SECTION BELOW IS FOR DISCUSSION / INFORMATION PURPOSES ONLY, SINCE POC WORKING GROUP DECIDED TO WAIT UNTIL THE MAPPERS ARE OUT AND SPLIT REWARD ADJUSTMENTS INTO A SEPARATE HIP

Adjusting Hotspot Rewards Based on Mapper Input

Confidence score is assigned to each cell by mobile rewards oracle; it represents how likely the cell's modeled coverage actually aligns with reality, based on data sent by mappers. We suggest that the confidence score ranges from 0 to 10 and, in the beginning, everybody starts out with a confidence score of 5. Each time a mapper shares data about cell's coverage, it's reading will either increase or decrease the confidence score of a cell as the function of the following:

- Does the mapper reading agree or disagree with modeled coverage. Initially we'll just be looking at the presence or absence of signal in a modeled coverage hex vs. validating actual signal strength. As such, an outcome of any mapper reading can be either positive (it confirms signal in modeled coverage hex) or negative (it reports absence of signal in modeled coverage hex).
- Does the reading validate or invalidate mapper reading that preceded it and to what extent.

Mapper reading will be considered "first reading" if the cell was never mapped before during a given period (TBD) or if the reading that preceded it does not agree with the current reading (i.e current reading is positive, but previous - was negative or vice versa). Mapper reading will be considered repeat reading, if it agrees with the previous reading.

For each first positive reading mobile oracle will increase the confidence score of a cell by 1 point. For each first negative reading - decrease it by 1 point. For each repeat reading, the confidence score increase or decrease will be adjusted by a repeat reading multiplier, which is a function of how relevant this repeat reading is and, therefore, how likely it is to confirm that the first reading was indicative of the true state. We propose to anchor repeat reading multipliers on the following conditions:

- Time interval between readings. To be most relevant, repeat reading should be some time after the first reading, but not too long after the first reading. I.e. if you have a mapper that provides multiple negative readings in a short period of time, it may have been a temporary issue (a truck parked in front of the cell or the cell was temporarily rebooting or temporary lapse in spectrum access etc.). If, however, you have a repeat negative reading 30 minutes after the one that preceded it - it is more indicative of a permanent issue.
- Is the repeat reading in the same hex. If we are reporting negative reading in the same hex over and over again, it doesn't mean that the cell is off or cheating. It may just imply a modeled coverage error.
- Is the reading by the same mapper. Maybe the mapper is broken?

Therefore, repeat reading will increase or decrease the confidence score by the number of points that is equal to [previous reading] * [repeat reading multiplier].

	If the last mapper reads the same as the last one...			
	Was the last reading more than 30 mins and less than 72 hours ago?	Was the last reading conducted by the same mapper?	Was the last reading in the same hex?	Repeat Reading Multiplier
Scenario 1	yes	yes	yes	1.2
Scenario 2	no	yes	yes	0
Scenario 3	no	no	yes	0.3
Scenario 4	no	no	no	2
Scenario 5	no	yes	no	1.5
Scenario 6	yes	no	no	3
Scenario 7	yes	yes	no	2
Scenario 8	yes	no	yes	1.4

As a result of mapping activity over time, all cell's confidence scores will start changing. Those cells that get a lot of positive readings will move towards 10. Those that have negative readings - will move towards zero.

Confidence score of each cell can then be used to reward honest players and weed out cheaters. It is reasonable to assume that we will uncover many corner cases and imperfections to the above algorithm as we roll out verification mapping. In particular, anchor conditions and values for repeat multipliers are our best case guesses at this point. However, we expect that we'll be able to tweak this framework over time by adding / changing anchor conditions and adjusting repeat multipliers through a community voting process to make the confidence score a fairly accurate predictor of how well the cell's verified coverage aligns with its modeled coverage.

Using The Confidence Score to Adjust Rewards

Confidence score of each cell will be used to adjust its reward points in one of the following ways.

- Introduce a universal multiplier of the total reward points, attributable to the cell, that would equal [confidence score]/10. I.e. if your cell earns 1000 points based on modeled coverage and your confidence score is 7, your actual reward points earned will be 700.
- Instead of multiplying reward points by confidence score, introduce a step function that would mostly punish and reward the outliers. I.e. those, who are most likely cheaters, will

get very little or no rewards. Those who are getting consistently verified, will get some small bonus. Example:

Confidence Score	Reward Multiplier
0-3	0
3-7	1
7-10	1.3

- Instead of looking at the raw confidence score to adjust the reward multiplier, we can use the confidence score of a given cell in its relation to the confidence score of the general population to calculate the probability of the cell cheating or outperforming and attach reward multipliers to certain probability thresholds.

We'd like to ask for community feedback on the best algorithm to adjust rewards as a result of the confidence score.