A Model of Decentralised Distribution Line using Layer 2 Blockchains

Subhasis Thakur[1[0000−0001−6579−724X]] and John Breslin[1[0000−0001−5790−050X]]

National University of Ireland, Galway, Ireland
{subhasis.thakur, john.breslin}@nuigalway.ie

Abstract. At least one-third of the food produced for human consumption is lost and wasted due to inefficient perishable product distribution plans. Currently used distribution plans are static and centralised as a single entity decides on a distribution plan where products may move along a predefined path. In this paper, we developed a dynamic distribution plan generation algorithm where distribution paths are dynamic. We used blockchain offline channels to develop decentralised, secure, and privacy-preserving dynamic distribution plans in a multi-party supply chain. Our algorithm generates distribution plans using real-time quality information about perishable goods so that products may be sold before their shelf life expires. Our algorithm ensures the privacy of all actors in a supply chain as inventory and order information of all actors remains privacy-preserved.

Keywords: Perishable supply chains · Distribution lines · Food loss and waste · Blockchains · Layer 2 Blockchains.

1 Introduction

At least one-third of all food produced is wasted and lost due to not consuming it before its expiry date. Reducing food loss and waste is an important goal to develop sustainable supply chains. In this paper, we design sustainable supply chains for perishable food products using blockchains. In this paper, we study a distribution line of one manufacturer with multiple manufacturing units, multiple distribution units, and multiple retail units. The manufacturer does not own and controls operations of the distribution units and the retail units. The distribution and retail units are competitors and may not reveal their order and demand information. Additionally, we consider the uncertainty of the shelf life of perishable food products. All distribution units are assumed to have IoT devices monitoring the perishable food items and can detect the remaining shelf life of perishable items. In these settings, in a static distribution line, retailers estimate the expected demand of the item and place orders of items using such demand estimation with a lead time (estimated time to acquire the product from the distributors and remaining shelf life). A distributor can estimate demand by aggregating the demands from the retailers or other distributors who placed an order of the items to it, and it will place its order to the manufacturer or other

retailers accordingly. In a static distribution line, the party that fulfils the order is fixed. If there is a disruption (products are expired due to environmental factors) then the order is not fulfilled. In this paper, we proposed to use dynamic distribution lines where the party who fulfils the order is not fixed.

The dynamic distribution line requires collaboration among various of a supply chain as order and inventory stock level information is needed to produce a dynamic distribution plan. However, there are security and privacy problems in developing a dynamic distribution line. Distributors and retailers are competitors and they may not wish to reveal their order information. It may be possible that a malicious distributor may not fulfil the order of a specific retailer or distributor. Thus the dynamic distribution lines route the products in a privacy-preserving fashion, where order information is not revealed to parties outside a group of trusted distributors. Additionally, the objective of dynamic distribution lines should reduce overall food loss and waste and all parties can verify a solution to such an optimisation problem without all order information. Our main contributions are as follows:
(1) We developed a blockchain-based dynamic distribution line management solution. We used public proof of work-based blockchains. We used blockchain offline channels to develop a high scale distribution line management algorithm.
(2) We developed a privacy-preserving distribution line management algorithm where order information of a distributor or retailer will not be known by any distributor.
(3) We developed a dynamic distribution line management algorithm that matches supply and demand information based on real-time product decay information.
(4) We prove the security and privacy-preserving property of the proposed distribution line management solution.
(5) Using experimental evaluation with simulations of distribution lines we show that the proposed distribution line management solution can significantly reduce food waste.

The paper is organised as follows: in Section 2 we discuss distribution line management literature, in Section 3 we discuss the dynamic distribution line management problem, in Section 4 we discuss the blockchain-based distribution line management algorithm, in Section 5 we analyse the security and privacy-preserving properties of the proposed algorithm, in Section 6 we present an experimental evaluation of the proposed solution, and we conclude the paper in Section 7.

2 Related Literature

In [4] a survey of supply chain optimisation methods was presented where optimisation conditions include waste reduction and sustainability parameters. In [5] the author developed methods to optimise perishable distribution lines to reduce food waste. In [3] the authors showed how to use IoT devices to monitor meat quality in a supply chain. In [8] the authors used Monte Carlo simulation to optimise distribution in a perishable supply chain. In [2] the authors developed an

algorithm for designing an optimal transportation network for perishable goods. In [13] the authors developed a multi-agent systems-based simulation of perishable supply chains. In [1] the authors developed pricing strategies for perishable goods. In [7] the authors optimised the transportation network to reduce food loss. In [14] authors used IoT devices to monitor product quality in perishable supply chains. In [12] authors presented optimisation algorithm for perishable supply chains. In [11], authors used Bitcoin offline channel network to develop a decentralised product serialisation method. In this paper, we used proof of work-based blockchains. Proof of work-based blockchains was proposed in [9]. There are several variations of blockchains in terms of consensus protocols. Offline channels for Bitcoin, i.e., Bitcoin Lightning network was proposed in [10], which allows peers to create and transfer funds among them without frequently updating the blockchain. We advance the state of the art as follows: (1) We use public blockchain and offline channel of public blockchains to execute dynamic distribution line generation algorithm. Public blockchain makes the participation of supply chain actors in blockchain-based solution and offline channel improves the scalability of the solution. (2) We used public blockchains to execute the distribution line generation algorithm. It improves interoperability among actors of a supply chain as each actor should only be interoperable with the public blockchain transaction and smart contract data structure. (3) The proposed solution preserves the privacy of actors of a supply chain as inventory and order information of all actors.

3 A model of perishable supply chain

In this paper, we will investigate distribution lines for the following supply chain: (a) The supply chain has multiple manufacturing units producing identical perishable items. (b) Demand of a distributor or retailer can be satisfied by any item in this supply chain, i.e., they can not differentiate among the items from different manufacturer or distributors. We will use the following terminology to describe the distribution line: We will represent a distribution line a multipartite graph $G = (L, M_1, \ldots, M_q, R, E)$ with set of vertices $L \cup M_1 \cup \cdots \cup M_q \cup R$ and set of edges E. The set of vertices represents the actors (manufacturer, distributors, and retailers) of a supply chain. The set of nodes L represents the manufacturers, $\{M_i\}$ represents the sets of distributors, and L represents the retailers. The set of edges represents the flow of items and the flow of demand information. The edges $(v_i \to v_j) \in E$ represent flow of goods. Edges are such that,
(1) Vertices L represent the manufacturing units and can only have edges with a set of vertices M_1.
(2) Vertices in the sets $M_1, \ldots, M_k$ are distribution centres. A vertex in M_x can only have edges with vertices from the set M_{x-1} with $x > 1$.
(3) Vertices R represent the retail units and can only have edges with a set of vertices M_q.

We will use the discrete-time to represent the time it takes to move the products and the shelf life of the products. We will denote time as a sequence

of uniformly increasing positive integer $t_1, t_2, t_3, \ldots$ with $t_i - t_{i-1} = dt$ for all i. We assume that distances between a pair of vertices can be traveled by time $d(e)$ where $e \in E^S$ is the edge between the vertices. Products are assumed to have a shelf life given by the equation: Decay level of a product i at time t is: $1/(1 + \frac{(.4*e)^t}{a})$, where t is the time since the product has been produced and $a \in [0, A]$ chosen uniformly at random from $[0, A]$ where A is a positive integer. Note that the maximum value of Q_i^t is 1 and the minimum is zero. If a is increased the products degrades slowly. We assume that $Q^* \geq .01$ is the threshold quality of the product that can be safely consumed. Hence any product i with quality less than Q^* is discarded and considered as food waste. We will use a just-in-time inventory management policy for all actors where an actor will only place an order for an item if an item is moved from its inventory or it has received an order for an item.
(1) If an actor v_x receives an order for an item with shelf life S then it will select an item with shelf life $S + \epsilon$ such that ϵ (positive number) is minimum. Thus an actor will move a product that meets the minimum shelf life requirement.
(2) If an actor v_x receives an order for an item with shelf life S and it does not have any item in its inventory to fulfil the order then it will unable to fulfil the order.
(3) If an actor has sent an item with shelf life $S + \epsilon$ from its inventory then it will order an item with a shelf life of at least $S + \epsilon$.
(4) If an actor has sent n items then it will order n items.

In this supply chain network, consumption of items, orders for items are created as follows:
(1) At a time t_1, the number of items to be sold at a retailer is a random number from a probability density function.
(2) At the time t_1, a retailer v_x will evaluate the number of items it has sold, let it has sold n items. The retailer will chose a distributor v_y such that $(v_y \to v_x) \in E$, i.e., product can move from v_y to v_x. It will place an order of n items with a shelf life of more than a threshold (estimated by the maximum time it takes to move a product from the manufacturer to the retailer).
(3) At the time t_1, a distributor v_x will evaluate the number of items it has sold and orders it has received at time t_1. If v_x has moved an item with shelf life S as it fulfils the order of either a distributor or a retailer then, v_x will place an order for an item with shelf life $S + \epsilon$ (ϵ is a positive number) to any distributor or manufacturer v_y such that $v_y \to v_x \in E$. If v_x has received an order for an item with shelf life S as it fails to immediately fulfils it then, v_x will place an order to either a distributor or a retailer for an item with a shelf life of at least $S + \epsilon$.
(3) At the time t_1, if a manufacturer received an order for an item with shelf life S then it will choose an item with shelf life at least S from its inventory and move the product. It will immediately produce a new product and replenish its inventory. If it can not immediately fulfils the order then it will wait until a new product is created with a shelf life of S.
(4) The order of an item from an actor v_x to another actor v_y with shelf life S

is represented by the tuple $O_i =< v_x, v_y, S >$.
(5) The movement of an item with serial number ID_z from an actor v_x to another actor v_y with shelf life S is represented by the tuple $T_i =< v_x, v_y, ID_z >$.

The above process of order placement and product movement represents a static distribution line as an actor v_x who has received an order for an item from the actor v_y then only v_x can fulfil the order. The problem of designing a dynamic distribution line is as follows: (1) If an actor v_x who has received an order for an item from the actor v_y then v_x can delegate the order to another actor v_z to fulfil the order. Product quality information will be available to decide on such order delegation. We will explain in the next section how to collect the product quality information in real time with IoTs. (2) Let at the k'th layer of the supply chain network actors D_k supplies to the actors D_{k+1}. O^k be the collection of orders from D_{k+1} to D_k and I^k be the set of inventory items for the actors D^k. A bipartite graph $H = (D_k, D_{k+1}, E^H)$ will be constructed where $E^H \subset E$ (It means a product can only move along any permitted distribution path indicated by E. It also represents the preference of an actor to send/receive from other actors. Only one item can be transferred along one edge.) and for any $(v_x, v_y) \in E^H$ such that the shelf life of the item in the inventory of $v_x \in D_k$ is at least the lead time of the order placed by $v_y \in D_{k+1}$. The weight of an edge $(v_x, v_y) \in E^H$ is $1/1 + (Shelf\_life(a_x) - O^3_y)$ where $Shelf\_life(a_x)$ is the shelf life of v_x's item and O^3_y is the lead time of order placed by v_y. (3) A dynamic distribution line algorithm produces a maximum weighted matching $\mathcal{M}$ for the graph H. If the edge $(v_x, v_y) \in \mathcal{M}$ then the item from v_x will move to v_y. In the next section, we will describe how to execute this matching algorithm in blockchain offline channels.

3.1 Problem of designing decentralised distribution lines

A generic model of decentralised distribution line may include an additional entity called 'mediators'. Mediators will buy and sell from all entities of the above-mentioned distribution line. In such a distribution line, the buyers and the seller will reveal quality information of products they want to buy or sell. Let $(q^1_B, \ldots, q^x_B)$ and $(q^1_S, \ldots, q^y_S)$ be the sets of quality of products to be bought or sold. Let $Match : (q^1_B, \ldots, q^x_B) \mapsto (q^1_S, \ldots, q^y_S)$ maps a product from a buyer to a seller. The optimal matching solution will minimize the following $\sum(q^i_B - Match(q^i_B))$, where $q^i_B - Match(q^i_B) \geq 0$. The problems of designing a decentralised distribution line that satisfies such optimisation condition are as follows: (a) Buyers would like to hide their quality information of products to be bought. This is because such information may reveal the inventory state of a buyer. (b) A mediator may not match the buyers and the sellers according to the objective shown in equation 1. A mediator may match a product with a long shelf life to a buyer who is demanding products with short shelf life. In this case, the buyer will benefit by procuring a product with long shelf life. (c) As the buyers want to hide information about the shelf life of ordered products, it is problematic for a buyer to verify if matching is performed with objective of equation 1.

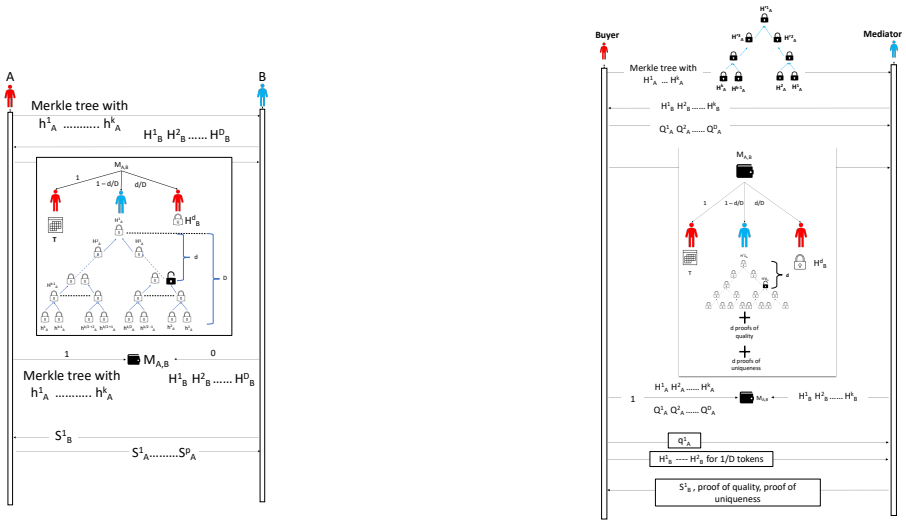
4 Dynamic distribution with blockchains

We assume that all actors of the supply chain network are part of public proof of work-based blockchains. The proposed decentralised distribution line management solution may be executed in any blockchain network that can execute Hashed-Time Locked Contracts or similar scripts. In a supply chain network $G = (M, D_1, \ldots, D_q, R, E)$ there are $q+1$ sets of mediator nodes $X_1, \ldots, X_{q+1}$ where each set X_i contains a fixed number (positive integer) peers of the blockchain network. The supply chain network is a multi-partite graph with q+2 layers where M is the first layer, D_1 is the second layer, and so on. Product move from layer x to layer y where $x < y$. Actors in the $i-1$th layer and $i+1$th layer will establish offline channels with mediators of the set X_i. Let $t_1, t_2, \ldots$ are discrete-time instances such that $t_i - t_{i-1} = dt$ where dt is the number of new blocks in the blockchain. We assume that new blocks are added to the blockchain after an approximately equal time interval.

At the beginning of a time interval t_x, all actors in the $i+1$th layer submit their order to any mediator node in the set X_i with the lead time of the order. These actors will be referred to as the buyers. The buyers will be allowed a time $dt' < dt$ to send their orders. Order with lead time can be placed by creating a new offline channel or by updating an offline channel. After time dt', the actors in the set i'th layer will send their intent to move inventory to the mediators X_i. These actors will be referred to as the sellers. Sellers will inform the mediators about their intent to move inventory by sending the self-life of their products to the mediators. They can do so by creating or updating offline channels with the mediators. The sellers can decide to move products from their inventory at any time in the time duration $t_i + dt'$ to $t_i + dt$ to sell their product before the time instance t_{i+1}. The mediators will execute an online maximum weighted bipartite graph matching algorithm to match the products from the sellers to the buyers. Next, we will explain how to open and update offline channels and execute the graph matching algorithm.

Blockchain offline channels [10] uses multi-signature addresses to open an offline channel among peers of the blockchain. This offline channel[10] is bidirectional and potentially infinite, i.e., it can execute the infinite number of transfers between two peers provided they do not close the channel and each of them has sufficient funds. We construct an offline channel for proof of work-based public blockchain with the following properties: (a) We construct a uni-directional channel between two peers, i.e., only one peer can send funds to another peer of this channel, (b) we construct a uni-directional channel that can be used for a finite number of transfers from a designated peer to another peer. The procedure for creating the uni-directional channel from A to B (A transfers token to B)is as follows: Let A and B are two peers of the channel network H. $M_{A,B}$ is a multi-signature address between A and B. This is a unidirectional channel from A to B.

1. A creates a set of k (k is a positive even integer) random strings $S_A^1, \ldots, S_A^k$. Using these random strings A creates a set of Hashes $h_A^1 = Hash(S_A^1)$,



(a) (b)

Fig. 1: (a) Procedure of creating unidirectional offline channels, (b) Channel between a buyer and a mediator

$h_A^2 = Hash(S_A^2)$, ..., $h_A^k = Hash(S_A^k)$ where $Hash$ is Hash function (using SHA256). A creates a Merkle tree height $D = Log_2 k$ using these Hashes. In this tree there are k leaf nodes and $k-1$ non-leaf nodes of this Merkle tree. We denote the non-leaf nodes as $H_A^1, \ldots, H^(k-1)_A$.

2. B creates a set of $D-1$ random strings and corresponding Hashes $H_B^1, \ldots, H_B^{D-1}$ such that there is a lexicographic order among these Hashes with $H_B^i \leq H_B^{i+1}$. We will call H_B^x is ranked more than H_B^y if the lexicographic order of H_B^x is more than H_B^y.
3. A will create a Hashed time-locked contract as follows: (a) Let A wants to transfer $1 - d/D$ tokens to B where $d \leq D$. (b) From the multi-signature address $M_{A,B}$ 1 token will be given to A after time T(will be measured as the number of new blocks to be created from the current block in the blockchain) if B does not claim these tokens by producing the key to any non-leaf node H_A^x of the Merkle tree created by A, which is at a distance d from the root of the Merkle tree. (c) If B produces the key to such a Hash H_A^x at depth d then $(1 - d/D)$ tokens will be given to B and remaining d/D tokens will be given to A if it can produce the key to a hash H_B^x in the set $H_B^1, \ldots, H_B^{D-1}$ which is ranked more than d hashes in this set.
4. A will sign this HTLC and send it to B.
5. Now A will send a transaction to $M_{A,B}$ of amount $1 + \epsilon$ token with the Merkle tree mentioned in the transaction. B will send ϵ tokens to $M_{A,B}$ with $H_B^1, \ldots, H_B^D$ in the transaction data field where ϵ is the transaction

processing fee. More than 1 token can be transferred by A. We are using 1 token as an example.

6. Before the next transfer from A to B, A will send the keys to the subset of Hashes $h_A^1, \ldots, h_A^k$ which can generate the Hash H_A^x. It will be possible for A to transfer multiples of $1/D - 1$ tokens to B, in such a case multiple subsets of non-leaf Hashes of the Merkle tree will be revealed by A.

Note that, (a) B signs and publishes the HTLC to claim the tokens. It gets the most number of tokens by using the last known keys of the Merkle tree leaf nodes. Thus B will always use the last known keys of this Merkle tree leaf node. (b) The channel is secure as A can not transfer to B more than the current balance of the channel as B will know the current balance by finding the non-leaf nodes from the keys supplied by A and depth of such a non-leaf node. (c) All peers of the blockchain will know the existence of this channel and Hashes used in creating the channel as transactions to $M_{A,B}$ are visible to all peers.

4.1 Channel between a buyer and a mediator

A buyer will pay a mediator if the following holds: (1) If the mediator can provide a product with shelf life more than the requested lead time of product ordered by the buyer to the mediator. (2) Additionally, the buyer should ensure that it is receiving a unique product. The protocol (shown in Fig. 1(b) with buyer is A and the mediator is B is as follows: We modify the unidirectional channel as follows: (1) A sends B a sequence of Hashes of qualities (indicating the lead time of products) $\{Q_A^i\}$. (2) A sends B an HTLC similar to the HTLC in Fig. 1(a). The only changes are B can claim tokens of value d/D tokens if it can provide d proofs of quality and d proof of product uniqueness. (3) After sending this HTLC, A sends the first quality requirement q_A^1 (whose Hash is Q_A^1) to B and a set of keys in the Merkle tree created by A to claim $1/D$ tokens from $M_{A,B}$. B can claim these tokens by presenting proof of quality and proof of uniqueness. (4) Proof of quality will verify the quality of a product as claimed by a seller. We will assume that there is an IoT network monitoring quality information of the products and such quality information is periodically uploaded to an IPFS blockchain. A buyer or a seller can find proof of quality by checking the IPFS blockchain for quality information. (5) If B can present such proof of quality and proof of uniqueness to A then A will reveal the next set of keys of the Merkle created by A to send the next set of $1/D$ tokens. Otherwise, A will reveal quality information q_A^2 (whose Hash is Q_A^2) to B and ask it to find an appropriate product.

4.2 Channel between a seller and a mediator

A channel between a seller and a mediator is similar to the uni-directional channel mentioned in the previous section with the following additional information: (a) A seller will commit to a set of serial numbers to a mediator. It will store the Hash of such serial numbers in the HTLC used to create the channel between

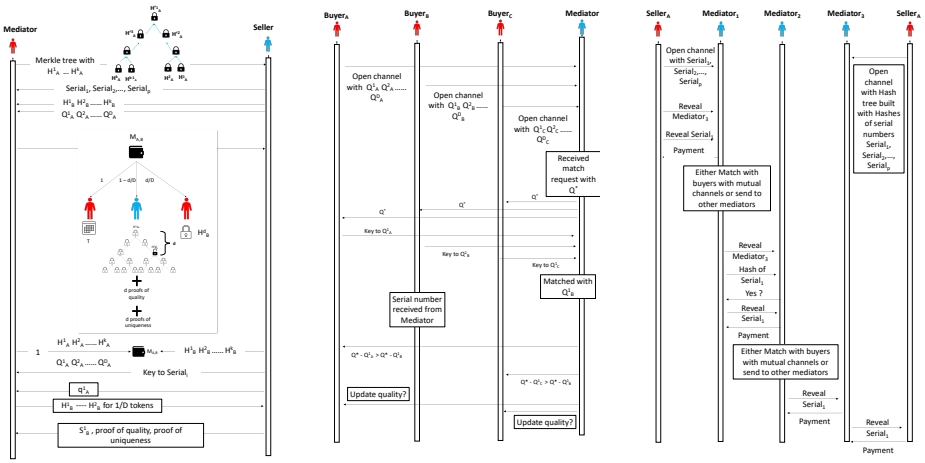
itself and a mediator. (b) Apart from revealing the quality of an item, the seller must reveal a new serial number by producing the key to the Hash of a serial number. Additionally, each seller may have a uni-directional channel (Fig. 1(a)) with at least one sink mediator with a Hash tree formed using the serial numbers of products owned by the seller according to the channel construction protocol. The sink mediators are mediators without channels from buyers.

4.3 Channel among mediators

The mediators are expected to build offline channels among themselves according to the bi-directional offline channel protocol developed in [10].

4.4 Decentralised product matching

We will describe the protocol for decentralised distribution as a decentralised matching algorithm that matches the order of a buyer to the inventory of a supplier. First, we will explain how a mediator can match a product with the order it has received from the buyers. Then we will explain how a seller can sell a product via the mediators. The protocol of product sale from mediator to



(a) a (b) b (c) c

Fig. 2: (a) Channel between a seller and a mediator, (b) Protocol of product sale from a mediator to buyers, (c) Protocol of product sale from a seller to mediators.

buyers is as follows:

1. Assume that a mediator has received a product (of quality Q^*) from a seller or from another mediator and it wants to sell the product to the buyers who have established channels with it.

2. The buyers have established uni-directional channels with the mediator. Let there are 3 buyers $Buyer_A$, $Buyer_B$, and $Buyer_C$. They have established channels with quality information (Q_A^1, Q_A^2,..., Q_A^D), (Q_B^1, A_B^2 , ... , Q_B^D), and (Q_C^1, A_C^2,..., Q_C^D) respectively.
3. The mediator reveals Q^* to $Buyer_A, Buyer_B$, and $Buyer_C$.
4. $Buyer_A, Buyer_B$, and $Buyer_C$ may reveal new reveal key to their respective quality Hashes (i.e., $Buyer_A$ may reveal the key to Q_A^i).
5. Mediator finds the buyer with the minimum difference between Q^* and qualities revealed by the buyers. Say $Q^* - Q_B^i < Q^* - Q_A^i < Q^* - Q_C^i$. Mediator sells the product to $Buyer_B$ as it reveals the serial number to $Buyer_B$.
6. Next, the mediator will send the information $Q^* - Q_B^i < Q^* - Q_A^i < Q^* - Q_C^i$ to all $Buyer_A, Buyer_B$, and $Buyer_C$.
7. Finally, the mediator will ask $Buyer_A, Buyer_B$, and $Buyer_C$ to reveal new quality information.

The protocol of product sale from the seller to mediator is as follows: (1) A seller $Seller_A$ will establish a unidirectional channel with a mediator $Mediator_1$ with serial number $Serial_1, \ldots, Serial_p$ according to the channel establishment protocol shown in Fig. 1(b). $Seller_A$ will establish a unidirectional channel with a sink mediator $Mediator_3$ with hash tree created by $Serial_1, \ldots, Serial_p$ according to the channel establishment protocol. (2) Now, $Seller_A$ will reveal key to $Serial_1$ to $Mediator_1$. (3) $Mediator_1$ will verify that there is a sink mediator $Mediator_3$ and there is a channel from $Seller_A$ to $Mediator_3$ with a hash tree that includes $Serial_1$. If $Mediator_1$ can verify this then it will pay $Seller_A$. (4) Now $Mediator_1$ will try to match the product with the serial number $Serial_1$ to the buyers who have a channel with it. If $Mediator_1$ can not do so (according to the previous protocol for product sale from a mediator to buyers) then it will try to resale this product to other mediators. (5) $Mediator_1$ can either sell it to another mediator or it can sell it to the sink mediator $Mediator_3$. (6) If it sells the product to sink mediator $Mediator_3$ by revealing the serial number then $Mediator_3$ can claim the payment from its with $Seller_A$. (7) If this sequence of product resale does not reach the sink mediator then it will indicate that the product is sold to a buyer otherwise it will indicate that the product can not be sold and it is returned to $Seller_A$.

5 Analysis

Lemma 1. *The decentralised product matching protocols are privacy-preserving.*

Proof. An adversary (who wants to know the quality information of products) will have access to the blockchain network and it has information on offline channels established among buyers, sellers, and mediators. However, it can not know the quality information because Quality information is kept in the channel from a buyer and a mediator or from a seller to a mediator. In both cases, Hashes of such information are included in the transaction used to establish the channels. Hence the adversary may not know the quality information of any buyer or seller if it can not control all mediators.

Lemma 2. *A mediator can not manipulate the matching protocol for product sale from mediator to buyers shown in Fig. 2(b).*

Proof. This holds because: (1) As shown in Fig. 2(b), $Buyer_A$ will verify if $Q^* - Q_A^1 > Q^* - Q_B^1$ as the mediator should produce the key to Q^* and Q_B^1 to $Buyer_A$. (2) $Buyer_A$ has access to the transaction records and it can find the transactions from sellers to mediators and buyers to mediators that include Q^* and Q_B^1. (3) $Buyer_A$ can verify that the above transactions are unspent and hence such quality information is not reused. (4) In case the mediator does not reveal the quality information, $Buyer_A$ may choose not to buy products via this mediator.

Lemma 3. *The mediators will not lose funds in the product sale protocol from a seller to a mediator shown in Fig. 2(c).*

Proof. This holds because: (1) A mediator can verify the existence of a channel from the seller to a sink mediator by searching the transaction list of the blockchain. (2) Such a transaction denotes the existence of a channel from the seller to a sink mediator. (3) A mediator can verify that the above-mentioned channel is not closed as it can check the transaction list to ensure that the above transaction is unspent. (4) In such a live channel from the seller to the sink mediator, it can check the existence of the Hash of the serial number of product being sold by the seller. This will mean that the mediator can always resale the product to this sink mediator who can claim funds from the seller by producing the key to the Hash of the serial number.

6 Evaluation

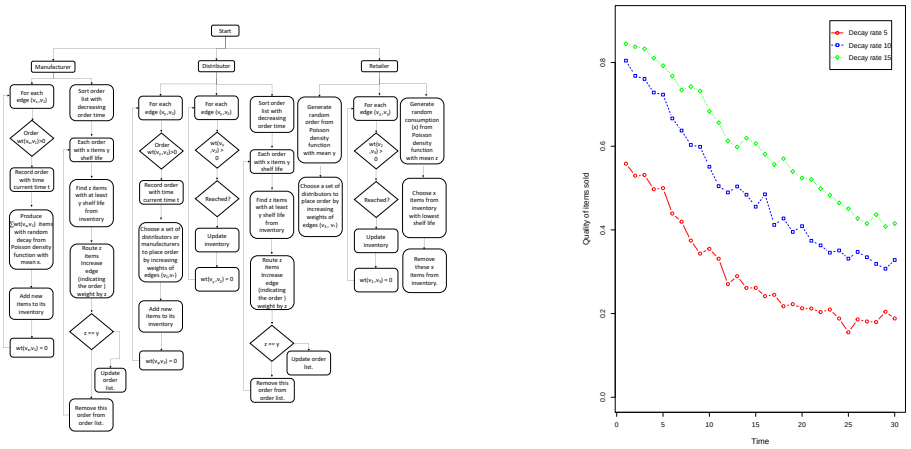
We will evaluate the performance of the proposed with a simulation of supply chain and blockchains. We use proof of work-based simulation of blockchains used in [6]. We developed a supply chain simulator using agent-based modelling where all actors of a supply chain are modelled as autonomous agents. Fig. 3(a) shows the workflow of a manufacturer as it executes two asynchronous processes. The first process simulates the product manufacturing rate as it manufacturer items (with a random decay) according to its demand. The second process simulates the order fulfilment procedure as it moves products from its inventory to the distributors. Fig. 3(a) shows the workflow of a distributor, which is similar to the workflow of a manufacturer with an additional process that checks if an item in the transit have reached the distributor or if a product in transit should be discarded due to expiry of its shelf life. Fig. 3(a) shows the workflow of a retailer who executes three asynchronous processes. We use the following supply chain network dataset:

We check the correctness of the supply chain simulation by executing it with increasing decay rate fro 5 to 15. We observe (Fig. 3(a)) that as decay rate is

Table 1: Data for experiments

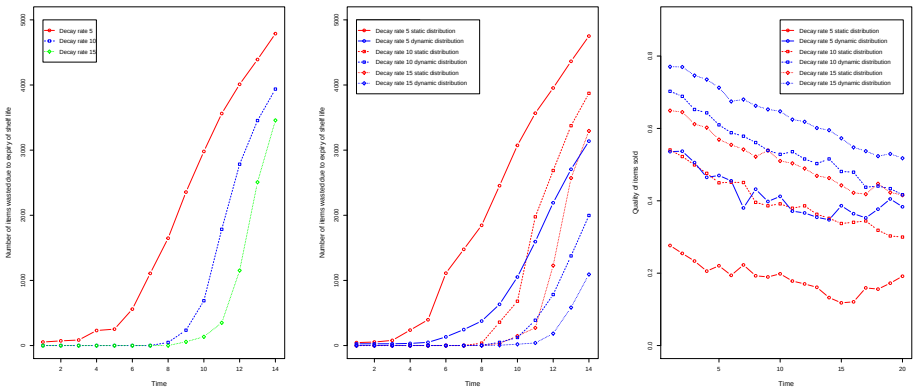
$\lvert M\rvert = 20$, $\{\lvert D_1\rvert = 20\}_{i=1,2,3}$, $\lvert R\rvert = 20$	# of manufacturers, distributors, & retailers.
$W(E) \in Random(1,6)$	Distance among the actors.
$\lvert \cup I_i\rvert = 10000$	Number of items
$a_j \in [5,15]$	Decay rate
$\lambda_{demand}, \lambda_{consume} = 6$	PDF for retail demand & consumption.

increased waste is decreased and quality of items sold is increased (Fig. 4(a)). Next, we evaluate the performance of the dynamic distribution line generation with a static distribution line generation algorithm. As mentioned in section 3, a static distribution line order can not be delegated and in a dynamic distribution, line order can be delegated using a graph matching algorithm executed via an offline channel network. We used an offline channel network with 5 matching nodes (nodes in the offline channels facilitating the matching process) between every pair of adjacent sets of nodes of the supply chain network (there are 5 layers in the supply chain network and we have 20 matching nodes). We execute the static distribution and dynamic distribution with the same sets of data by increasing the decay rate from 5 to 15. We observe that waste is significantly reduced using dynamic distribution lines (Fig. 4(b)) and the quality of items sold is increased by dynamic distribution lines (Fig. 4(c)).



(a) a (b) b

Fig. 3: (a) Workflow of the manufacturers, distributor, and retailers, (b) It shows the correctness of the simulator: as we increase the decay rate, waste is reduced and quality of products sold is increased.



(a) a (b) b (c) c

Fig. 4: (a) It shows the correctness of the simulator: as we increase the decay rate, waste is reduced and quality of products sold is increased, (b) the waste is significantly reduced using dynamic distribution lines, (c) Quality of consumed product is significantly improved.

7 Conclusion

In this paper, we proposed a decentralised distribution line protocol that dynamically routes perishable products to reduce food loss and waste. We used blockchain offline channels to implement such a distribution process for the high-scale execution of the distribution procedure. In the future, we will extend such distribution protocol for IPFS blockchains.

Acknowledgement

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289, co-funded by the European Regional Development Fund.

References

1. Chen, W., Liu, H., Xu, D.: Dynamic Pricing Strategies for Perishable Product in a Competitive Multi-Agent Retailers Market. Journal of Artificial Societies and Social Simulation **21**(2), 1–12 (2018), https://ideas.repec.org/a/jas/jasssj/2017-4-3.html

2. de Keizer, M., Akkerman, R., Grunow, M., Bloemhof, J.M., Haijema, R., van der Vorst, J.G.: Logistics network design for perishable products with heterogeneous quality decay. European Journal of Operational Research **262**(2), 535–549 (2017). https://doi.org/https://doi.org/10.1016/j.ejor.2017.03.049, https://www.sciencedirect.com/science/article/pii/S0377221717302710
3. Eom, K.H., Hyun, K.H., Lin, S., Kim, J.W.: The meat freshness monitoring system using the smart rfid tag. International Journal of Distributed Sensor Networks **10**(7), 591812 (2014). https://doi.org/10.1155/2014/591812, https://doi.org/10.1155/2014/591812
4. Eskandarpour, M., Dejax, P., Miemczyk, J., Péton, O.: Sustainable supply chain network design: An optimization-oriented review. Omega **54**, 11–32 (2015). https://doi.org/https://doi.org/10.1016/j.omega.2015.01.006, https://www.sciencedirect.com/science/article/pii/S0305048315000080
5. Gaggero, M., Tonelli, F.: Optimal control of distribution chains for perishable goods. IFAC-PapersOnLine **48**(3), 1049–1054 (2015). https://doi.org/https://doi.org/10.1016/j.ifacol.2015.06.222, https://www.sciencedirect.com/science/article/pii/S2405896315004619, 15th IFAC Symposium onInformation Control Problems inManufacturing
6. Hayes, B., Thakur, S., Breslin, J.: Co-simulation of electricity distribution networks and peer to peer energy trading platforms. International Journal of Electrical Power & Energy Systems **115**, 105419 (2020). https://doi.org/https://doi.org/10.1016/j.ijepes.2019.105419, https://www.sciencedirect.com/science/article/pii/S0142061519302972
7. Jedermann Reiner, Nicometo Mike, U.I., Walter, L.: Reducing food losses by intelligent food logistics. Phil. Trans. R. Soc. A. (2014), https://doi.org/10.1098/rsta.2013.0302
8. La Scalia, G., Micale, R., Miglietta, P.P., Toma, P.: Reducing waste and ecological impacts through a sustainable and efficient management of perishable food based on the monte carlo simulation. Ecological Indicators **97**, 363–371 (2019). https://doi.org/https://doi.org/10.1016/j.ecolind.2018.10.041, https://www.sciencedirect.com/science/article/pii/S1470160X18308136
9. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. www.bitcoin.org (2008)
10. Poon, J., Dryja, T.: The Bitcoin Lightning Network:Scalable Off-Chain Instant Payments https://lightning.network/lightning-network-paper.pdf
11. Thakur, S., Breslin, J.G.: Scalable and secure product serialization for multi-party perishable good supply chains using blockchain. Internet of Things **11**, 100253 (2020). https://doi.org/https://doi.org/10.1016/j.iot.2020.100253, https://www.sciencedirect.com/science/article/pii/S2542660520300883
12. Tsao, Y.C.: Designing a fresh food supply chain network: An application of nonlinear programming. Journal of Applied Mathematics **2013** (2013), https://doi.org/10.1155/2013/506531
13. Tykhonov, D., Jonker, C., Meijer, S., Verwaart, T.: Agent-based simulation of the trust and tracing game for supply chains and networks. Journal of Artificial Societies and Social Simulation **11**(3), 1 (2008), http://jasss.soc.surrey.ac.uk/11/3/1.html
14. Yan, B., Lee, D.: Application of rfid in cold chain temperature monitoring system. In: 2009 ISECS International Colloquium on Computing, Communication, Control, and Management. vol. 2, pp. 258–261 (2009). https://doi.org/10.1109/CCCM.2009.5270408