sum_{i=1}^{n} x_i)^2}{n \times \sum_{i=1}^{n} x_i^2} \tag{5}$$

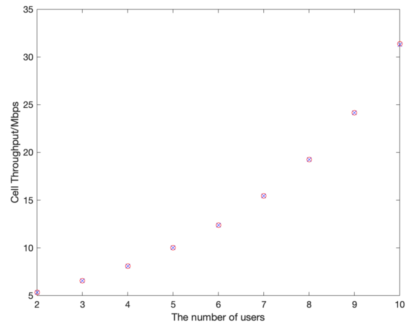
$x_i$ in this equation can be throughput experienced by each user or the number of resources (PRB). All the results are shown in the Appendix Figure 4 table 2. Scheme 1 is Round Robin only in time domain. Scheme 2 is maximum rate scheduling only in time domain. Scheme 3 is proportional fair scheduling only in time domain. Scheme 4 is Round Robin in both time and frequency domain. Scheme 5 is maximum rate scheduling in time/frequency domain. Scheme 6 is proportional fair scheduling in both time and frequency domain.

## 3 Discussion

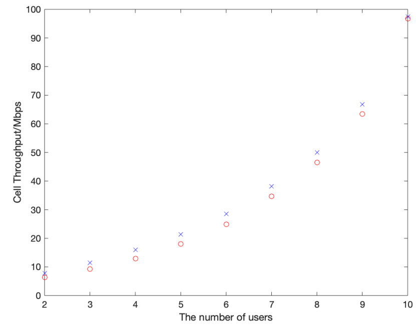
In this part, we are going to discuss the results from section 2 and show the phenomena of multiuser diversity gain, frequency diversity gain and some trade-offs between efficiency and throughput or resource fairness.

Multiuser diversity gain is defined as the growth of network throughput with the increase of number of users[1]. We can observe this phenomenon from results shown in Figure 2. But this is the gain from adding user closer to the base station. Therefore, I simulate another 2 cases. The first one is adding user further to the station and the second one is adding users in the middle. In first case, except for maximum rate scheduler in time/frequency domain, all the other cases don't show any gains in throughput. The same thing happens to adding users in the middle. Therefore, we cannot observe multiuser diversity gain all the time in all cases.

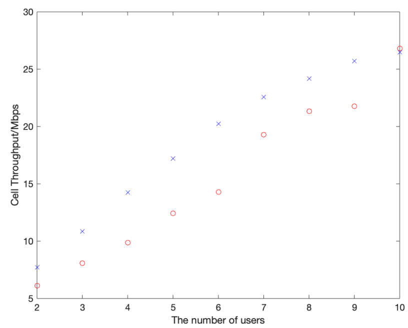
The results about frequency diversity gain are shown in Figure 2. Each sub-figure corresponds to different scheduler. As we can see, the introduction of frequency domain cannot bring throughput gain to the Round Robin scheduler. This is because the ways of distributing are quite similar in both cases, shown in Figure 4 in Appendix. For maximum rate scheduler, we can observe this frequency diversity gain, shown in Figure 2 (b). Among all of these three scheduler, we can observe that the largest gain is from proportional fair scheduler, shown in Figure 2 (c). In 2 of these 3 cases, taking frequency domain into




(a) Introducing frequency domain to Round Robin. x: with frequency




(b) Introducing frequency domain to Maximum rate. x: with frequency

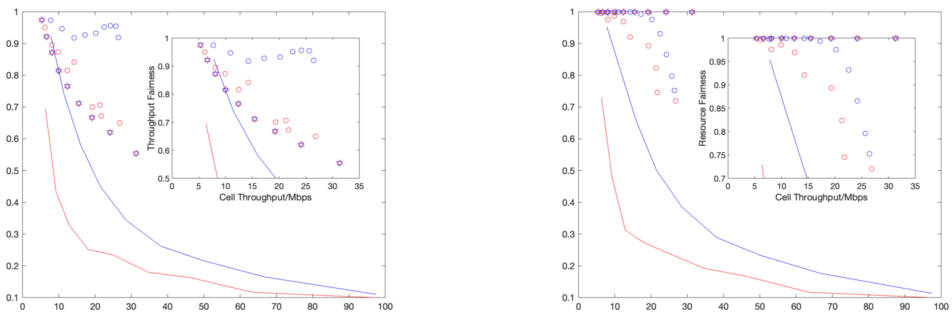



(c) Introducing frequency domain to Proportional Fair. x: with frequency

Figure 2: Comparison

consideration can provide better results. For the other case, at least the results are the same. We can attribute this to the higher granularity of assigning resources. In the time-domain-only scheduler, we only choose the best one (this "best" has different definitions regarding different scheduling strategies) in one TTI. However, one TTI has 50 PRBs. It is unclear how each user performs in terms of each PRB. But if we take frequency into account, we can determine which one performs best at PRB level. This will lead to a gain by choosing those have best throughput for each PRB, which is here frequency diversity gain.

The trade-offs between efficiency and throughput or resource fairness are illustrated in Figure 3. Figure 3 (a) is cell throughput versus throughput fairness, (b) is cell throughput versus resource fairness. The small figure is the enlargement of the top left part to make results clearer. As shown in this figure (a), maximum rate scheduler has the lowest throughput fairness index. And the throughput fairness index of round robin is larger than that of maximum rate. The proportional fair has the largest fairness index. In the Figure 3 (b), we




(a) Cell throughout versus Throughput Fairness

(b) Cell throughout versus Resource Fairness

Figure 3: Trade-offs. Blue: with frequency, red: without frequency. Line: Maximum rate, triangle: Round robin, circle: Proportional fair

can see that maximum rate scheduler has the lowest fairness value as well. But in this case, the fairness index value of proportional fair scheduler is between maximum rate and round robin which has the highest fairness value 1 all the time.

We can go into horizontal direction. In both cases, we can see that maximum rate scheduler can generate highest cell throughput, and round robin is much lower than that but still slightly larger than proportional fair scheduler.

So from the implementations of these 3 schedulers in both time-domain-only case and time-frequency-domain case, we can conclude that, first, from the perspective of efficiency, maximum rate scheduler performs best because it always chooses the user which has best SINR and can generate highest bit rate. So from the perspective of cell, it's the most efficient. From the perspective of resource fairness, round robin performs best because it distributes all the resources equally to all the users in an unbiased way but it's not efficient from the perspective of efficiency. From the perspective of throughput fairness, no one is definitely fair but the proportional fair in time-frequency-domain scheduler performs better than any other. Because it can somewhat pick up the peak value of bit rate curve of a user and assign the resource block, which can have more uniform users' bit rates.

And the trade-offs between efficiency and throughput or resource fairness can be observed here as well. If we opt for efficiency, for example, implementing the maximum rate scheduler, we are at the cost of reducing the fairness of throughput, for example, the results shown in Figure 3 (a) or reducing the fairness of resource, as the results show in Figure 3 (b). And if we want to make sure that regarding resource or throughput each user's experience is equal, we will sacrifice the efficiency, namely, the throughput of the cell as a whole. For example, as shown in Figure 3 (a), if we opt for proportional fair in time/frequency domain, we are the fairest regarding throughput experienced by users, but we will have the lowest cell throughput. And in Figure 3, the same happens to Round robin as well. In a word, we can't achieve efficiency and fairness at the same time in this case, that's where the trade-off comes from.

## 4 Time Estimation

This assignment approximately took me 25-30 hours to do assignment and write the report. 10-15 hours are spent on writing Matlab codes, checking results, making tables, plotting and debugging. Around 15 hours are spent on thinking, refreshing with video, writing and checking the report.

## A Simulation Results

| Users-Scheme 1 | User Number | Resource Fairness | Traffic Fairness | Aggregate Throughput/Mbps | User Throughput (Corresponding to user list)/Mbps |
|---|---|---|---|---|---|
| 8 9 | 2 | 1 | 0.9746 | 5.3123 | 3.0846; 2.2277 |
| 7 9 | 3 | 1 | 0.9221 | 6.5672 | 3.0258; 2.0568; 1.4846 |
| 6 9 | 4 | 1 | 0.8716 | 8.0826 | 3.1574; 2.2689; 1.5424; 1.1138 |
| 5 9 | 5 | 1 | 0.8159 | 10.0148 | 3.5484; 2.5262; 1.8152; 1.2339; 0.8911 |
| 4 9 | 6 | 1 | 0.7649 | 12.3956 | 4.0499; 2.9593; 2.1042; 1.5121; 1.0278; 0.7422 |
| 3 9 | 7 | 1 | 0.7112 | 15.4652 | 4.8431; 3.4724; 2.5327; 1.8036; 1.2962; 0.8811; 0.6362 |
| 2 9 | 8 | 1 | 0.6673 | 19.2349 | 5.7063; 4.2341; 3.0346; 2.2181; 1.5790; 1.1346; 0.7713; 0.5569 |
| 1 9 | 9 | 1 | 0.6202 | 24.1513 | 7.0471; 5.0795; 3.7662; 2.6996; 1.9726; 1.4021; 1.0077; 0.6849; 0.4946 |
| 0 9 | 10 | 1 | 0.5536 | 31.3979 | 9.6732; 6.3366; 4.5651; 3.3872; 2.4274; 1.7749; 1.2633; 0.9077; 0.6170; 0.4455 |
| Users-Scheme 2 | | | | | |
| 8 9 | 2 | 0.7291 | 0.6946 | 6.3401 | 5.2723; 1.0678 |
| 7 9 | 3 | 0.4796 | 0.4324 | 9.2467 | 8.0457; 1.0770; 0.1240 |
| 6 9 | 4 | 0.3125 | 0.3285 | 12.8801 | 11.0937; 1.7864; 0; 0 |
| 5 9 | 5 | 0.2728 | 0.2516 | 18.0837 | 16.0172; 1.8249; 0.2416; 0; 0 |
| 4 9 | 6 | 0.2391 | 0.2345 | 24.9191 | 20.5456; 4.3735; 0; 0; 0; 0 |
| 3 9 | 7 | 0.192 | 0.1797 | 34.6907 | 30.6646; 4.0261; 0; 0; 0; 0; 0 |
| 2 9 | 8 | 0.1665 | 0.1625 | 46.5175 | 40.3308; 6.1867; 0; 0; 0; 0; 0; 0 |
| 1 9 | 9 | 0.1169 | 0.1164 | 63.4357 | 61.9643; 1.4713; 0; 0; 0; 0; 0; 0; 0 |
| 0 9 | 10 | 0.1 | 0.1 | 96.7427 | 96.7427; 0; 0; 0; 0; 0; 0; 0; 0; 0 |
| Users-Scheme 3 | | | | | |
| 8 9 | 2 | 0.9958 | 0.9503 | 6.1232 | 3.7620; 2.3612 |
| 7 9 | 3 | 0.9755 | 0.8949 | 8.0836 | 3.7520; 2.8292; 1.5024 |
| 6 9 | 4 | 0.9856 | 0.8728 | 9.8602 | 3.0234; 3.6797; 1.8813; 1.2759 |
| 5 9 | 5 | 0.9695 | 0.8155 | 12.4257 | 4.2247; 2.3942; 3.3150; 1.5548; 0.9370 |
| 4 9 | 6 | 0.9211 | 0.842 | 14.2826 | 3.2798; 3.5696; 1.8624; 3.2651; 1.3687; 0.9370 |
| 3 9 | 7 | 0.8933 | 0.7 | 19.297 | 6.5682; 2.9606; 3.0331; 1.8309; 3.1071; 1.2664; 0.5307 |
| 2 9 | 8 | 0.8235 | 0.7066 | 21.3211 | 2.4527; 6.5682; 2.9312; 2.9758; 1.6581; 3.0507; 1.2541; 0.4303 |
| 1 9 | 9 | 0.7454 | 0.6718 | 21.7645 | 0.6838; 2.4527; 6.3864; 2.9312; 2.9758; 1.6058; 3.0507; 1.2478; 0.4303 |
| 0 9 | 10 | 0.72 | 0.6496 | 26.8245 | 6.3699; 0.3537; 2.4527; 5.8722; 2.9312; 2.9481; 1.2459; 3.0165; 1.2040; 0.4303 |
| Users-Scheme 4 | | | | | |
| 8 9 | 2 | 1 | 0.9749 | 5.3249 | 3.0893; 2.2355 |
| 7 9 | 3 | 1 | 0.9222 | 6.567 | 3.0254; 2.0565; 1.4851 |
| 6 9 | 4 | 1 | 0.8721 | 8.0896 | 3.1587; 2.2684; 1.5447; 1.1178 |
| 5 9 | 5 | 1 | 0.8142 | 9.9998 | 3.5584; 2.5192; 1.8039; 1.2294; 0.8889 |
| 4 9 | 6 | 1 | 0.7657 | 12.3958 | 4.0438; 2.9588; 2.1060; 1.5123; 1.0298; 0.7451 |
| 3 9 | 7 | 1 | 0.7116 | 15.461 | 4.8385; 3.4687; 2.5347; 1.8045; 1/2967; 0.8813; 0.6365 |
| 2 9 | 8 | 1 | 0.6674 | 19.2395 | 5.7073; 4.2356; 3.0329; 2.2189; 1.5793; 1.1341; 0.7724; 0.5589 |
| 1 9 | 9 | 1 | 0.6205 | 24.1387 | 7.0411; 5.0726; 3.7633; 2.6978; 1.9714; 1.4035; 1.0084; 0.6855; 0.4951 |
| 0 9 | 10 | 1 | 0.5538 | 31.2822 | 9.6645; 6.2564; 4.5470; 3.3775; 2.4428; 1.7739; 1.2592; 0.9031; 0.6096; 0.4483 |
| Users-Scheme 5 | | | | | |
| 8 9 | 2 | 0.9532 | 0.9241 | 7.809 | 5.0238; 2.7852 |
| 7 9 | 3 | 0.8194 | 0.7397 | 11.4682 | 6.9055; 3.0469; 1.5157 |
| 6 9 | 4 | 0.655 | 0.5823 | 15.9712 | 9.3575; 4.3815; 1.4797; 0.7525 |
| 5 9 | 5 | 0.5022 | 0.4501 | 21.4509 | 13.0282; 5.4073; 2.2803; 0.5082; 0.2268 |
| 4 9 | 6 | 0.3853 | 0.3443 | 28.4929 | 18.5008; 6.6895; 2.2812; 0.8663; 0.0954; 0.0598 |
| 3 9 | 7 | 0.2888 | 0.2618 | 38.1453 | 26.7758; 8.4902; 2.1029; 0.6348; 0.1302; 0.0075; 0.0039 |
| 2 9 | 8 | 0.2337 | 0.2159 | 49.97 | 36.3901; 10.7205; 2.5185; 0.2251; 0.0989; 0.0169; 0; 0 |
| 1 9 | 9 | 0.1764 | 0.1651 | 66.7239 | 53.5706; 11.0861; 1.7890; 0.2635; 0.0072; 0.0075; 0; 0; 0 |
| 0 9 | 10 | 0.1133 | 0.1104 | 97.4924 | 92.6519; 4.6079; 0.1867; 0.0459; 0; 0; 0; 0; 0; 0 |
| Users-Scheme 6 | | | | | |
| 8 9 | 2 | 0.9975 | 0.9733 | 7.73 | 4.5048; 3.2251 |
| 7 9 | 3 | 0.9993 | 0.947 | 10.869 | 4.7358; 3.4839; 2.6493 |
| 6 9 | 4 | 0.9999 | 0.9173 | 14.2528 | 5.1201; 3.9336; 2.8805; 2.3186 |
| 5 9 | 5 | 0.9934 | 0.9272 | 17.2199 | 4.6319; 4.3698; 3.4700; 2.5923; 2.1559 |
| 4 9 | 6 | 0.9754 | 0.9323 | 20.2195 | 4.6807; 3.8979; 3.9267; 3.2386; 2.4172; 2.0585 |
| 3 9 | 7 | 0.9316 | 0.9523 | 22.5644 | 3.627; 4.1322; 3.6147; 3.7355; 3.0898; 2.3535; 2.0117 |
| 2 9 | 8 | 0.8654 | 0.9562 | 24.1953 | 2.5187; 3.2342; 3.9443; 3.4901; 3.6712; 3.0302; 2.3168; 1.9898 |
| 1 9 | 9 | 0.7964 | 0.955 | 25.7028 | 2.3485; 2.2726; 2.9639; 3.7811; 3.4082; 3.6264; 3.0158; 2.3034; 1.9830 |
| 0 9 | 10 | 0.7523 | 0.9198 | 26.4829 | 1.1722; 2.1481; 2.1875; 2.9339; 3.7582; 3.3818; 3.6089; 3.0109; 2.3007; 1.9806 |

Figure 4: Table 2

# References

[1] A. Tajer and X. Wang, "Multiuser diversity gain in cognitive networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 6, pp. 1766–1779, 2010.