Improving Drone-based Parcel Delivery in a Delivery System at Its Capacity Limit

Niclas zum Felde, Michael Köhler-Bußmeier, and Jan Sudeikat

HAW Hamburg,
Berliner Tor 7, 20099 Hamburg, Germany
niclasjulius.zumfelde@haw-hamburg.de,
michael.koehler-bussmeier@haw-hamburg.de,
jan.sudeikat@haw-hamburg.de

Abstract. In this contribution we explore how a cyber physical system of drones can be used for the delivery of parcels. The basic question is: How can such a delivery system, that is congested, be improved in terms of its delivery time? To find an answer, we model a system like this using timed Petri nets. Simulations are then carried out using this model where the capacities of the system and the capabilities of some drones are manipulated. These simulations show that a small population of improved drones produces similar improvements in delivery times as doubling the charging capacities at saturated points in the system.

Keywords: Cyber physical systems · Petri nets · Simulation · Drones

1 Introduction

The term cyber physical system (CPS) characterises a system that integrates software and physical processes. Here, a physical process is monitored and controlled by embedded, interconnected computers. Typically there is an interdependence and the physical process influences the computations performed by a software component [16]. Unlike conventional embedded systems, however, a CPS focuses heavily on the connectivity of different devices [16]. The services offered and information collected by a CPS should always be available.

There are different factors that favour the trend towards CPS. Powerful sensor technology is becoming increasingly cheaper and smaller in form factor. The possibilities of wireless networks have increased rapidly in recent years and the emergence of alternative technologies for energy generation and storage are factors [18] as well. There is a great demand for CPS, as they can be applied in various different areas. For example, in health care [11], aviation [20], energy grids [4] and in industrial applications [13].

Logistics offers great potential for the use of CPS too. The industry is already considering the use of drones for the delivery of parcels [1]. It is hoped to achieve faster delivery times and to better satisfy customers through this service. In addition, drone delivery also opens up other products for shipping. Groceries are particularly time-sensitive due to their perishability [7]. While there are efforts by retail chains to deliver groceries to their customers, they are not profitable due to the last mile problem [24].

Key questions that arise when using drones in logistics are the distribution of warehouses and charging stations in the area of operation and what types of drones are used. The difference can be, for instance the battery size of a drone. The impact of the topology of a drone network is also an interesting aspect. When using such a system with different topologies for charging stations, the outcomes may vary.

Our general research interest is the adaptivity of distributed cyber physical systems. Therefore, we are particularly curious about strategies that can be applied when a delivery system of drones is saturated. How can delivery times be reduced under these circumstances? The two most obvious parameters are the capabilities of the drones and the capacity of the charging stations. To examine this, we perform a simulation using timed Petri nets [2].

With this aim in mind, this paper is structured as follows: Section 2 presents literature on other analyses of drone-based delivery systems conducted using simulations. It also presents and justifies the use of timed and coloured Petri nets to analyse issues of logistics. The designed model is presented in Section 3. Exactly how it is parameterised for the experiments is then explained in Section 4. The results of these experiments are examined in Section 5 and the paper closes with a conclusion.

2 Related Work

Initial considerations on the use of drones in parcel delivery have been around since 2013 [1, 19]. Since then, various aspects of these systems have been examined. There are limiting factors for parcel delivery by drone, like flight time or parcel weight. To overcome these limitations, the concept of a modular drone was proposed. Modularisation makes a drone more flexible because parts are interchangeable, for example batteries, propellers, motors and parcel holders. This concept was proposed by Lee and investigated using simulation [17]. Two delivery systems were compared, one using modular drones and another using regular drones. The result shows that the use of modular drones can reduce delivery times and energy consumption [17].

Petri nets can be used to simulate such a system. Particularly in the field of logistics, a wide variety of experiments are carried out in this way. In the recent past, processes for transporting cotton by train [6], the flow of goods in a warehouse [8] or the logistics of resources for steel production were analysed using Petri nets [25].

A Petri net is very suitable for modelling because of its clear formalisms and structure. One way of using Petri nets for simulation on a model is also described in the work of Strümpel [22]. Using timed Petri nets a model is designed in an object-oriented way with nested nets [14, 5, 23]. Subsequently, this model is used in RENEW [15] to analyse it for the waiting times of trucks at a loading bay [22].

Drones are much more limited in their range than delivery trucks, a modern drone can fly an average of 4 km [21]. In order to still be able to serve a large area with the delivery system, charging stations need to be distributed in the service area. A method to optimise the placement of charging stations was developed by Hong et al. The resulting algorithm is able to plan the placement of charging stations while taking into account obstacles in the path of drones, e.g. airports and tall buildings [12]. The resulting topologies look very similar to the spanning trees of a path graph used for our model.

In order to represent the concurrency of our drone system, we use Petri nets. We design our model in a similar way to the one described by Strümpel [22]. The model will include different components like the drone, the warehouse and the charging station, so an object-oriented modelling approach is necessary. This can be realised with Petri nets by nesting nets within nets [14, 5, 23]. To simulate our model we use RENEW as well. It allows us to use time-constrained formalisms and to use additional user-implemented Java classes, e.g. for statistical purposes [15].

3 Model

As the first step, the application domain is structured following the Smart-Grid-Reference-Architecture (SGAM) approach [10]. This architecture is intended for the use in energy grids. However, since smart grids are also part of CPS and the architecture can be generalised, it is possible to apply it to other application areas of CPS [9]. This approach allows us to derive the systems context and the involved components. To achieve this the architecture divides the system into five interoperability levels.

- In the **component layer** the individual components of the system are represented.
- The **communication layer** shows the technologies used for communication and in which direction communication between system components flows.
- What information is exchanged between individual sub-systems is described in the **information layer**.
- The actual business processes are mapped in the **function layer**. Use cases are utilised as a methodology for this.
- The **business layer** shows the business case of the CPS from a strategic point of view.

Each level has two dimensions, the *domains* and the *zones*. Domains represent the different localities involved in the process for the drone delivery system, i.e.

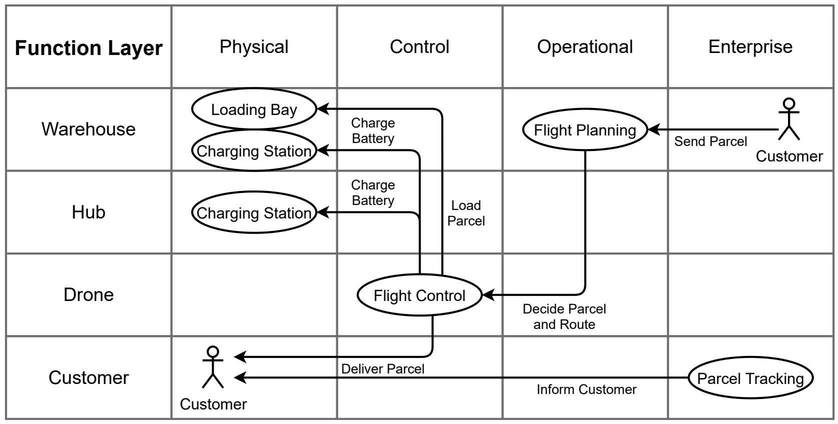

Fig. 1. The function layer of the delivery system

the central warehouse, the stations where drones are charged (hubs), the drone itself and the customer. As shown in Figure 1, the zones are then divided into the physical zone, the control zone, the operational zone and the enterprise zone.

These zones each cover a sub-area of the delivery system, e.g. the physical level in Figure 1 contains the physical functions that are provided. The warehouse has been modeled in a way where it directly plans the flight route for received parcels because it knows all hubs. The drones then receive a flight plan along the hubs together with a package. Where a drone charges its battery, however, is decided by its control system, referred to as the flight control in Figure 1.

This architecture is mainly used to create an overview of the delivery system and the context in which it is used. Nevertheless, conclusions can already be drawn from the architecture for the modelling of Petri nets. Especially the function layer shown in Figure 1 is used for this purpose. The planning of flights is done by a planning system at the warehouse. The drone carries the predetermined flight out independently and decides when to charge its battery. A drone can charge at the warehouse or at a hub. Hubs and the warehouse have limited charging stations, but the warehouse has an unlimited capacity of landing spots. If a drone wants to charge at a hub that has no capacity, it must wait. Only one central warehouse exists in the model, there parcels can be picked up by drones.

Units are then selected for the simulation. In one elapsed time unit in RENEW, a drone can cover one way unit. For a regular drone, this consumes one energy unit. Charging one energy unit also takes one time unit. However, these two parameters are adjusted in the simulations to simulate adding more efficient drones to a congested delivery system. For modelling in RENEW, the model splits into five nets, taking the *warehouse*, the *hub* and the *drone* from the domains of the architecture. The customer is present in the *hub net*. In addition, the flight planning system is mapped to a *path generator net*. Since many simulation runs with differently set parameters have to be started for the experiments a fifth net is used. This *master net* works in conjunction with the four nets that form the model of our delivery system. It has the sole task of starting simulation runs with the correct parameters. An overview of the model is shown in Figure 2. The boxes represent the individual nets of the model, and the arrows are used to illustrate the relationships between the nets. The nets themselves can be seen in detail in Appendix A.

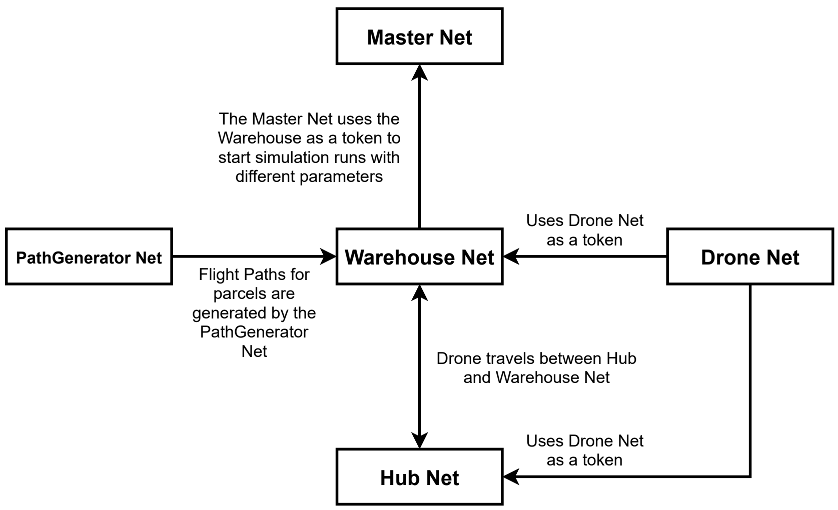

Fig. 2. An overview of the model

3.1 Warehouse

The warehouse net is the starting point of the model, here all drones are instantiated. They then move through the warehouse as tokens. At the centre of the warehouse is a landing pad for drones, into which a drone moves after it has been instantiated. From here, the drone can go to a charging station where it can charge its battery. The capacity of these charging stations is limited.

Secondly, a drone that is not loaded with a parcel can go to a loading bay, which are also limited in capacity. There the drone loads a parcel and receives a flight path determined for it by the path generator and moves back to the landing pad. A loaded drone can take off from there into the hub net to deliver its parcel to the customer. On the other hand, drones that have delivered a parcel can return to the warehouse from the hub net and land at the landing pad.

3.2 PathGenerator

In the warehouse, flight paths are determined for each parcel, which the drone will move along to deliver the parcel. This system is simulated by the path generator net. A flight path is a tuple consisting of an id and two lists. The first list represents the route that still has to be covered for the flight path to the customer, the second represents the route that has already been covered. Using the second list, a drone can then navigate from the customer back to the warehouse. The entries in both lists are tuples consisting of the id of the next hub and the distance from the current hub. Customers are represented as hubs with the id zero.

A topology file is used as the basis for the topology of the hub net. In this file, the hub net is described as a tree. This approach is chosen because we presume that our hubs are placed in the optimal location for our service area. Over the graph of all connections from hub to hub and warehouse to hub we build a spanning tree. This tree then becomes the topology of our delivery system.

The topology of the delivery system is also a parameter that can be set. RENEW offers the possibility to import user-developed Java classes and to use their methods. The actual generation of the paths takes place in a Java class developed for the net. The resulting list then travels through the path generator net as a token and is converted by it into the form of a flight path described above. This is then combined in the warehouse net to a token with a parcel and waits in the warehouse for a drone to load the parcel and complete the path.

3.3 Drone

All actions that a drone can perform are modelled by the drone net. The net travels as a token through the warehouse and hub net. Internally, the drone net has space for a parcel. It delivers the parcel according to the flight path that is associated with it. Like shown in the architecture, the drone itself decides when to charge its battery. The default strategy is to charge only when the next route cannot be covered with the available energy. Drones that arrive at the warehouse are always charged so that they need to charge as little as possible when flying to a customer. The energy that a drone consumes during the flight is modelled using this function:

$$bat_t = bat_{t-1} - d \cdot (1 + \alpha) \tag{1}$$

The energy level of a drones battery at time t is given by bat_t, this is calculated from the energy level at a previous time $t-1$ minus the distance d travelled. The effect that the weight of a parcel has on the range of a drone is given by α. For the simulation, the weight of the parcels is set to five weight units. Our α is then modelled as:

$$\alpha = p/10$$

Where p is the weight of the parcel indicated. For a parcel with 5 weight units, this results in a value of 0.5 for α. This value is provisionally used for the simulation. Of course, α can be changed with more data so that it corresponds to the real effect of additional weight on a drone.

3.4 Hub

The hub net represents the system of charging stations in our delivery system. As described in Section 3.2, the topology of the charging stations is determined by a topology file. In order not to design a new hub net for each topology it is parameterised. For this purpose, the hub net holds the states of the respective hubs in tuples consisting of the hub's id and the capacity of charging stations at that hub. The place hub only represents a generic hub; it is only when its battery has to be charged that it is relevant to a drone at which hub of its flight path it is located. This is due to the capacity of charging places at the current hub, which must be checked and a drone may have to wait for a free spot.

By launching at the warehouse, a drone enters the hub net. To simplify the modelling, a drone cannot fly directly from the warehouse to a customer, but must go to at least one hub. At a hub, a drone can then fly to the next hub, charge its battery if needed, or fly to a customer and unload its package. If a drone has to fly to the customer next, it recognises this by the hub id zero in its flight path, which is reserved for a customer. When a drone has dropped off its parcel at a customer, it follows its flight path in reverse order. If there are no entries in its flight path, the drone moves to the warehouse and enters the warehouse net.

4 Experiments

In our experiments, a situation where a delivery system of drones is under heavy load and its hubs are close to their maximal capacity is recreated [12]. In such a situation, there are two ways to ensure that parcels can still be delivered quickly. The performance of the drones or the capacity of the hubs have to be increased. Both are not always possible, so we compare which is the better strategy. During the creation of the model, many parameters were made variable. For the comparability of the experiments, a set of parameters is chosen from these to remain variable, the rest will stay constant. Adjustable parameters for the study are:

- **Topology:** Two different topologies for hub placement are tested. A broad and a deep tree.
- **Population share:** This parameter describes the share a second type of drones has in the total drone population. This second drone type is considerably more powerful than regular drones.
- **Charging:** The charging speed specifies how fast energy can be absorbed by a drone. For a regular drone, one unit of time is equal to one unit of energy charged.
- **Energy usage:** The energy usage indicates how much energy a drone uses during flight. Regular drones consume one unit of energy for one unit of distance travelled.
- **Hub capacity:** Indicates how many charging stations are available at a hub.

The charging strategy and package size remain static for our experiments. Drones implement a charging strategy in which they charge their entire battery whenever they do not have enough energy to complete the next section of their journey or when they arrive at the warehouse. This should not have much effect on the generalisability of the experiments. Other charging strategies will achieve this goal more optimally, but the aim remains the same. The maximum range is calculated according to the formula for the energy consumption of a drone given in Section 3.3. For the selected parcel size, the maximum loaded range for the drones is 95.23 way units.

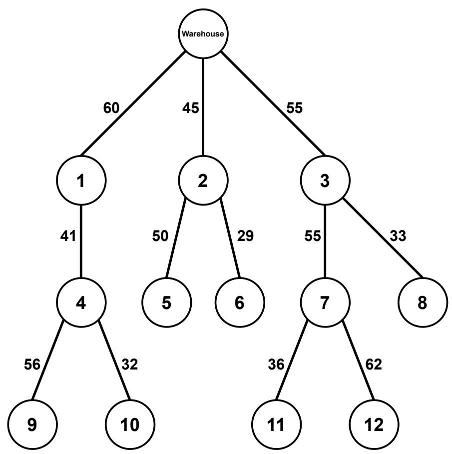

Fig. 3. The topology of a broad delivery system (Topology 1)

Hubs are shown in both Figures 3 and 4 as nodes of each tree and are labelled with their id. The edges represent the flight routes between hubs, the number indicates how many way units lie between both hubs. Customers are modelled as hubs with an id of zero and can be approached from any hub except the warehouse at a distance less than half the maximum range of a regular drone. The probability that a drone is already at the correct hub and must fly to a customer next is determined during path creation. For each node of the tree, it is checked whether it is the last one before a customer.

The two topologies were chosen in order to obtain a broad and a deep tree. Deep or broad are the two dimensions a tree can have. Why trees are used to represent our topologies was explained in Section 3.2. Of course, many more topologies than just these two are possible. But the computation effort to support randomly created topologies is very high, as for each topology all tests using the same parameters have to be performed again. In order to be able to generalise the results, the two different topologies were designed. Nevertheless, other topologies may have different effects and results.

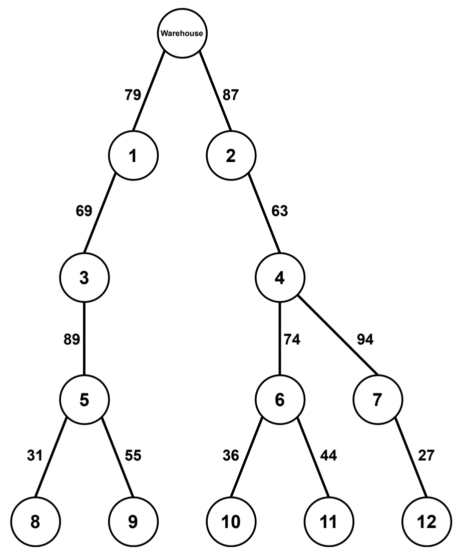

Fig. 4. The topology of a deep delivery system (Topology 2)

The charging speed of regular drones, as described in Section 3.3 , is one-to-one with the time a drone spends at a charging station. For the more powerful drones, this value becomes faster by a factor of 20%, 25% or 30%. To implement this the function for calculating the charging time is supplemented by the factor ct.

$$t_{charge} = (cap - bat_t) * (1 - ct) \tag{2}$$

Here, t_{charge} is the time until the battery is fully charged, this results from the capacity of the battery (cap) minus the current charge level of the battery (bat_t) and is then multiplied with the speed factor ct. The factor ct must be selected between 0 and 1, for $ct = 0$ there is no improvement in charging speed.

The values for improvement are chosen in such a way that at the lower limit a clear improvement can already be observed when charging a completely empty battery from 100 time units to 80 time units. The upper limit is chosen to show that the more powerful drones use similar battery technology that is more efficient but does not rely on a completely different method, such as modular batteries that are only swapped at the hub.

As described above, a drone without cargo consumes one unit of energy for one unit of travel. A loaded drone consumes more energy because the weight of the package being transported is included in the calculation (Section 3.3). This factor is changed for the more powerful drones, which, for example, use more efficient motors or rotor blades to generate lift. This allows for the more efficient drones without cargo to use 20%, 25% or 30% less energy units to fly a path unit. For loaded drones, the corresponding factor is taken into account.

$$bat_t = bat_{t-1} - (d \cdot (1 + \alpha) \cdot (1 - con)) \tag{3}$$

In the formula shown above, you can see the modification to the formula from Section 3.3 for the energy consumption of a drone. The factor *con* indicates how many percent less energy is consumed during flight. A drone that is 20% more energy-efficient thus only consumes 0.80 energy units to fly one way unit.

In all experiments, at least 400 drones are used. This limit was experimentally determined for both topologies as the point at which adding regular drones leads to a worsening in delivery time. The population of drones remains constant over the respective simulation run. However, the proportion of additional more efficient drones is increased over the experiments from zero to proportions of 20%, 25% or 30% of the regular drone population. In a second trial, improved drones replace 20%, 25% or 30% of the regular drone population. To make a comparison, trials are also conducted in which the population of drones is not being changed. Instead, the capacity of two particularly overloaded hubs is increased by 10%, 20%, 30%, 50% and 100%. This allows us to assess whether changing the population of drones or changing the capacity of congested hubs can prevent or reduce congestion in the system and thus reduce delivery times.

Due to our decision to charge all drones arriving at the warehouse, we set the capacity of charging stations there to 50. At this capacity, there will be no queues at the warehouse. This was also determined by experiments. The regular capacity of a hub is 10 charging stations.

5 Results

For the first experiment with topology one, additional drones were added to the system. These were improved as described in Section 4. Results from the first topology show that using additional drones whose charging speed is improved keeps the average delivery speed in the range of 700 to 800 time units, but the delivery speed is not improved compared to a system with only 400 regular drones.

Table 1. Average delivery times for topology 1 with additional drones

Improvement	80 additional drones	100 additional drones	120 additional drones
20% less energy usage	600.84	545.13	530.29
25% less energy usage	486.57	524.30	515.88
30% less energy usage	475.08	488.93	506.86
20% more charging speed	733.77	693.39	720.63
25% more charging speed	607.16	759.34	780.08
30% more charging speed	711.62	705.21	704.75
only regular drones	623.14		

This is different for the trials in which the drones were improved in terms of their energy consumption. In the trials with drones that consume less energy, parcels are delivered in under 550 time units on average. This is a significant improvement over the system without additional drones. Nevertheless, it can be observed here that our delivery system is getting worse on average with more than 480 drones, this applies to two of the three levels of performance improvement. For drones that use 20% less energy, the trend is reversed. This could be a statistical effect, as each test was only performed twice, or it may be an effect similar to the Braess paradox [3].

Table 2. Average delivery times for topology 2 with additional drones

Improvement	80 additional. drones	100 additional drones	120 additional drones
20% less energy usage	1878.00	2208.52	2228.15
25% less energy usage	1934.17	2248.71	2221.46
30% less energy usage	2085.81	2199.40	2066.05
20% more charging speed	2095.11	2231.54	2178.96
25% more charging speed	1965.77	2223.34	2249.74
30% more charging speed	2085.41	2015.05	2156.89
only regular drones	1761.93		

It is obvious that the delivery times for topology two are significantly longer than for topology one. With 400 regular drones, parcels are delivered in an average of 1761.93 time units. The addition of improved drones does not lead to any improvement. Any additional drone population worsens the delivery times, some even significantly. The difference between drones that charge faster and those that use less energy is not as obvious as in topology one. But even here, statistical effects can be observed, especially for the drones that charge faster. At 80 additional drones with less energy consumption, an effect similar to the Braess Paradox can be observed again. More efficient drones need longer to deliver a package. What caused this effect needs to be researched further.

It seems that additional drones with improvements can only help to a limited extent and the reduction in load is dependent on the topology. However, from Tables 1 and 2, it can be concluded that improving energy consumption is more beneficial than faster charging times.

5.1 Replacing Drones

It may be that the approach of improving the performance of the overall system by adding more powerful drones is wrong in a situation where the system is already operating at its capacity limit. Conceivably much better results can be achieved by replacing some of the regular drones with more capable drones. So that the system continues to have only 400 drones in operation.

For the delivery time of a parcel, Table 3 shows a clear advantage when using improved drones. On average, the systems with drones that consume less energy achieve better results than the systems with drones that can recharge their batteries faster. The mean delivery time for both improvements is better

Table 3. Average delivery times for topology 1 when drones are replaced by more capable drones

Improvement	80 drones replaced	100 drones replaced	120 drones replaced
20% less energy usage	442.54	404.98	342.49
25% less energy usage	447.00	400.37	355.63
30% less energy usage	396.30	363.70	355.35
20% more charging speed	558.14	637.15	520.86
25% more charging speed	574.14	563.14	523.57
30% more charging speed	596.22	492.58	560.86
only regular drones	623.14		

on average than in a system that relies only on regular drones. Compared to the trials with additional drones, replacing drones achieves lower average delivery times for both improvements. However, again some statistical anomalies can be seen in this table. Delivery time increased for drones that charged faster in the trials with 80 and 120 replaced drones. Such an effect was not observed for drones that used less energy. Nevertheless, in a population of 120 improved drones, the improved efficiency of 30% less energy consumption brought only a minimal reduction in delivery time compared to an improved efficiency of 25%. Whether this is a statistical effect or whether this is an effect of Braess Paradox would have to be investigated in further simulations.

Table 4. Average delivery times for topology 2 when drones are replaced by more capable drones

Improvement	80 drones replaced	100 drones replaced	120 drones replaced
20% less energy usage	1647.68	1723.38	1644.47
25% less energy usage	1540.92	1596.02	1477.68
30% less energy usage	1796.90	1652.09	1506.78
20% more charging speed	1726.41	1735.13	1526.29
25% more charging speed	1721.52	1570.33	1641.49
30% more charging speed	1650.46	1608.81	1604.15
only regular drones	1761.93		

For the second topology, statistical effects are once again observed in Table 4, which can possibly be explained by the low number of simulations for each trial. For example, a population of 12 drones charging 20% faster reduces the delivery time more than the same population charging 30% faster. However, this effect could also be an analogy to the Braess paradox. Further research is needed here. Nevertheless, it can be said that replacing drones with a population of improved drones can reduce delivery times. The difference between drones that recharge faster and drones that use less energy is not as pronounced as for topology one. In summary, replacing drones with improved drones brings an improvement in delivery times for our system. This approach is more promising than adding additional drones to a congested system and is a more intuitive strategy.

5.2 Increasing Hub Capacity

The third conceivable strategy to improve delivery times is to increase capacity at the hubs. In Section 4 it was described that this is only done at the two most congested hubs. To do this, we need to determine which ones are most congested. Since the warehouse has enough capacity, the most overloaded hubs are presumably the ones that come after the warehouse. Based on our model design, all drones must fly to at least one of these hubs before flying to a customer. If we look at the waiting times per hub, this assumption is confirmed. For topology one, hubs one and three are the most crowded. In topology two, drones have to wait the longest at hubs one and two. The capacities at these hubs will now be increased for the trials.

Table 5. Average delivery times for both topologies when capacity is increased at congested hubs

Topology	No incr.	10% incr.	20% incr.	30% incr.	50% incr.	100% incr.
Topology one	623.14	550.79	471.36	464.62	381.89	302.78
Topology two	1761.93	1663.98	1646.73	1572.18	1470.63	1388.87

This has an immediate effect. Drones don't have to wait to charge at the hubs where capacity was increased. For topology one, the waiting times are distributed fairly evenly over the remaining hubs. The waiting times in topology two only shift to the level below hubs one and two. Now hubs three and four are congested but noticeably less. This can be explained by the depth of the tree spanned by topology two. These hubs are still in a position where they become bottlenecks. Still, in both topologies, the average delivery time of a parcel decreases.

In the case of topology one, we can see that the increase in capacity at the hubs is reflected in the delivery time. For 10%, 20% or 30% more capacity, the mean delivery time remains greater than using drones that consume less energy. Compared to the faster charging drones, however, the adjustments are advantageous. What is surprising is that a 100% increase in capacity brings only a small improvement over a 30% drone population that uses less energy. We expected a larger difference.

We observed a similar reduction in mean delivery time for topology two as well. The improvement in hub capacity is beneficial for this topology in comparison to a drone population that charges faster. However, compared to drones that consume less energy, increasing capacity by 10%, 20% or 30% is not advantageous. Comparing to topology one, we see a clear improvement with a 100% increase in capacity over drone populations that use less energy. This is more in line with our expectation. But it also shows that there is a dependency between the degree of improvement in delivery times and the topology.

6 Conclusion

This paper presents a model for a drone-based parcel delivery system. We implemented this model using timed Petri nets. Subsequently, we performed simulations with the model to test different strategies for reducing the load on such a system.

Our results suggest that both increasing the capacity of charging stations and replacing regular drones with improved drones are valid strategies. Both approaches can reduce delivery time in a congested delivery system. As an upgrade to the drones, lower energy consumption was clearly found to be better. Faster charging drones reduced delivery time but only marginally. This result is consistent across all of our experiments. Simply adding drones, even if they are improved, to the system has not led to much improvement and in some cases to a degradation, so it does not play a role in the further evaluation.

The question of what can be done to reduce the load in a situation like this, which is important for the operation of such a system, can only be answered with reservations. Our results suggest that by replacing drones with better performing drones, delivery times can be reduced. The reduction is noticeable in comparison to doubling the capacity at particularly busy hubs, as it is greater than expected. In this context, economic considerations and the environment in which the system is embedded will play a key role in deciding. Since a drastic development like this is not always possible.

However, we also observe a dependency between the degree of improvement and the topology of the delivery system in our results. For the experiments, a broad and a deep topology were chosen to represent the two dimensions of a spanning tree. This does not seem to be sufficient to determine if the dependency is only a statistical effect or the topology really has an impact on the size of improvements in delivery time. Further simulation experiments in which the topologies are randomly generated could help shed light on this.

7 Acknowledgements

We would like to thank Torben Christian Schrader for his help in designing and implementing the model of our delivery system.

References

1. Amazon: Prime Air. `https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8\&node=8037720011`, (Accessed: 19 February 2021)
2. Bause, F., Kritzinger, P.: Stochastic Petri Nets – An Introduction to the Theory. Advanced Studies in Computer Science, Vieweg Verlagsgesellschaft (1996), http://ls4-www.cs.tu-dortmund.de/cms/de/home/bause/bause_kritzinger_spn_book_print.pdf
3. Braess, D.: Über ein Paradoxon aus der Verkehrsplanung. Unternehmensforschung **12**(1), 258–268 (1968)

4. Bruinenberg, J., Colton, L., Darmois, E., Dorn, J., Doyle, J., Elloumi, O., Englert, H., Forbes, R., Heiles, J., Hermans, P., Uslar, M.: CEN -CENELEC - ETSI: Smart Grid Coordination Group - Smart Grid Reference Architecture Report 2.0 (01 2012)
5. Desel, J., Reisig, W., Rozenberg, G. (eds.): Advanced Course on Petri Nets 2003, vol. 3098. Springer-Verlag (2003)
6. Dong, X., Wang, Y.: Modelling iot-enabled logistics process adaptations with coloured petri nets. In: Proceedings of the 3rd International Conference on Judicial, Administrative and Humanitarian Problems of State Structures and Economic Subjects (JAHP 2018). pp. 655–660. Atlantis Press (2018/08). https://doi.org/https://doi.org/10.2991/jahp-18.2018.135, `https://doi.org/10.2991/jahp-18.2018.135`
7. Frachtenberg, E.: Practical drone delivery. Computer **52**(12), 53–57 (2019)
8. Gerini, C., Sciomachen, A.: Evaluation of the flow of goods at a warehouse logistic department by petri nets. Flexible Services and Manufacturing Journal **31**(2), 354–380 (2019)
9. Gottschalk, M., Uslar, M., Delfs, C.: The Use Case and Smart Grid Architecture Model Approach. Springer International Publishing (2017). https://doi.org/https://doi.org/10.1007/978-3-319-49229-2
10. Group, C.C.E.S.G.C.: Smart Grid Reference Architecture. `https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf` (2012), (Accessed: 23 March 2021)
11. Haque, S.A., Aziz, S.M., Rahman, M.: Review of cyber-physical system in healthcare. international journal of distributed sensor networks **10**(4), 217415 (2014)
12. Hong, I., Kuby, M., Murray, A.T.: A range-restricted recharging station coverage model for drone delivery service planning. Transportation Research Part C: Emerging Technologies **90**, 198–212 (2018)
13. Jazdi, N.: Cyber physical systems in the context of industry 4.0. In: 2014 IEEE international conference on automation, quality and testing, robotics. pp. 1–4. IEEE (2014)
14. Köhler-Bußmeier, M.: A survey on decidability results for elementary object systems. Fundamenta Informaticae **130**(1), 99–123 (2014)
15. Kummer, O., Wienberg, F., Duvigneau, M., Schumacher, J., Köhler, M., Moldt, D., Rölke, H., Valk, R.: An extensible editor and simulation engine for Petri nets: Renew. In: Cortadella, J., Reisig, W. (eds.) International Conference on Application and Theory of Petri Nets 2004. vol. 3099, pp. 484 – 493. Springer-Verlag (2004)
16. Lee, E.A.: Cyber physical systems: Design challenges. In: 2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC). pp. 363–369. IEEE (2008)
17. Lee, J.: Optimization of a modular drone delivery system. In: 2017 Annual IEEE International Systems Conference (SysCon). pp. 1–8. IEEE (2017)
18. Rajkumar, R., Lee, I., Sha, L., Stankovic, J.: Cyber-physical systems: the next computing revolution. In: Design automation conference. pp. 731–736. IEEE (2010)
19. Research, D.T.: Unmanned aerial vehicle in logistics: A dhl perspective on implications and use cases for the logistics industry. `https://www.dhl.com/content/dam/downloads/g0/about_us/logistics_insights/DHL_TrendReport_UAV.pdf` (2014), (Accessed: 19 February 2021)
20. Sampigethaya, K., Poovendran, R.: Aviation cyber–physical systems: Foundations for future aircraft and air transport. Proceedings of the IEEE **101**(8), 1834–1855 (2013)

21. Stolaroff, J.K., Samaras, C., O'Neill, E.R., Lubers, A., Mitchell, A.S., Ceperley, D.: Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. Nature communications **9**(1), 1–13 (2018)
22. Strümpel, F.: Simulation zeitdiskreter Modelle mit Referenznetzen. Diplomarbeit, Universität Hamburg, Fachbereich Informatik (2003)
23. Valk, R.: Object Petri nets: Using the nets-within-nets paradigm. In: Desel et al. [5], pp. 819–848
24. WirtschaftsWoche: Logistik auf der letzten Meile: Das Liefer-Dilemma. `https://www.wiwo.de/adv/capgemini/maerkte/logistik-auf-der-letzten-meile-das-liefer-dilemma/24034896.html` (Feb 2019), (Accessed: 19 February 2021)
25. Zhao, J., Wang, Y., Xi, X., Wu, G.: Simulation of steel production logistics system based on multi-agents. Int. J. Simul. Model **16**(1), 167–175 (2017)

A Petri Nets of the Model

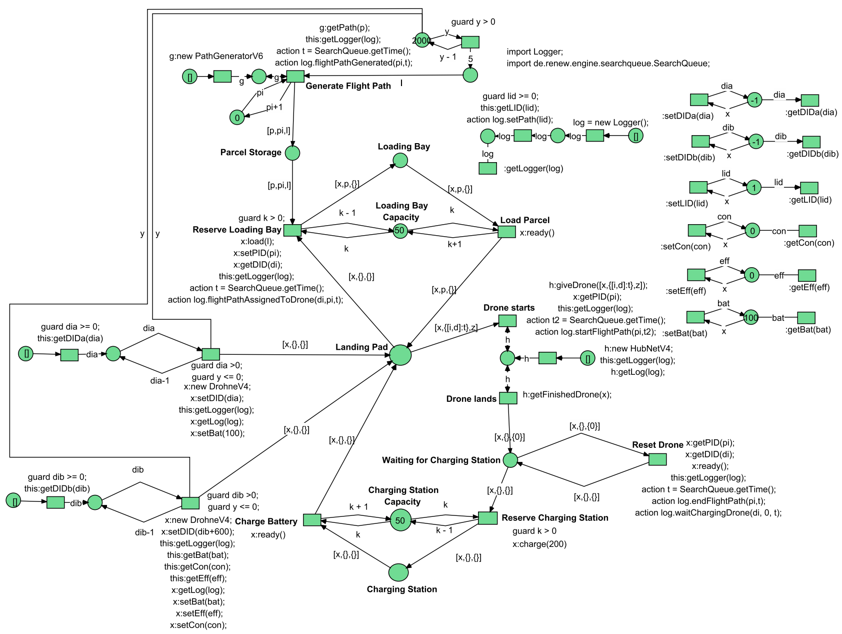

Fig. 5. The Warehouse Net

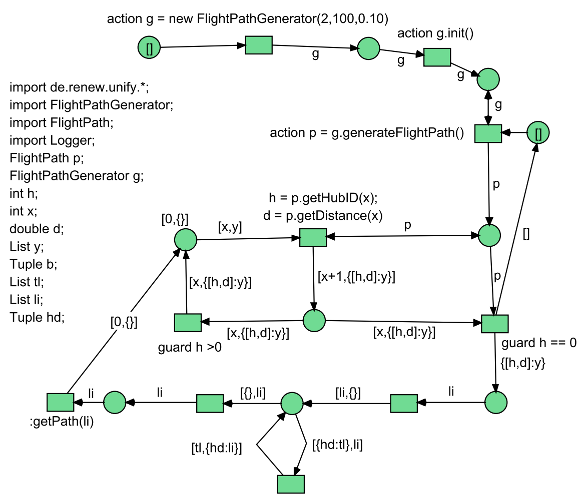

Fig. 6. The PathGenerator Net

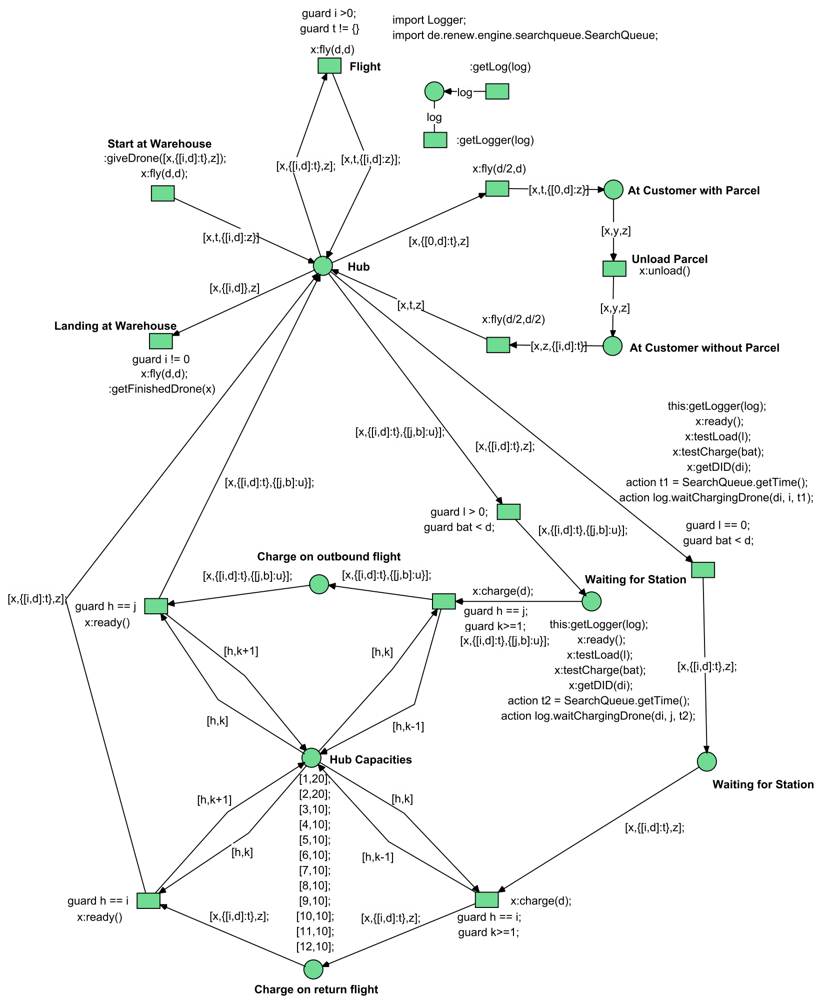

Fig. 7. The Hub Net

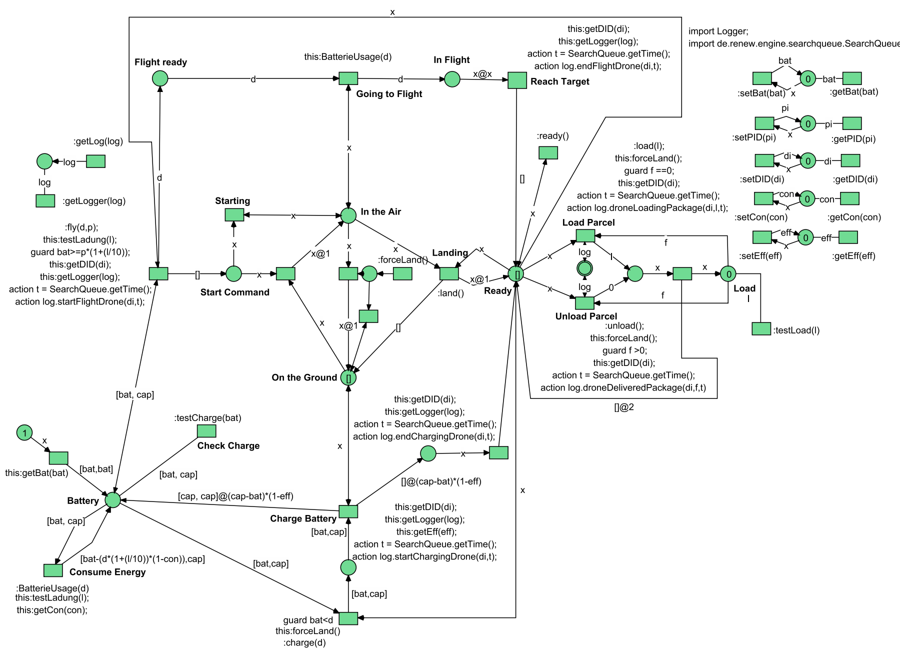

Fig. 8. The Drone Net

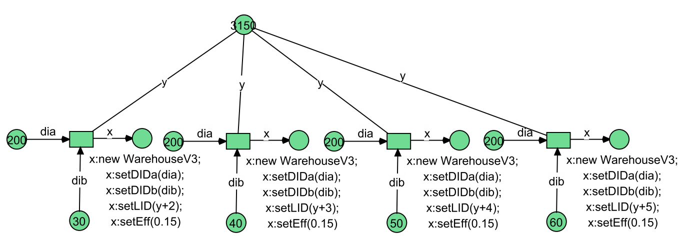

Fig. 9. An example for a Master Net