Estimating Robot Power Consumption: Theory and Experiments

u/promercyonetrick

July 21, 2018

In the course of building megabases, we often want to estimate the power consumption of logistic robots, so that we can plan out how much space to leave for roboports and how many robots are required. In this article, we discuss several methods for estimating the power consumption of logistic robots.

1 Basic Case: Single Requester Chest, Single Provider Chest

1.1 Roundtrip Power Consumption

The simplest use case of logistic robots is to deliver from a provider chest to a requester chest. For this calculation, we assume that the provider chest is always filled with more than enough items, and the items in the requester chest are cleared (by, for instance, an inserter) at a constant rate. We list some of the variables that can affect robot power usage:

robot speed: v
: Robot speed can be viewed from the information box that appears when you place your mouse over a robot in your inventory. This number increases as you complete more worker robot speed research; for worker robot speed 6, this number is 43.7km/h, or 12.15m/s.

distance: d
: This is the distance between the requester and the provider chest, measured in tiles (or meters).

robot cargo size: s
: The number of items a single robot can carry at once. We will assume $s = 4$ since this is the cargo size when all relevant research has been completed.

throughput: T
: The number of items removed from the requester chest per unit time.

We first compute the power consumption P_0 of a single moving robot. We know that the robot has a drain of 3kW and consumes 5kJ per meter, so the moving robot has power:

$$P_0 = 3\text{kW} + (5\text{kJ/m})v. \tag{1}$$

Now we compute the throughput T_0 of a single robot. The robot delivers items in roundtrips between the two chests: it first goes to the requester chest to pick up the items, then it goes to the provider chest to deliver the items, then it goes back to the requester chest to repeat the process. The time t for a round trip is thus:

$$t = 2d/v, \tag{2}$$

and the throughput T_0 of the robot is:

$$T_0 = s/t = \frac{sv}{2d}. \tag{3}$$

To reach a given throughput T, we need T/T_0 robots for delivery, so the total power consumption P is:

$$P = \frac{T}{T_0}P_0 = \frac{2P_0}{sv}Td. \tag{4}$$

We see that for fixed v and s, the power consumption is proportional to throughput T and distance d.

1.2 Accounting For Robot Charge Time

When fully charged, a robot have 1500kJ of energy, so we expect the robot to charge for 1.5s each time the robot's energy is completely used up. The amount of time for the robot to use up its energy is $1500\text{kJ}/P_0$, so the number of robots required for a given throughput T should be multiplied this factor:

$$\frac{1500\text{kJ}/P_0 + 1.5\text{s}}{1500\text{kJ}/P_0} = 1.003 + (0.005\text{s/m})v. \tag{5}$$

At worker robot speed 6, this factor is only 1.06375; even at worker robot speed 20, this factor is just 1.20025. We see that this factor is not very high; as long as we have some redundancies in our design, this factor should not matter too much.

The corrected total power consumption is thus:

$$P = \frac{T}{T_0}P_0 = \frac{1500\text{kJ}/P_0 + 1.5\text{s}}{1500\text{kJ}/P_0}\frac{2P_0}{sv}Td. \tag{6}$$

Note that we didn't account for the time for the robot to fly to a nearby roboport, so the actual power consumption should be a bit higher than the above formula. Placing the roboports on the path of the robots should reduce that extra time to a minimum.

1.3 Experiment

Now we test our calculations with creative mod. The experimental setup, shown in Figure 1, includes:

- A passive provider chest, providing iron plates obtained from an infinite chest.
- A requester chest, requesting 1000 iron plates.
- A stack inserter taking iron plates out of the requester chest into a void.

The throughput of stack inserter is $T = 12/(26/60)\mathrm{s}^{-1}$. The provider and the requester chest are 32 tiles apart. Worker robot speed is at level 6, and the cargo size is 4. The relevant variables are thus given by:

$$d = 32\mathrm{m} \tag{7}$$
$$v = 12.15\mathrm{m/s} \tag{8}$$
$$s = 4 \tag{9}$$
$$T = 12/(26/60)\mathrm{s}^{-1} \tag{10}$$

Plugging these numbers into Equation (6), we have:

$$P = 2.47\mathrm{MW}. \tag{11}$$

Figure 2 shows the average power consumption of roboports in a 10-minute span. Since roboports have a drain of 50kW, the power consumption due to robot recharge is $2.9\mathrm{MW} - 8 * 0.05\mathrm{MW} = 2.5\mathrm{MW}$. This number is very close to our theoretical value.

2 Simple Case: Single Requester Chest, Multiple Provider Chest

2.1 General Formula

To simplify our formulas a bit, we fix s and v, and let:

$$C = \frac{1500\mathrm{kJ}/P_0 + 1.5\mathrm{s}}{1500\mathrm{kJ}/P_0} \frac{2P_0}{sv}, \tag{12}$$

then Equation 6 simplifies to:

$$P = CTd. \tag{13}$$

In Section 1, we assumed that there are always enough items in that single provider chest that we have. However, if the requester chest consumes items faster than what a single provider can provide, robots will move items from multiple provider chests into the requester chest. Suppose the provider chests

Figure 1: Experiment 1 setup

are labelled by i, the throughput of chest i is T_i, and the distance from chest i to the requester chest is d_i, then the total power consumption is:

$$P = C \sum_i T_i d_i. \tag{14}$$

Also, the total throughput is:

$$T = \sum_i T_i. \tag{15}$$

Since requester chests always pick from the closest provider chest, the throughput of the closer provider chests will be maxed-out before other chests; in other words, the T_i of the closer chests will be equal to the throuput of whatever entity that is putting items into chest i. Assuming d_i's are all distinct, we can order the provider chests by distance:

$$d_1 < d_2 < ... \tag{16}$$

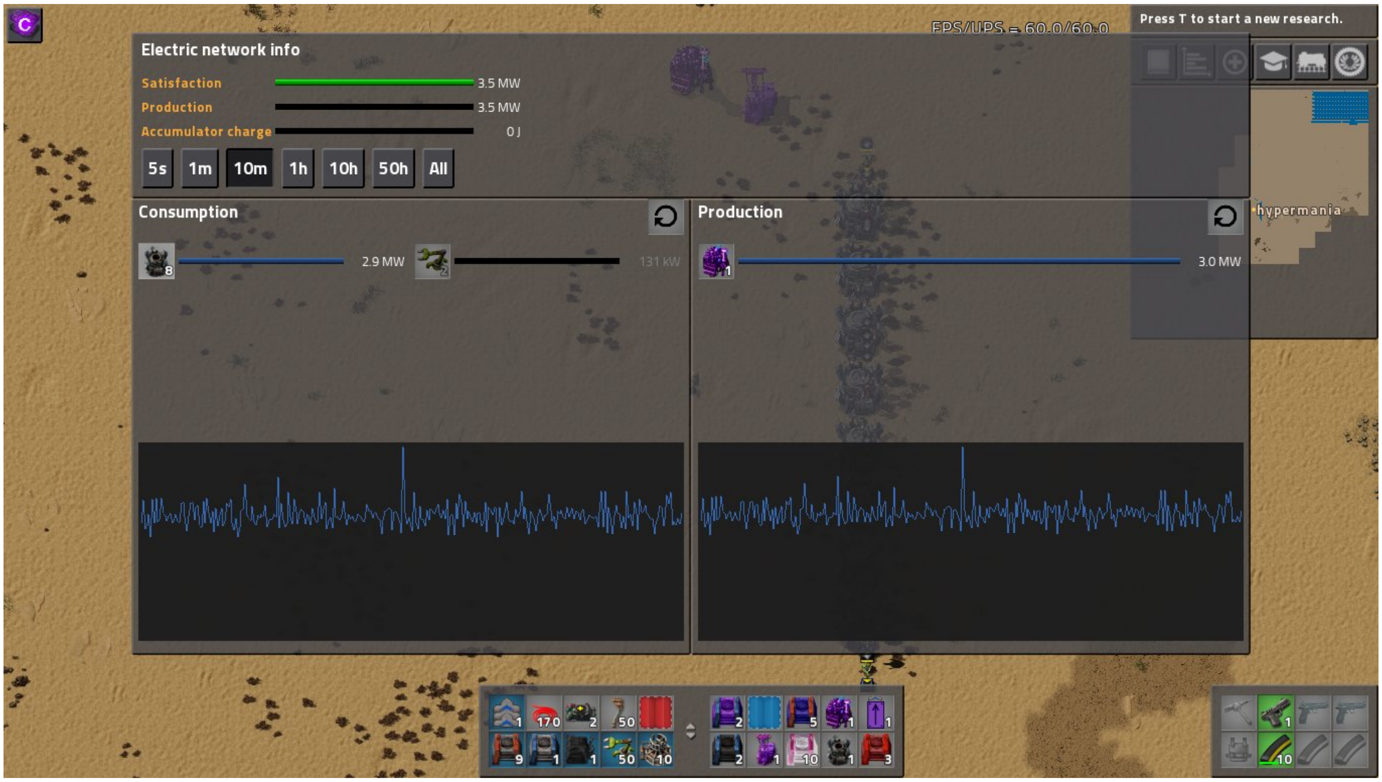

Figure 2: Experiment 1 result

Let the maximum number of items provided per unit time for chest i be S_i, then for a given goal of total throughput T, the total power consumption is:

$$P = C(\sum_{i=1}^{n} S_i d_i + (T - \sum_{i=1}^{n} S_i) d_{n+1}), \tag{17}$$

where n is the maximum integer such that:

$$T - \sum_{i=1}^{n} S_i \geq 0. \tag{18}$$

2.2 Example: Iron Smelting

We consider the setup shown in Figure 3. Below an 8-beacon iron smelting setup, two requester chests each requests 1000 iron plates, and the 5 loaders consume $T = 5 * 40 = 200/\text{s}$. (because the two requester chests are very close they can practically be regarded as one chest) The throughput S of each smelter is given by a standard calculation.

v and s are the same as those given by Equation 8 and 9. Other relevant numbers are listed below:

$$S = 1.2 * 9.4/3.5 = 3.22286/\mathrm{s} \tag{19}$$

$$T = 200/\mathrm{s} \tag{20}$$

The number n of active smelters needed is:

$$n = \frac{T}{S} = 62.0567. \tag{21}$$

Since logistic requests are fulfilled from the closest provider chests first, at a fixed throughput T, the requests are fulfilled from provider chests within a half disk of radius R. In our setup, provider chests are spaced horizontally gaps of size $\Delta x = 3$ and vertical gaps of size $\Delta y = 8$. Assuming that iron plate production is smoothed out over 3×8 areas, we can now estimate R:

$$\frac{1}{2}\pi R^2 = n\Delta x\Delta y \tag{22}$$

$$\Rightarrow R = \sqrt{(n\Delta x\Delta y)/(\frac{1}{2}\pi)} = 30.7922\mathrm{m}. \tag{23}$$

So logistic requests are fulfilled from chests within a radius of 30.7922 tiles.

Now we estimate the power consumption:

$$P = CS \sum_{(i\Delta x)^2+(j\Delta y)^2 \le R^2, j\ge 0} \sqrt{(i\Delta x)^2 + (j\Delta y)^2} \tag{24}$$

$$\approx CS \int_{(u\Delta x)^2+(v\Delta y)^2 \le R^2, v\ge 0} \mathrm{d}u\,\mathrm{d}v\, \sqrt{(u\Delta x)^2 + (v\Delta y)^2} \tag{25}$$

$$= CS \frac{1}{\Delta x\Delta y} \int_{x^2+y^2\le R^2, y\ge 0} \mathrm{d}^2\vec{r}\,|\vec{r}| \tag{26}$$

$$= CS \frac{1}{\Delta x\Delta y} \int_0^\pi \mathrm{d}\theta \int_0^R r^2\,\mathrm{d}r \tag{27}$$

$$= CS \frac{1}{\Delta x\Delta y} \frac{1}{3}\pi R^3 \tag{28}$$

$$= 11467.6\mathrm{kW}. \tag{29}$$

We approximated the summation by an integral at step 2. Since the distance function from a fixed point is convex, this approximation is likely to be lower than the actual value due to Jensen's inequaltiy.

Figure 4 shows the actual power usage of such a setup over a 10-minute span. Correcting for roboport drain, the experimental value for robot power consumption is $P = 16.5\mathrm{MW} - 0.05\mathrm{MW} * 80 = 12.5\mathrm{MW}$. This turns out to be a bit higher than our theoretical value 11467.6kW, which may be due to error in our integral approximation, error in the distance in Equation 24 (I didn't count the exact number of tiles), the time it takes for a robot to fly to a roboport for recharge, and the fact that some bot trips carry less items than the cargo size s. That said, this estimate is fairly good for practical purposes.

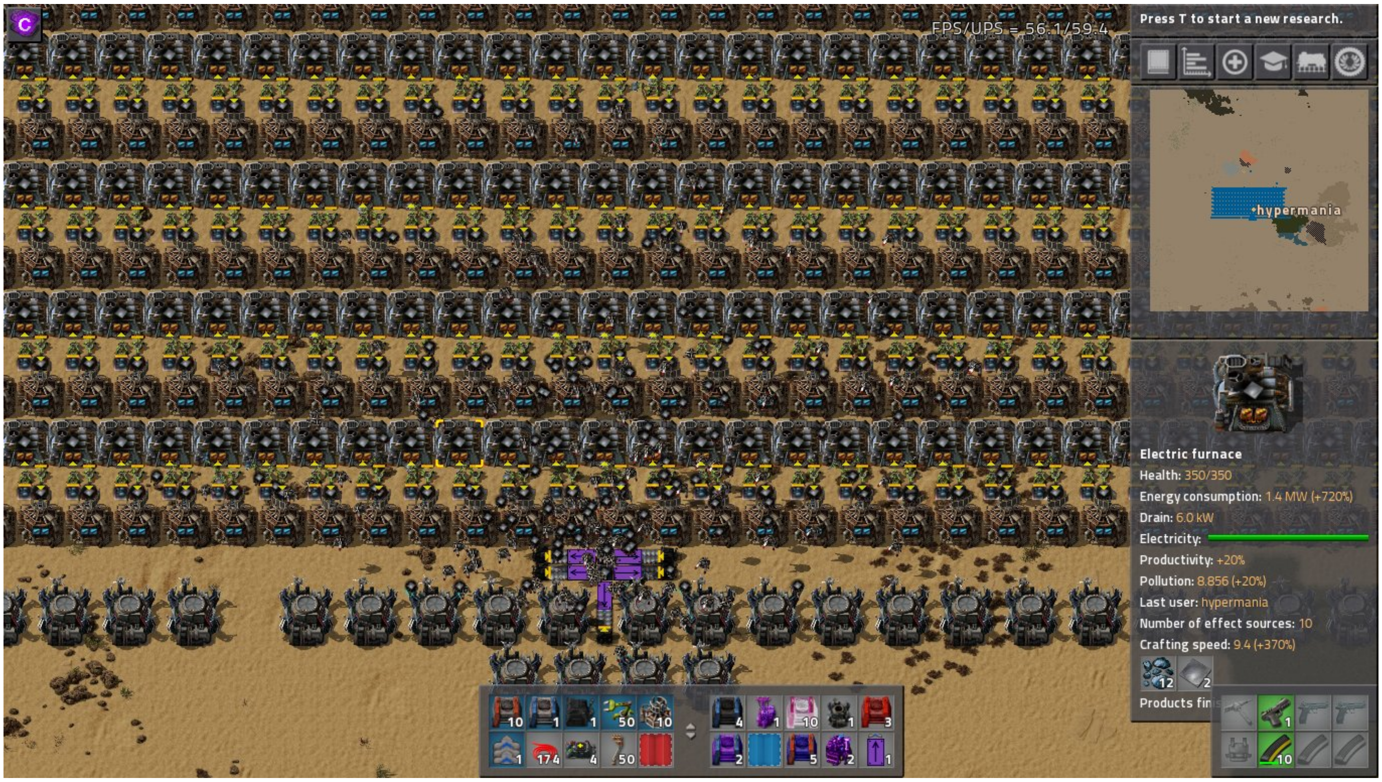

Figure 3: Experiment 2 setup

3 General Case: Multiple Requester Chest, Multiple Provider Chest

3.1 Theory

Practical use cases for logistic bots usually involve multiple requester chests and multiple provider chests. In the previous two sections, we assumed that all the throughput of a single provider belong to a single requester; However, we cannot assume these for the multiple requester chest case, since multiple requester can compete for the throughput of a single provider. As in Section 2, we label the provider chests by i, and furthermore we label the requester chests by j. Let the average throughput from provider i to requester j be T_{ij} and the distance between the two chests be d_{ij}, we have:

$$P = C \sum_{i,j} T_{ij} d_{ij}. \tag{30}$$

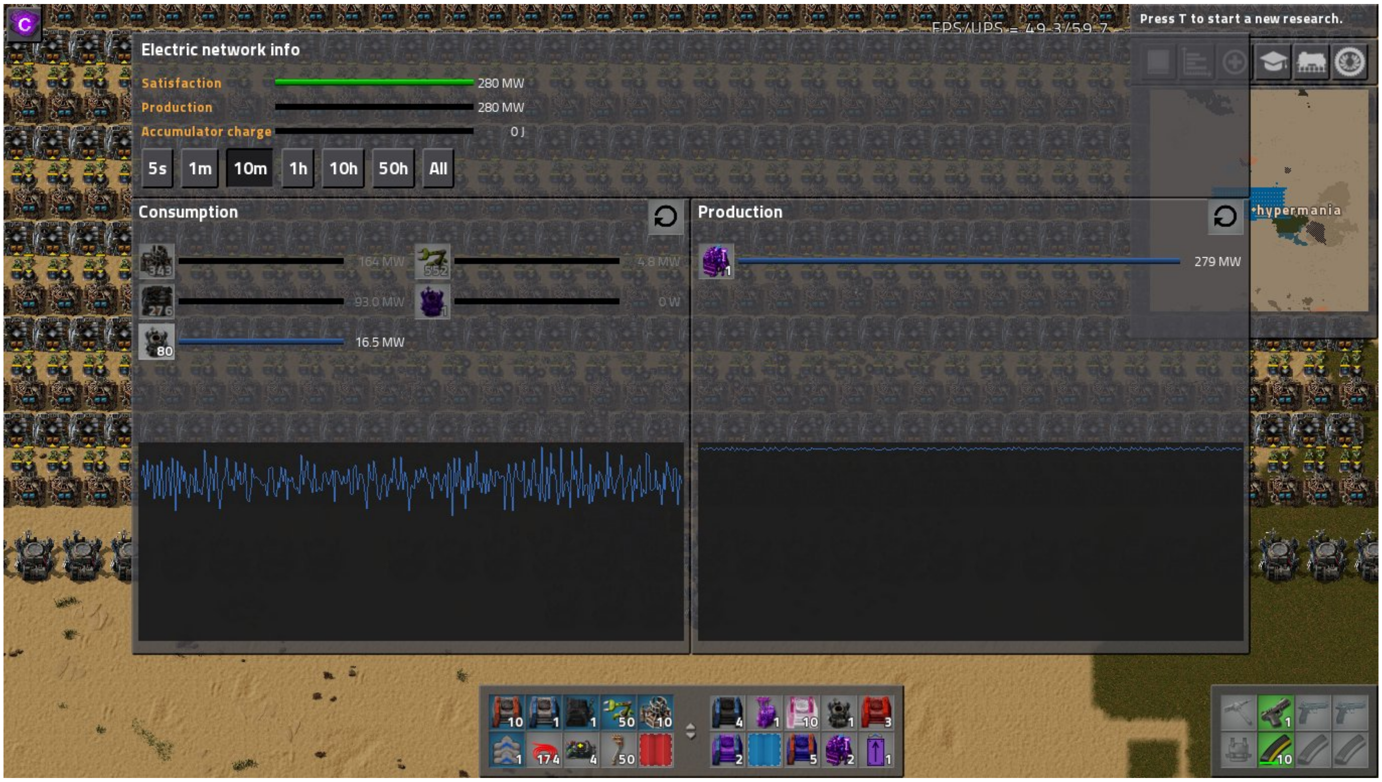

Figure 4: Experiment 2 result

If the throughput goal for requester j is T_j, then we have constraints:

$$T_j = \sum_i T_{ij}. \tag{31}$$

Also, let the maximum throughput for provider i be S_i, then we have:

$$S_i \geq \sum_j T_{ij}. \tag{32}$$

Assuming that the total throughput of all providers is enough for fulfilling all logistic requests over time, given T_j's and S_i's, how do we compute T_{ij}'s? The difficulty here is that we don't know the order in which the requesters make the logistic requests. At a given tick, when a requester makes a request, it simply takes items from the closest provider; when another requester makes a request after the first one, it may have to take items from other providers, because the first provider may be already emptied. If we have the order in which the requesters request items, the bot trips becomes deterministic , and (in principle) we can compute the T_{ij}'s.

Now let us consider the robot trips over a time span of t. The maximum number of items provided by provider i is $N_i := S_i t$, and the number of items delivered to requester j is $M_i := T_j t$. It takes a total of $N := \sum_j T_j t/s$ trips for all logistic requests to be fulfilled. Let the order in which the bots make these N requests be σ, then the total number of deliveries D_{ij} from provider i to requester j is determined. We write $D_{ij}(\sigma)$ to emphasize that D_{ij} is a function of σ.

In the course of the game, the order σ gets randomized, so the expected value for D_{ij} is:

$$\mathbb{E}(D_{ij}) = \frac{1}{N!} \sum_{\sigma} D_{ij}(\sigma). \tag{33}$$

The average throughput can be thus computed via:

$$T_{ij} = \mathbb{E}(D_{ij}) s/t = \frac{s/t}{N!} \sum_{\sigma} D_{ij}(\sigma). \tag{34}$$

We expect T_{ij} to reach a limit when we take $t \to \infty$. With this Equation we can compute the power consumption of logistic robots with Equation 30.

Equation 34 is hard to be used directly. We suspect that there are ways to approximate this result using the combinatorial and statistical structure of the problem, but we won't try that in this article.

3.2 Example: 2 Requester and 2 Providers

Figure 5 shows the setup for experiment 3. The two providers and the two requesters are 16 tiles apart horizontally, and the vertical distance between providers and requesters is 16 tiles; also, the provider and requester near the middle are 16 tiles apart horizontally. All chests have throughput of exactly 1 stack inserter.

To compute the power consumption, we can directly apply Equation 34 with $N = 2$: if left requester requests first, then right requester will have to request from right provider, so the total bot distance is $16\sqrt{5} \times 2$; if right requester requests first, then left requester will have to request from right provider, so the total bot distance is $16(\sqrt{10} + \sqrt{2})$. Assuming the same s and v as in Section 2, we now plug these into Equation 30 to get:

$$S = 12/(26/60) \tag{35}$$

$$P = CS\frac{1}{2}((16\sqrt{5} \times 2) + (16(\sqrt{10} + \sqrt{2}))) \tag{36}$$

$$= 5594.29\text{kW}. \tag{37}$$

The result of experiment 3 is given in Figure 6: the actual power consumption was $P = 8.1\text{MW} - 0.05\text{MW} * 81 = 5.55\text{MW}$. This is very close to our theoretical value. Note that this experiment may not be a strong confirmation for Equation 34, since the difference in total distance between the two orderings is rather low.

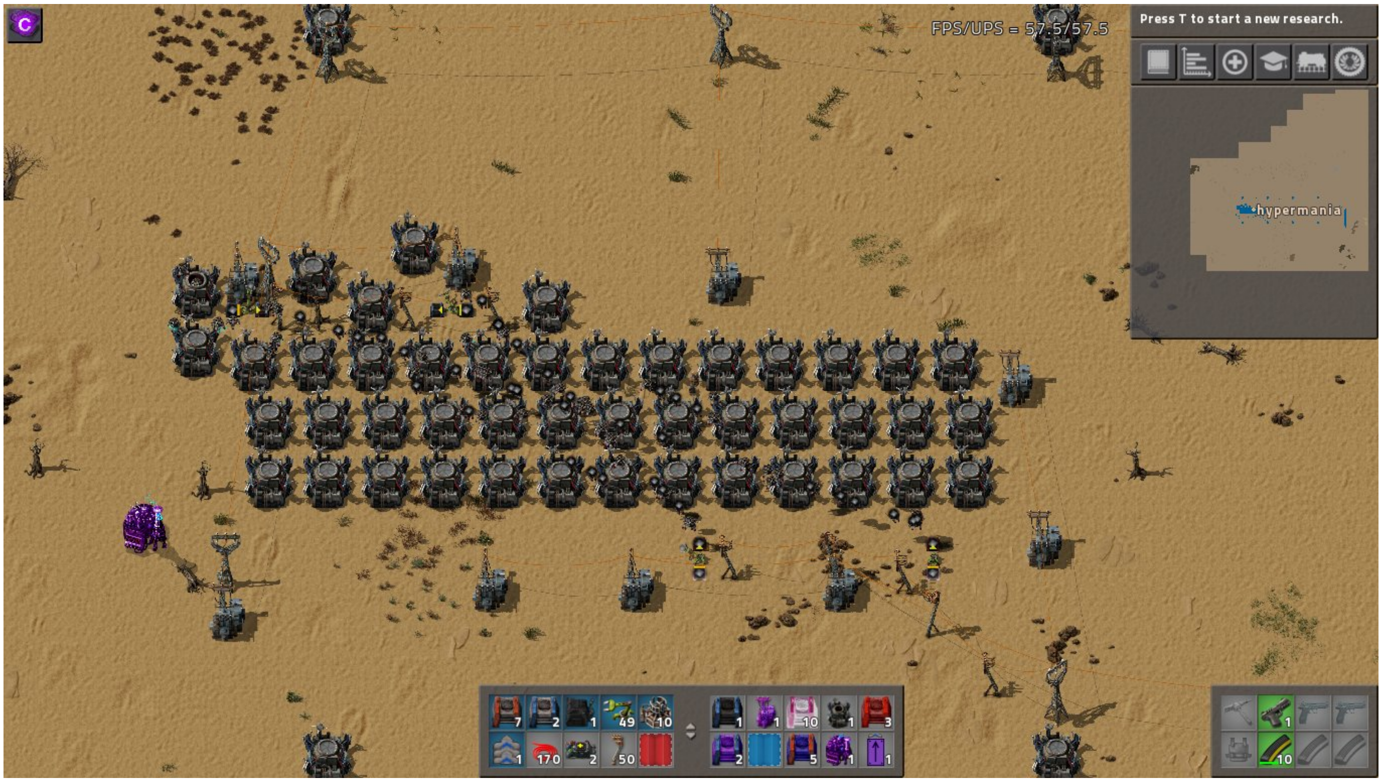

Figure 5: Experiment 3 setup

3.3 Example: Loading Iron Plates from Smelter

Figure 7 shows the setup for experiment 4. In this experiment, we made a 8-beacon iron smelting site with 5 rows and 108 columns. A total of 108 requester chests are directly below the provider chests, and the iron plates in each requester chest is being consumed by a yellow belt. (So $T_j = 13.333/\text{s}$.) The provider chests and requester chests together form a rectangular grid of 6×108. The lowest row of provider chests are 8 tiles from the row of requester chests, so the distance between a requester chest and the provider chests directly above it are 8, 16, 24, 32, 40 tiles.

For this example, we do not actually compute the power consumption using Equation 30 and 34 (because we don't know how to do it), but rather give a lower bound on the power consumption. Assuming that each requester chest only takes items from provider chests directly above it, then to reach throughput goal $T_j = 13.333/\text{s}$, it must take from $T_j/S = 13.333/(1.2 * 9.4/3.5) = 4.13712$ provider chests; the bottom 4 rows of provider chests are thus drained completely, and items will be taken out of the top row of provider chests occasionally.

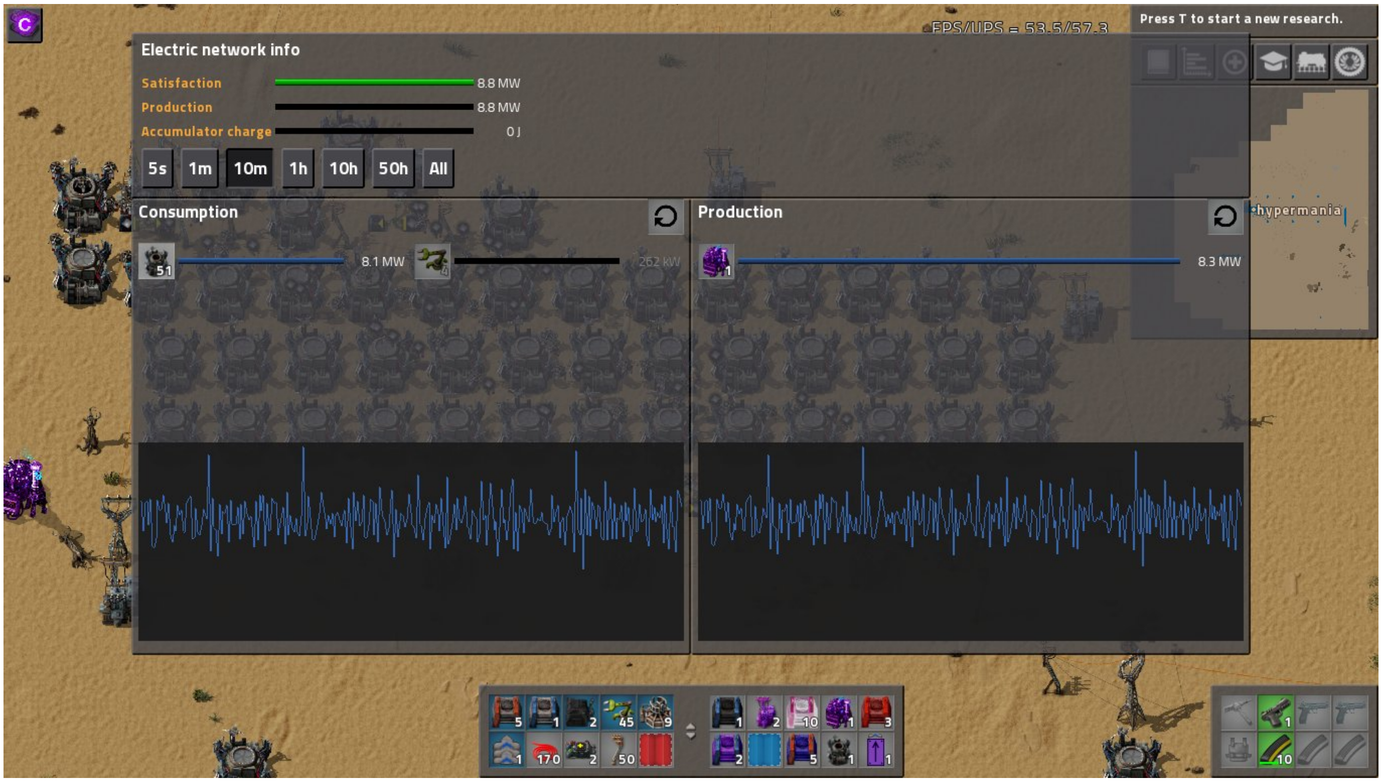

Figure 6: Experiment 3 result

The power consumption for a single requester chest is thus:

$$P_j = C(S * (8 + 16 + 24 + 32) + (T_j - 4 * S) * 40) = 768.852\text{kW}. \qquad (38)$$

The total power consumption thus computed is:

$$P = 108P_j = 83036\text{kW}. \qquad (39)$$

In reality, the requester chests do not request only from providers directly above it. That said, to satisfy the iron plate consumption of all requesters, all 4 bottom rows plus a bit of the top row of providers will be drained. The shortest distance from a provider to the row of requesters is distance between the provider and the requester directly below it. Any delivery that does not follow a vertical path will result in a total power consumption higher than P, so we see that P is indeed a lower bound on the power consumption. It is also not hard to find an ordering σ such that Equation 30 gives the power P we just calculated.

The result of experiment 4 is given in Figure 8; the actual robot power consumption is $P = 103\text{MW} - 0.05\text{MW} * 170 = 94.5\text{MW}$. This result is higher than our lower bound 83036kW by 14%, which is pretty much expected. We see that the lower bound is actually not too far from the actual value, so it can still be a useful estimate for planning purposes.

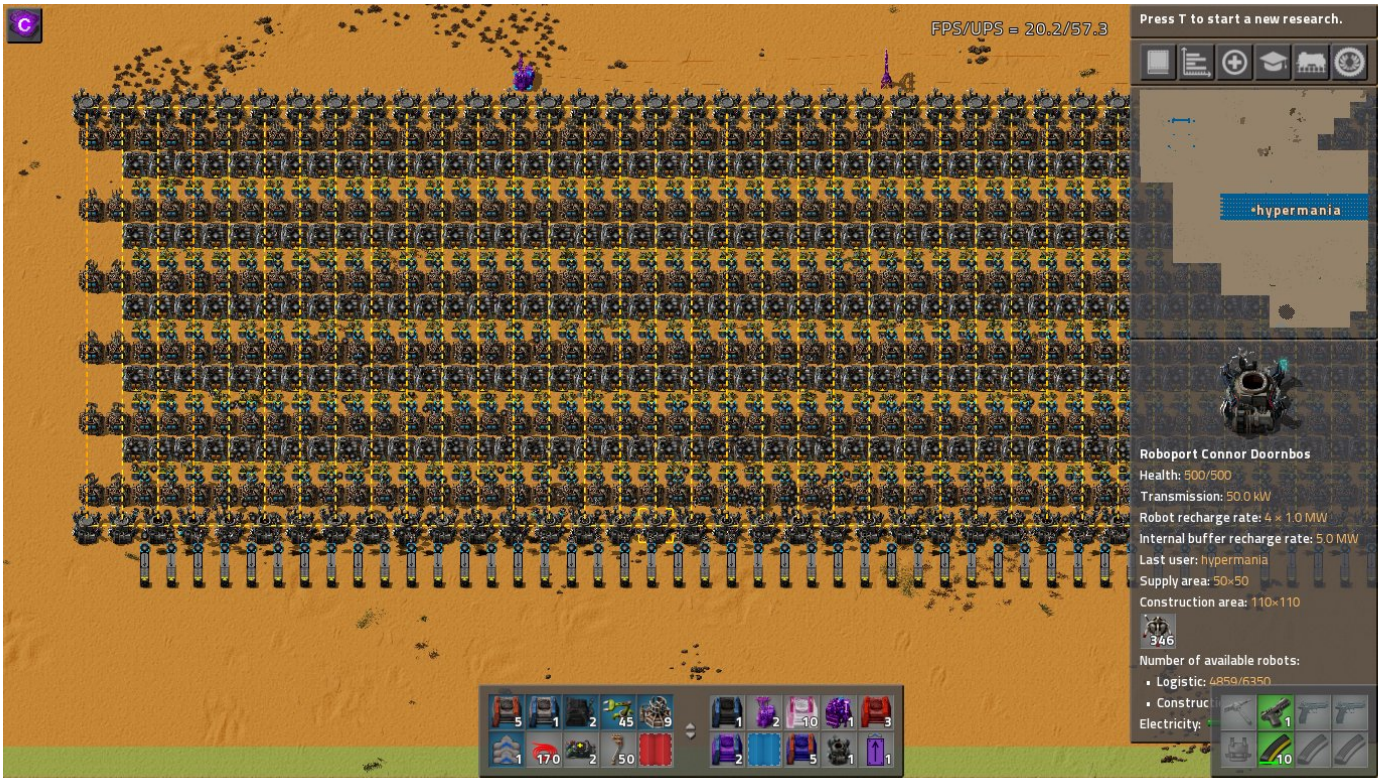

Figure 7: Experiment 4 setup

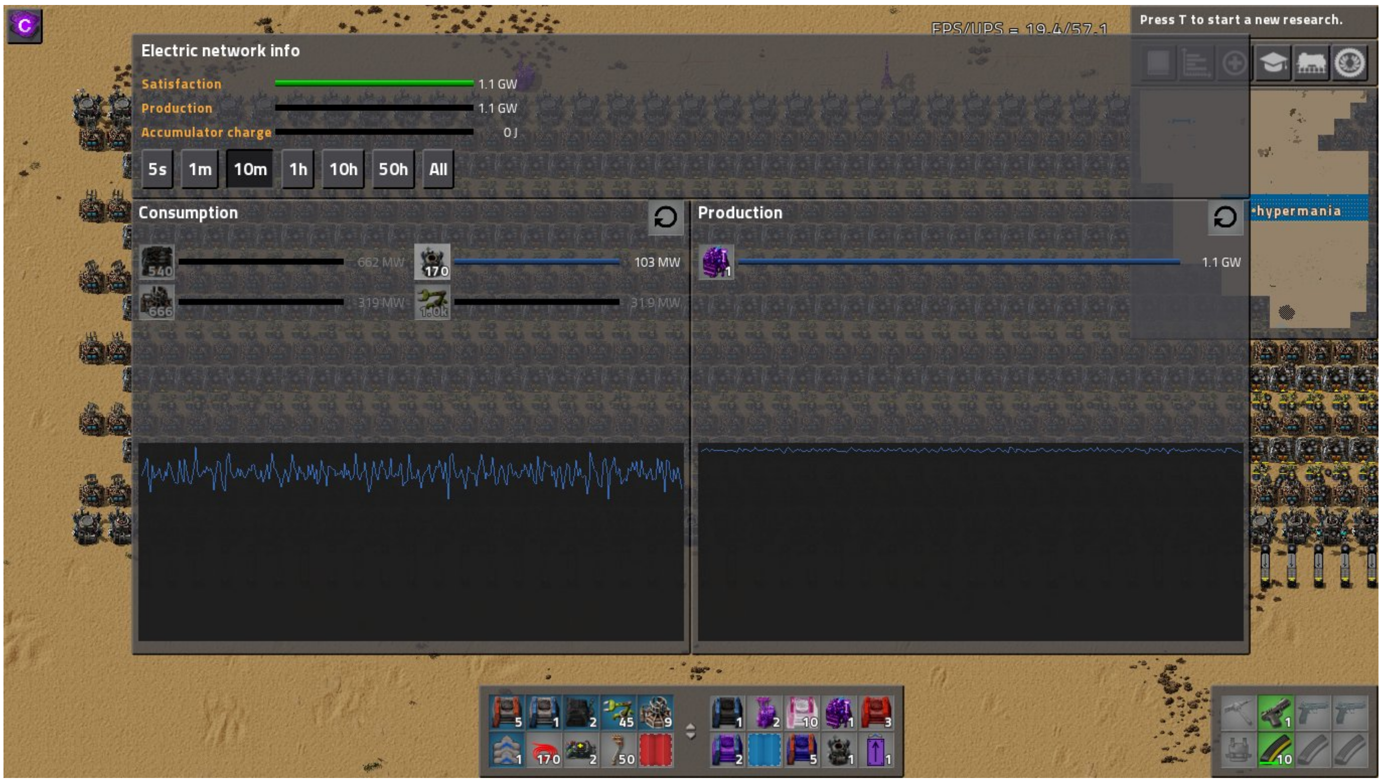

Figure 8: Experiment 4 result