Smart Courier System

Brandon Simoko
2700870
Middle East Technical University

***Abstract*—This paper presents a discrete-event simulation model of a smart courier system designed to replicate real-world package delivery operations within a small city environment. The model incorporates four primary entities: a central main station, multiple sub-nodes (delivery hubs), delivery trucks, and parcels. The system operates over an 8-hour working day cycle, simulating key processes including parcel arrival and scanning, pickup and delivery operations, dynamic vehicle routing, and truck maintenance events. Utilizing various probability distributions including inverse exponential, uniform Linear Congruential Generator (LCG), and inverse Poisson functions to model realistic event timing and system behaviors along with network communication between nodes utilizes LoRa technology in a star network architecture for data transmission to the central station. The model tracks critical performance metrics including parcel throughput, delivery success rates, truck breakdown frequency, and operational timing across all system components. Experimental results from a four-node system demonstrate that the model accurately replicates expected courier operations, with parcel arrival times, truck travel durations, and repair times falling within predicted ranges of 11-37 minutes depending on the specific operation. System efficiency analysis reveals significant performance variations between nodes, with throughput rates ranging from 39% to 68% for same-day delivery likelihood. The simulation successfully captures real-world operational challenges including vehicle breakdowns, varying delivery success rates, and the impact of system bottlenecks on overall performance. The proposed model provides a valuable tool for analyzing courier system performance without the costs and risks associated with physical implementation. It enables optimization of routing strategies, resource allocation, and system capacity planning while accounting for realistic operational constraints and failure scenarios. Future work will extend the simulation to longer operational periods and incorporate additional transportation modes including drone delivery systems.**

***Index Terms*—Discrete Event Simulation, Omnet++, Flora, IEEE, IEEEtran.**

The code can be accessed on GitHub https://github.com/branrx/CNG476.git.

I. Introduction

The rapid growth of e-commerce has created unprecedented demand for efficient courier systems. Modern consumers expect faster delivery times and real-time tracking, pressuring logistics providers to optimize operations while maintaining cost-effectiveness. Traditional courier systems, with limited network visibility and static routing, struggle to adapt to dynamic urban delivery environments.

Optimizing courier operations involves complex interactions between distribution centers, vehicles, routing algorithms, and communication networks. Factors such as varying parcel volumes, traffic conditions, vehicle breakdowns, and dynamic demands create a highly stochastic environment difficult to analyze using traditional methods. The costs of implementing and testing new strategies in real-world systems can be prohibitively expensive, particularly for smaller logistics companies.

Discrete-event simulation has emerged as a powerful tool for modeling complex logistics systems, enabling experimentation without physical implementation risks and costs. Recent advances in IoT technologies, particularly Long Range (LoRa) communication, offer promising solutions for intelligent courier systems with enhanced monitoring capabilities. However, existing studies often focus on specific aspects like routing optimization, failing to provide holistic models that capture full system complexity including equipment failures and communication delays.

This paper presents a comprehensive discrete-event simulation model of a smart courier system operating within a small city environment. Our model incorporates realistic representations of central hubs, sub-stations, delivery vehicles, and parcels, using various probability distributions to model stochastic events. A key innovation is the integration of LoRa-based communication networks, enabling realistic modeling of information flow between system components.

The primary objectives are to develop a simulation model that accurately represents courier system interactions, demonstrate the model's capability to evaluate performance across multiple metrics, and provide a foundation for future research into courier system optimization and emerging technologies.

II. Methodology

A. System Model

Our model represents a smart courier system operating within a small city. It includes a central hub and several sub-hubs (nodes), all connected by delivery routes. Parcels are transported from their source to destination using delivery trucks. The model simulates key real-world courier operations like parcel arrival and scanning, pick-up and delivery, and vehicle routing.

A typical scenario would be: a parcel arrives at a node, gets scanned, is picked up by a truck, the node generates a delivery route, and the truck delivers parcels to each node along that route. Once deliveries are complete, the truck returns. This cycle repeats continuously over an 8-hour period to simulate a full working day.

Our system model is built around four main entities, objects that represent real-world components within the simulation. These entities each have properties (variables) that define the state of the system at any given time. The four entities in our model are: the main station, sub-stations, trucks, and parcels. Below is a description of each and their properties.

B. Entities

1) Main node: The central station which monitors other sub nodes, it keeps tracks of system analytics within the network, for instance, how many parcels are at a node at a particular time. Updates from nodes are transmitted from a node to a dedicated server.

Attributes

- Number of packages at each location

2) Sub Nodes: A point where parcels are handed in or delivered. These node handles some sub entities like parcels and trucks and lora nodes. Nodes generate parcels along with their properties, generates routes, dispatch trucks and keeps tracks of sub node statistics like parcel arrival count, delivery success rate, truck location. All sub nodes are linked therefore, routing does not depend on node links.

Attributes

- **P**arcel count: number of parcels at any point in time.
- **I**D: identity descriptor.
- **D**elivered to: success deliveries.
- **N**ot Delivered to: failed deliveries.
- **P**arcel vector: a vector of parcels at that node.

Random functions used

- Server update time: inverseExponential, time when node information is sent to the server.

3) Parcels: These are objects that represent actual packages thus have attributes to identify them as such, including origin, destination, weight etc, which are all randomized when package is generated. These are listed below along with descriptions for each attribute. Packages are handed in at a node, while awaiting pick up and delivery, when ready packages are picked up by a truck and delivered to the destination node. On arrival the packages can either be flagged as delivered or not based on an LCG generator function.

Attributes

- **D**estination
- **W**eight
- **D**elivery status

Random fuctions used

- **P**arcel attributes : uniform LCG
- **P**arcel arrival time: inverseExponential
- **P**arcel arrival count: inversePoisson
- **D**elivery status: uniform LCG

4) Delivery trucks: These represent objects that can travel between nodes and act as the only medium between sub-nodes. A truck is defined by properties, including a truck ID that matches the parent node ID. Its attributes are set at initialization like capacity, which defines the maximum load it can carry per dispatch.

A truck always starts at its parent node, picks up parcels, and is then assigned a random route to follow for deliveries. When it arrives at a node, it unloads any parcels addressed to that node's ID, then continues to the next stop.

Trucks don't pick up parcels from any node other than their parent. They can also break down during delivery, which adds repair time. Once the route is complete, the truck returns to the parent node, picks up new parcels, and repeats the process.

Attributes

- **I**D: identity descriptor.
- **D**irection: where it is going.
- **C**apacity: maximum carry capacity.
- **I**D: where it is going.
- **i**sRepairing: is the truck broken down.
- **P**arcels vector: a vector of parcels being delivered.
- **R**oute vector: a vector of nodes the truck is delivering to.
- **i**sStatus: current truck status, 0 idle, 1 en-route, 2 loading

Random functions used

- **C**apacity: uniform LCG.
- **R**oute vector: uniform LCG, order of visit is random.
- **T**ruck repair time: uniform LCG, time to repair upon breadown.
- **T**ruck break down: uniform LCG, describes how frequent break down occur.
- **T**ruck travel time: inverseExponential, time take to travel between nodes.
- **T**ruck loading and unloading time: inverseExponential, how much time it takes to load or unload parcels at a node.

C. Events

These are activities that are scheduled to happen at a certain instance in time within our simulation. Once triggered, these trigger events which happen instantly or are scheduled to occur later in time.

Events in our system include:

- A parcel being handed in at a node.
- A parcel delivery status.
- A truck arriving at a node.
- A truck starting to load or unload at a node.
- A parcel being delivered.
- Truck breakdown.

D. Activity (Unconditional Wait)

These are processes which happen over a period of time, therefore to model this we have to simulate amount of time the process takes until it triggers an event. For example, a truck traveling between nodes is an activity, and it arriving at the intended destination node is the event triggered after the said event.

Activites in our system include:

- Time till next parcel arrival.
- Time for truck to travel between nodes.
- Time taken for truck to load or unload.
- Time take for truck to be repaired.
- Truck idle time, while waiting for parcel arrival.

III. Network Architecture

For network communication between nodes we utilized Flora which is a long-range transmission, this acts as the only form of communication between the main node and sub nodes. Given this, we implemented a star network architecture where all nodes are directly connected to a Gateway.

Event	Count	Time	Bias	Expected Values
Parcel arrive	4	1700	0.05	15 to 37 mins
Truck repair	2	400	0.01	12 to 22 mins
Truck travel	4	2700	0.1	11 to 33 mins
Truck unloading	2	500	0.05	10 to 20 mins
Truck idle	4	1700	0.05	15 to 37 mins
Server update	4	1700	0.05	15 to 37 mins

TABLE I
EXPECTED TIMES OF EVENTS

Messages are transmitted as packets, thus we prepare a string containing for instance the id of the sending node and the type of message, along with a value an example would be " node:1, parcelCount:41 ", which identifies origin as node 1 and parcels at node 1 is 41. This message is sent to the gateway, which by default routes the packet to the server using a wired connection and an ipv4 routing table. The main station receives the message and stores it.

IV. Metrics of Evaluation

To evaluate the performance of our simulation, we consider the following: accuracy of generated events and correctness of the model based on the actual objects modeled. In the following subsections we describe these 2 techniques.

A. Accuracy of Generated Events

Generating events is a delicate process, thus modeling the probabilities of each intended entity or attribute of an entity is critical. Given that in our case we have numerous types of events and time periods which are associated with real events, it is critical to make sure the values generated are according to the expected values. Thus for each mean value we test whether the ranges are in accordance with the expected values.

In Table I, we present the mean values for most of the random events we have in our system. Where Count and Time are used to compute the lambda which is then used to compute the inter-arrival time using inverse exponential distribution. In addition to this formulation we added a custom bias variable, which is used to make emphasis on large values rather than small ones. For instance, when drawing from a large distribution with $min = 5$ mins and $max = 3$ hrs, it won't make sense to say a that truck can take 5 minutes to repair or 3 hours, therefore the bias makes sure if the range is massive the top most is more likely to occur, slightly ignoring the uniform distribution properties.

B. System efficiency

In addition to the accuracy of the modeled system, we also wanted to evaluate the performance of our model based on the output of each entity, in terms of how they replicate their real-world counterparts. In this case we measure metrics like how many parcels are received at a node, how many are delivered (successfully or not), how many times a truck breaks down, and total time taken to repair. We also use an aggregate of multiple attributes, for instance measuring the throughput of a node shown in Equation 1, defined as how many parcels are received and how many of those are delivered within a working day. Also, we measure how many parcels are delivered successfully as presented in Equation 2. Note that equation variables are defined in the footnote 1.

$$Throughput = TPD/TPR \tag{1}$$

$$DeliverySuccessRate = PSD/TPD \tag{2}$$

1

V. Results

In this section we present the results of our simulations. Starting from initialization till simulation conclusion, which is 28800s, equivalent to 8hrs. At the end of each 8 hours we collect the statistics from the system. Our system was initialized with seeds 7, 9, 5, 6, for Nodes 1, 2, 3, and 4, respectively. In Tables III, IV, VI, V, we represent the results for Nodes 1 to 4, respectively, these show the performance of each node during simulation. In Table.II we show an average of these statistics across all nodes to visualize the overall performance of the system and whether it conforms to the expected values.

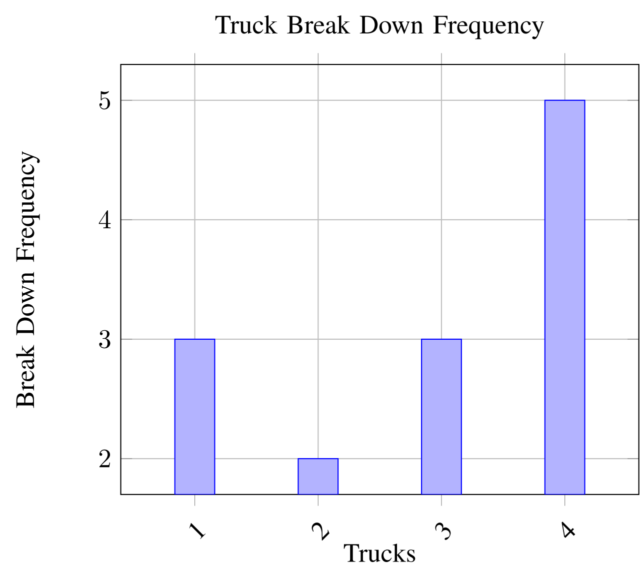

Fig. 1. Truck Break Down Frequency

VI. Discussion

From the results presented in the previous section we can observe that the model functions as designed, but how does it compare to the actual real-world model it is modeled against? Well, in this section we discuss the performance of the model, for both verification and validation, while also highlighting the efficiency of each entity, including internode performance comparison. The parcels received at each node are relatively similar for Nodes 1, 2, and 4, though in Node 3 they deviate from the rest shown in Table.2, which is attributed to the fact

[1]TPD = Total Parcels Delivered, TPR = Total Parcels Received, PSD = Parcels Successfully Delivered

Event	Frequency
Total time truck idle	2210.5s
Total time broken down	3435s
Time traveling between nodes	19686s
Total broken down count	3.25
Time spent unloading	6225,25s
Parcels remaining at base	89.75
Parcels still in truck	13.75
Parcels delivered successfully	128
Parcels failed to deliver	3

TABLE II
AVERAGE VALUES FOR EACH EVENT AND ACTIVITY FOR 1 DAY, ACROSS 4 NODES

Event	Frequency
Total time truck idle	2720
Total time broken down	2863
Time traveling between nodes	19526
Total broken down count	3
Time spent unloading	5441
Parcels remaining at base	52
Parcels still in truck	26
Parcels delivered successfully	162
Parcels failed to deliver	4

TABLE III
STATISTICS FOR NODE 1.

Event	Frequency
Total time truck idle	0
Total time broken down	1773
Time traveling between nodes	19438
Total broken down count	2
Time spent unloading	7992
Parcels remaining at base	67
Parcels still in truck	29
Parcels delivered successfully	166
Parcels failed to deliver	5

TABLE IV
STATISTICS FOR NODE 2.

Event	Frequency
Total time truck idle	2712
Total time broken down	5888
Time traveling between nodes	20111
Total broken down count	5
Time spent unloading	5340
Parcels remaining at base	156
Parcels still in truck	0
Parcels delivered successfully	100
Parcels failed to deliver	2

TABLE V
STATISTICS FOR NODE 4.

Event	Frequency
Total time truck idle	3410
Total time broken down	3216
Time traveling between nodes	19669
Total broken down count	3
Time spent unloading	6128
Parcels remaining at base	84
Parcels still in truck	0
Parcels delivered successfully	84
Parcels failed to deliver	1

TABLE VI
STATISTICS FOR NODE 3.

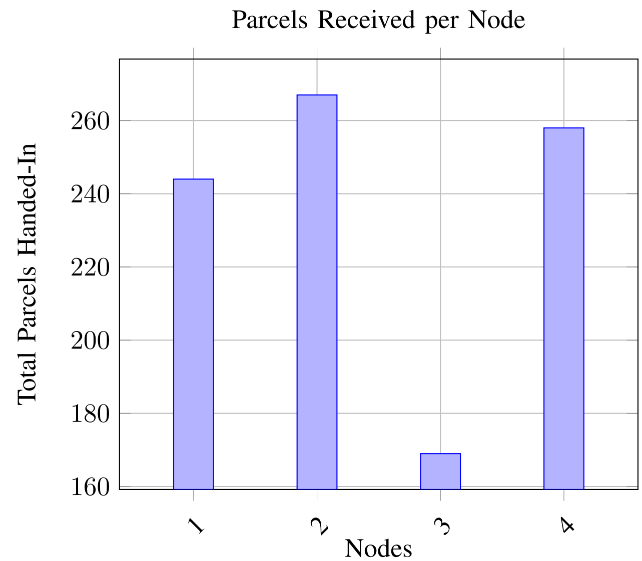

Fig. 2. Total Parcels Handed-In per Node

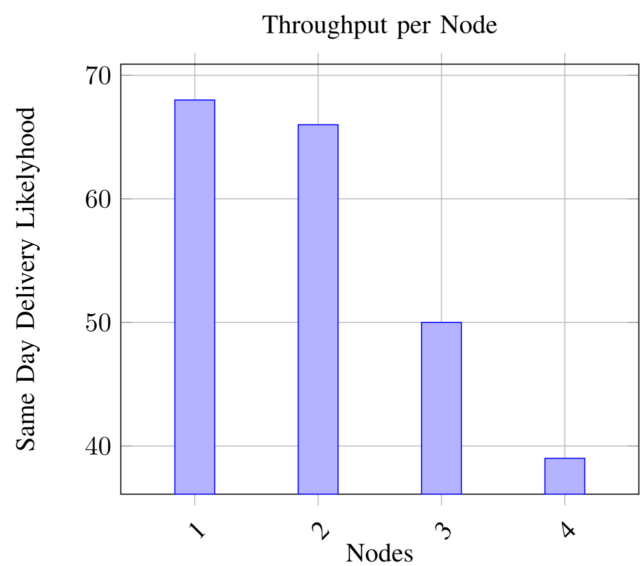

Fig. 3. Throughput per Node

that Node 3 is initialized with a low mean parcel count of 5, compared to other nodes with 7, 9, and 6 respectively. In hindsight it seems inconsequential, but also factoring in other variables like the seed, and what values are drawn in what order, this could result in such deviations. Overall, parcel arrival count values are as expected from the modeling.

In terms of efficiency, seen in Fig.3, Nodes 1 and 2 exhibit the highest throughput, meaning parcels delivered at those 2 nodes are more likely to be delivered within the same day compared to those handed in at Nodes 3 and 4. This is also supported by the number of parcels handed in at Nodes 1 and 2, shown in Table.2, showing these 2 nodes receive the highest parcels compared to Node 3 and similar to that of Node 4, but still have more than twice the throughput of Node 4.

In Table.II, are mean values averages across the 4 nodes in our system. It shows that our model is accurately modeled and consistent with the expected values presented in Table.II, for instance total time for repair is 17 mins across all nodes assuming all average breakdown count is 3, we can see that this value conforms with the range represented in Table.I, 12 to 22 mins.

VII. Future Works

Above are some good references that show the best possible properties, which could significantly influence model design in a real-life implementation. But we should note the system could use a lot of improvements for it to be fit for actual comparison to a real life system. For instance the model we presented was run on a single day cycle, but these results could drastically be different over the course of 1 month, therefore in our future work we plan to extensively test the model and from this we can yield accurate data. In addition we would also like to expand efforts into different modes of transportation as drones are becoming popular it would be nice to compare the efficiency and effectiveness of multiple modes of transportation while simulating real world scenarios like traffic and weather etc.

VIII. Conclusion

The results presented in the results section highlight the strength of the proposed system as it creates a courier system based on real life events, and based on this we can observe how components interact with each other to achieve tasks. This gives us an overview of what can be changed to improve performance without wasting resources on a physical implementation. We can note that from the overall performance of our system, it replicates how actual couriers work, as they receive, route, and deliver packages, some taking short time and some taking longer time to arrive. Along with anomalies such as truck breakdown that influence delivery times.

References

[1] A. Varga et al., *OMNeT++ Discrete Event Simulation Framework*, Version 6.0. Available: https://omnetpp.org/, 2023.

[2] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, ICST, 2008, pp. 1–10.

[3] A. Varga, *OMNeT++ User Manual*, Version 6.0. Available: https://doc.omnetpp.org/omnetpp/manual/, 2023.

[4] A. Varga et al., *INET Framework for OMNeT++*. Available: https://inet.omnetpp.org/, 2023.

[5] H. Sundani, H. Li, V. Devabhaktuni, M. Alam, and P. Bhattacharya, "Flora: A wireless sensor network simulation framework," in *2011 IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications*, IEEE, 2011, pp. 65–72.