A Blockchain-based NFV Market in the Multi-node Edge Computing Network

Abstract: Currently, network function virtualization (NFV) incorporates cloud computing (CC) and forms a market, providing elastic and cost-efficient chained network services. This paper considers a new NFV market in edge computing, where NFV providers deploy service chains on near-by EC nodes instead of remote data centers. An optimization problem is formulated to minimize the deployment costs of a required service chain from an NFV provider's perspective, with user service delay guarantees. Due to its NP-hardness, we investigate two special network models, where we can turn to dynamic programming solutions. We propose a pricing mechanism based on bargaining theory to decide fair resource prices for EC nodes. To relinquish the full power of the NFV provider, we design a blockchain-based system to implement our algorithms using smart contract. Simulations are conducted and numerical evaluations are presented to demonstrate the efficiency of our solutions and the applicability of our system.

Keywords: Bargaining Theory; Blockchain, Dynamic Programming; Edge Computing; NFV Market; Service Chain.

1 Introduction

Network Functions Virtualization (NFV) separates network functions from proprietary hardware by virtualizing them as software, *i.e.*, virtual network functions (VNFs), that can run on any standardized computing node, and then, enables an elastic and cost-reduced way for users to access network services. With this technique becoming mature, an NFV market has formed, where dedicated NFV service providers create VNF instances, build service chains, and offer them to users on demand. Currently, this market is tightly coupled with Cloud Computing (CC) paradigm, as an NFV provider usually deploys service chains with resources (computing and bandwidth) rented from a certain CC platform. Thus, user traffic will be routed to and served on remote data centers.

However, given the widespread penetration of mobile devices and users' increasing desire for low-latency network service responses, a remote and centralized CC platform is no longer a sufficient solution. Thus, the concept of edge computing (EC), *i.e.*, moving the functionality of CC towards the network edge, has been recently proposed. As a new computing paradigm, EC brings intermediate nodes with resources to the network edge, and hence yields many benefits, *e.g.* shorter response times, more efficient processing, and less pressure on the whole network. This trend enables the emergence of a new NFV market, which brings NFV and EC together by allowing VNFs to be hosted in a distributed, heterogeneous edge network.

This paper studies such an EC-based NFV market, where an NFV provider cooperates with multiple geo-distributed EC nodes and profitably deploys service chains upon user requests. We assume that the geographic span of all EC nodes can be divided into non-overlapping zones and a user is within a unique EC node's coverage zone, given his relatively fixed scope of activity. A discriminatory pricing scheme on bandwidth is applied by each node, given that served users can be inside/outside its coverage zone and the corresponding service delays are also different. Fig. 1 shows an overview of this market. Usually, a group of users, although unevenly distributed across EC nodes, may request an identical service chain. For each VNF requested in the service chain, the NFV provider has to determine both instantiation, *i.e.*, how many instances to create, and placement, *i.e.*, which EC nodes to install them, in order to minimize his deployment costs while not violating his service quality guarantee, *i.e.*, maximal delay constraint. It is challenging for the NFV provider since both cost efficiency and service quality are taken into consideration.

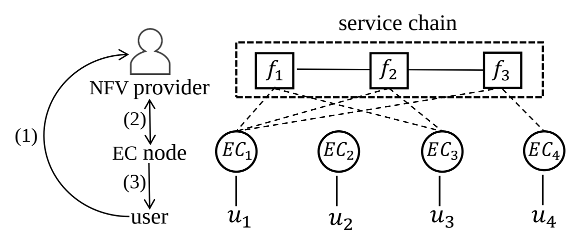


Figure 1: Overview of an EC-based NFV market: (1) geo-distributed users send requests to NFV provider in the form of service chains, (2) NFV provider and EC nodes bargain on resource prices and deploy service chains, (3) users get served from EC nodes.

To deploy a service chain, two types of costs are included: VNF installation costs and traffic transfer costs. The installation costs can be affected by the NFV provider's decision on instantiation (since fewer instances always save money), as well as placement (given unit resource prices vary among EC nodes). The transfer costs mainly depend on instance placement as we only consider cross-node bandwidth consumption. Unfortunately, these two parts are conflicting and cannot be reduced at the same time. Consider the example shown in Fig. 1. A service chain with three network functions, *i.e.*, f_1, f_2, and f_3, needs to be deployed on four candidate EC nodes. If the NFV provider deploys a complete service chain on each EC node, he maximizes his costs on installation while eliminating transfer costs. The NFV provider can only install one instance in EC_1 for each f_i, and thus, directing all user traffics to EC_1. This solution reduces installation costs at the expense of high cross-node bandwidth consumption. In terms of service delay, we only consider cross-node link delays, as well. Thus the second deployment method definitely leads to a long delay for all users except u_1. A suitable deployment is shown by those dash lines in Fig. 1, as it tries to balance VNF installation costs (two instances for each nfv), traffic transfer costs (u_2 and u_4 are served outside), as well as cross-node link delays. Obviously, the complex trade-off between the cost efficiency and service quality makes the deployment problem quite difficult even if only a service chain is considered.

In this paper, we explore service chain deployment in a multi-node EC network. We mainly focus on the cost-minimization objective for an NFV provider while still considering the delay constraints on the user side. Without over-complicating our problem, we assume all users are categorized into different groups and all users in the same group are requesting an identical service chain. We only discuss the deployment of each service chain individually and independently, although some VNF instances can be shared among different service chains. Due to its NP-hardness, we investigate two special network models: a sparsely deployed scenario for loose delay constraints and a densely deployed scenario under a uniform-cross-node-link scheme, both of which can be optimally solved by dynamic programming.

To incentivize EC nodes to participate in the NFV provider-issued resource sharing, we propose a pricing mechanism based on Nash bargaining theory, to decide fair resource prices for all self-interested EC nodes. To relinquish the full power of the NFV provider, we want our proposed solutions to be implemented in a decentralized and traceable way. Thus, we apply the blockchain technique, and design a blockchain-based system that integrates with the existing NFV-enabled EC architecture and our service chain deployment algorithms and pricing mechanism. The contributions of this paper are summarized as follows:

- We present an EC-based NFV market and characterize the interaction among users, EC nodes, and the NFV provider.

- We formulate a service chain deployment problem for the NFV provider by considering both the geographical features of EC nodes and latency constraints of users.

- Since the formulated problem is NP-hard, we solve it using dynamic programming in two scenarios with different requirements on the VNF instance number.

- We propose a pricing mechanism based on bargaining theory, for both the NFV provider and EC nodes to reach an agreement on the prices of each resource unit.

- We design a blockchain-based system to implement the presented NFV market and conduct experiments on it to validate our analysis.

2 Market Model and Problem Formulation

2.1 The NFV Market

This paper focuses on an NFV market in the multi-node EC network. Corresponding notations are listed in Table 1. There exist three entities: (i) a set of EC nodes, providing resources and covering a geographic area without overlapping, (ii) a group of users, distributed across EC nodes and requesting an identical service chain composed of a set of VNFs, and (iii) an NFV provider who rents resources from EC nodes and sells the service chain on user demand.

User traffics are identified by their origin zones. Traffics are generalized as a traffic, denote t, if they are initiated from the same zone covered by node s_t. Let v_t represent t's volume, *i.e.*, the total traffic units of t. We assume all EC nodes are fully-connected and each of them can process traffic from all users. If a node is requested to serve users outside its coverage zone, the corresponding traffic should be transferred to this node first, which will lead to cross-node bandwidth consumption and traffic transfer latency. Given node n and node n', $b_{nn'}$ denotes bandwidth cost per traffic unit and $l_{nn'}$ denotes transfer latency between them. Thus, transferring t from n to n' incurs a bandwidth cost of $v_t b_{nn'}$ and an extra service delay of $l_{nn'}$. When serving its local users, node n will not charge extra cost on bandwidth, *i.e.*, $b_{nn} = 0$, and its service delay is negligible as zero, *i.e.*, $l_{nn} = 0$, since there exists no transfer latency.

We further define p_n as node n's resource price per unit and meanwhile, we assume VNF installation cost is type-related, *i.e.*, any two types of VNFs differ on the resource requirements while the same type of VNF instances are identical on the resource consumption. For simplicity, we define f as the f-th VNF in the requested service chain and then r_f represents its resource requirement per instance. Thus, it costs the NFV provider $r_f p_n$ to install one instance of VNF f on node n.

2.2 Problem Formulation

In this part, we set up a general service chain deployment (SCD) problem in a multi-node EC network as an integer linear programming (ILP) optimization problem.

The goal of the presented optimization model is to minimize the overall cost of the NFV provider, denoted C, with a low service delay constraint required by

Table 1 Summary of Notations.

Symbol	Description
N, T, F	set of EC nodes, traffics, and requested VNFs
$\|N\|, \|T\|, \|F\|$	size of set N, T, and F
$n/n', t/t', f/f'$	a node, a traffic, and an VNF
p_n, c_n	n's unit price, unit cost of resources
$b_{nn'}$	bandwidth cost per traffic unit between n and n'
$l_{nn'}$	traffic transfer latency between n and n'
s_t, v_t	t's source node and volume
p_t, d_t	t's assigned path and path-related service delay
f^-, f^+	f's previous VNF and subsequent VNF
a_f, r_f	f's instance amount and resource requirement
D	service delay constraint
C	NFV provider's overall cost
t_{nf}	decision variable of t through f on n
x_{nf}	indicator of f on n

users, denoted D. We first introduce a binary decision variable t_{nf}, where $t_{nf} = 1$ indicates that traffic t will be processed by VNF f instantiated on node n, otherwise, $t_{nf} = 0$. Therefore, we can easily derive a binary indicator, $t_{nf}t_{n'f^+}$, to represent whether traffic t is transferred from f instantiated on n to its successor f^+ instantiated on n'. Meanwhile, we obtain another binary indicator, denoted x_{nf}, where $x_{nf} = \max\{t_{nf}|\forall t \in T\}$, to represent whether VNF f is instantiated on node n.

Then, the NFV provider's overall cost, *i.e.*, the costs of bandwidth together with the resource usage, and the service delay for traffic t, can be expressed as below:

$$\begin{aligned} C &= \mathbf{p}\mathbf{x}\mathbf{r}^{\intercal} + \mathbf{v}\mathbf{B}^{\intercal} \\ d_t &= tr(\mathbf{t}\mathbf{l}), \end{aligned} \tag{1}$$

where tr is the trace operator which sums up diagonal elements of a given matrix and $\mathbf{p}$, $\mathbf{r}$, and $\mathbf{v}$ are vector expressions of p_n, r_f, and v_t. Meanwhile $\mathbf{x}$ and $\mathbf{t}$ are decision matrices. $\mathbf{B}$ is a vector of unit bandwidth cost and $\mathbf{l}$ is a matrix representing traffic delay between any two nodes. Thus, we get the ILP model to describe our SCD problem.

Problem 1 (SCD).

$$\textit{minimize} \quad C \tag{2a}$$

$$\textit{subject to} \quad d_t \le D, \forall t \in T \tag{2b}$$

As we can reduce the NP-Hard Capacitated Plant Location Problem with Single Source constraints (CPLPSS) [19] to Problem 1, Problem 1 is NP-hard as well. We provide the proof in the below.

Theorem 1. *Problem 1 is NP-hard.*

W: e reduce the NP-Hard Capacitated Plant Location Problem with Single Source constraints (CPLPSS) [19] to Problem 1. In CPLPSS, we are given a set of potential locations for production plants with fixed costs and capacities. A commodity produced by these plants is to be supplied to a set of customers with fixed demands and associated transportation costs. Moreover, each customer must be served by a single plant. The objective is to find a subset of the plats that should be operated to minimize cost without violating capacity and demand constraints. Given an instance of the CPLPSS we can transform it to an instance of Problem 1 in the following manner: (i) for each customer we create the chain DS $\rightarrow$ plant $\rightarrow$ customer where DS is a dummy ingress node, customer is the egress node, and plant is a VNF, (ii) set the user traffic to be equal to the customer demand, (iii) use the transportation cost as the traffic forwarding cost, and (iv) set the installation cost of a VNF instance as the cost of a plant. These operations can be performed in polynomial time of the problem size. If we can solve this instance of problem 1, we will also get a solution for the CPLPSS. However, CPLPSS is NP-hard, so Problem 1 is NP-hard as well.

3 Dynamic Programming Solutions

In this section, we consider two special network models, where we can turn to dynamic programming and obtain the optimal deployment solution. The first one is a sparsely deployed scenario for loose delay constraints, where the instance number of each required VNF is exactly one. The second one is a densely deployed scenario under a uniform-cross-node-link scheme, where we assume the bandwidth unit cost and transfer delay among all EC nodes are identical.

3.1 One-instance Service Chain Deployment

3.1.1 Problem Formulation

In this part, we consider a deterministic-instance-number scenario where the NFV provider only creates an instance for each requested VNF. If the only instance of

f is installed on node n, *i.e.*, $x_{nf} = 1$, then any traffic t will go through f on n, *i.e.*, $t_{nf} = 1, \forall t$. $\sum_t v_t$ units in total will be processed on n. Thus, if the instance of f and its successor f^+ are respectively placed on node n and n', *i.e.*, $x_{nf} = 1$ and $x_{n'f^+} = 1$, then all traffics will be transferred from f on n to f^+ on n'. Thus, $x_{nf}x_{n'f^+}$ indicates whether there exists traffic transfer between n and n'. The corresponding bandwidth cost for all traffic is $\sum_t v_t b_{nn'}$ if $x_{nf}x_{n'f^+} = 1$, otherwise, the cost is 0. Then, the NFV provider's overall cost and the service delay of each traffic can expressed in Eq. (3):

$$\begin{aligned} C &= \mathbf{p}\mathbf{x}\mathbf{r}^\intercal + \mathbf{v}\mathbf{B}^\intercal, \\ d_t &= tr(\mathbf{x}\mathbf{l}). \end{aligned} \tag{3}$$

Accordingly, the optimization problem faced by the NFV provider can be refined as below.

Problem 2 (One-instance SCD)**.**

$$\begin{aligned} &\textit{minimize} && C && \text{(4a)} \\ &\textit{subject to} && d_t \le D \quad \forall t && \text{(4b)} \\ & && \sum\nolimits_n x_{nf} = 1 \quad \forall f && \text{(4c)} \end{aligned}$$

The new constraint (4c) illustrates any required f has exactly one instance in the whole network. By figuring out the optimal x_{nf} for $\forall n, f$, the NFV provider can figure out the most suitable node for each instance in order to minimize his cost while satisfying the service delay constraint.

3.1.2 Dynamic Programming Solution

In the following, we use the dynamic programming method to give an efficient algorithm to solve Problem 2. The problem is broken down into stages and the aim at every stage is to select the optimal decision so that the objective is optimized over the current number of stages. Hence, in each stage we solve only once the corresponding subproblem. The results of each stage are stored and later used to backtrack the optimal values.

Let $A_f(n,d)$ represent the minimal cost of deploying f to the last VNF in the requested service chain, given f is placed on node n and the current accumulated delay is no more than d. We start from the last VNF to get the optimal solution by successively moving to the predecessors.Assuming f is the last VNF, without any successor, the minimal cost of deploying VNF f on the node n can be simply expressed as $A_f(n,d) = r_f p_n$ for $\forall\ d \le D$. For any other f, $A_f(n,d)$ is related to installation cost $r_f p_n$, as well as the minimal costs of connecting f and its succeeded service chain. The cost expression must be extended with these dependencies and the DP approach consequently gives rise to the following recurrence formula:

$$A_f(n,d) = r_f p_n + \min_{n'} \left\{ \sum_t v_t b_{nn'} + A_{f^+}(n', d - l_{nn'}) \right\}.$$

Given f placed in the node n, to find the minimal cost of deploying a sub-service chain (f to the last VNF) within a delay constraint d, equals to find the cost of deploying a sub-service chain (f^+ to the last VNF) in a node n' such that the sum of $\sum_t v_t b_{nn'}$ (as the bandwidth cost from f to f^+) and $A_{f^+}(n', d - l_{nn'})$ (as the minimal deployment cost of the sub-service chain (f^+ to the last VNF) given f^+ placed in the node n' within a delay constraint $d - l_{nn'}$ since the transmission from f to f^+ takes a time of $l_{nn'}$). The DP solution holds a time complexity of $O(|F||N|^2)$ and Algorithm 1 gives the relative details. We also provide an example, shown in Fig. 2, to illustrate how the proposed algorithm proceeds.

Algorithm 1 One-instance Minimal Cost (OMC)

Input: N, T, F, and service delay constraint D

```
pick f from F in a reversed order
    associate a matrix A_f to f (|A_f| = |N| × D)
    if f is the last VNF then
        initialize A_f(n,d) = r_f p_n
    if f is not the last then
        initialize A_f(n,d) = ∞
        for n ∈ N do
            for d = 0 to D do
                update A_f(n,d) using Eq. (??)
C = min{A_f(n,d)}   ∀n, d and f is the first VNF
```

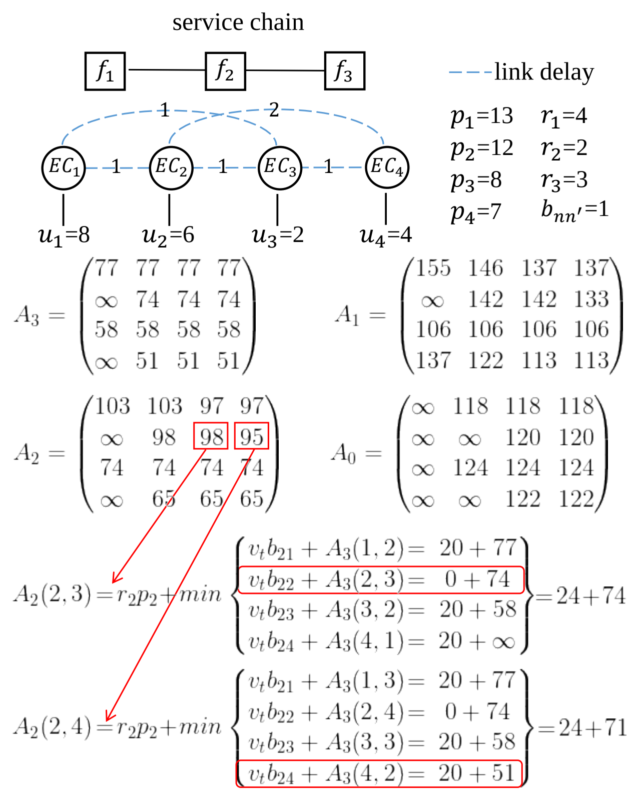


Figure 2: An example using Algorithm 1.

3.2 Uniform-link Service Chain Deployment

Since the one-instance SCD problem may be unsolvable given a harsh delay constraint, we move to a multiple-instance scenario, where one or more instances of f can be installed in different EC nodes to reduce cross-node transfer delay. In this part, we assume connecting

Algorithm 2 Multi-instance Minimal Cost (MMC)

```
Input: N, T, F, and service delay constraint D
pick f from F in a reversed order
    associate a matrix A_f to f (|A_f| = |N| × D)
    if f is the last VNF then
        initialize a A_f(a_f, d) = C_f(a_f)
    if f is not the last VNF then
        initialize A_f(a_f, d) = ∞
        for each a_f = 1 to |N| do
            for each d = 0 to D do
                update A_f(a_f, d) using Eq. (8)
C = min{A_f(a_f, d)}   ∀a_f, d and f is the first VNF.
```

links among all EC nodes are identical, *i.e.*, identical bandwidth cost as b and identical latency as l, so that we can apply dynamic programming to obtain optimal solutions.

3.2.1 Single VNF Optimal Placement

Given that the instance number of each f can be different, user traffic may sometimes be grouped together through a node and sometimes be separated among several nodes. Processing traffic t in its origin node s_t, if f is available there, is always more cost-efficient than directing t to any other node.

The delay-minimized deployment solution is to place an instance of f in each node so that any traffic will be processed locally instead of being merged with other traffics. Obviously, the cross-node cost is eliminated and the corresponding deployment cost incurred by installing instances of f is $\sum_{n\in N} r_f p_n$.

Another way to deploy f is to merge all traffics into k parts (where $0 < k < |T|$) and process them in k nodes. With a dedicated selection of k nodes, $C_f(k)$, the minimized cost of installing f with k instances, can be achieved. (There is no need to consider the bandwidth cost and delay under the uniform-link condition.) Thus, we are confronted with a problem: given a deterministic k instances of f, how to pick k EC nodes to install those instances and how to distribute all traffics through k instances at the cost of $C_f(k)$. Without loss of generality, assume that all EC nodes are decreasingly sorted according to traffic unit v_t covered by them and pick the first k nodes as a small set N_k. Our main observation is that for the problem listed above, there always exists an optimal deployment strategy, where each of the first k nodes holds an instance, and t will be locally processed in the node s_t if s_t belongs to N_k, otherwise t will be directed to the node of N_k. Thus, $C_f(k)$ can be calculated as $C_f(k) = \sum_{n\in N_k} r_f p_n$.

3.2.2 Problem Formulation

We apply a decision variable a_f to represent the instance number of f to be installed in the EC network. To capture the delay between EC nodes, we use an indicator function $z(\cdot)$, taking as input a_f and a_{f^+}, and outputs if there exists service delay between some instance f and f^+.

$$z(a_f, a_{f^+}) = \begin{cases} 0 & a_f = a_{f^+} \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

Based on the analysis above, we obtain the expression of C in Eq. (6).

$$C = \sum\nolimits_f C_f(a_f) + \sum\nolimits_{s_t \notin N_{a_f}} b v_t z(a_f, a_{f^+}) \tag{6}$$

The new optimization problem can be formulated as below.

Problem 3 (Multiple-instance SCD).

$$\begin{aligned} &\textit{minimize} && C && \text{(7a)} \\ &\textit{subject to} && \sum\nolimits_f l z(a_f, a_{f^+}) \le D && \text{(7b)} \\ & && a_f \le |N| \quad \forall f && \text{(7c)} \end{aligned}$$

In the multiple-instance scenario, the NFV provider still aims to minimize his deployment cost (reflected by the objective function (7a)) within the service delay bound (captured by the constraint (7b)). The constraint (7c) allows multiple instances of each f installed in the network.

3.2.3 Dynamic Programming Solution

Let $A_f(a_f, d)$ represent the minimal cost of deploying the f-th to the last VNF, given that f is implemented with a_f instances (using the optimal deployment strategy) and the current accumulated delay is no more than d. Similarly, we start from the last VNF to get the optimal solution by successively moving to the predecessors. Here, we directly show the recurrence formula used in our DP solution (shown in Algorithm 4, of which the time complexity is $O(|F||N|^2 D)$):

$$\begin{aligned} A_f(a_f, d) = C_f(a_f) + \min\Big\{ &\sum\nolimits_{s_t \notin N_{a_f}} b v_t z(a_f, a_{f^+}) \\ &+ A_{f^+}(a_{f^+}, d - l z(a_f, a_{f^+})) \Big\}. \end{aligned} \tag{8}$$

4 Bargaining on Unit Resource Price

Previously, we focus on the placement problem as we assume the unit price of resources in the EC node n, *i.e.*, p_n, is given. In this section, we discuss how to determine the price of each resource unit for each EC node. We denote by R the NFV provider's average resource reward per unit, which is the ratio between the deployment reward that the NFV provider receives from users and the total amount of resource units to be consumed by users. We further define c_n as the unit resource cost for EC node n to run any VNF instance deployed by the NFV provider. Based on the cost c_n, n negotiates with the NFV provider about the unit price p_n. To design an efficient and fair pricing mechanism, we formulate a one-to-many bargaining problem and solve it based on Nash Bargaining Solution(NBS) [1].

4.1 One-to-One Unit Price Bargaining

We start with a simple negotiation with exactly one EC node n. In this case, the bargaining problem is a one-to-one bargaining. The unit reward gained by the NFV provider and n are both 0, if they fail to reach an agreement. For an agreement on unit price p_n, $R - p_n$ and $p_n - c_n$ are the potential gains on each resource unit for the NFV provider and n, respectively. The corresponding NBS provides a fair price for both the NFV provider and n, which can be obtained by solving the problem in the following.

Problem 3 (ONE-TO-ONE BARGAIN)**.**

$$\begin{aligned} &\textit{maximize} \quad && [(R - p_n) - 0]\,[(p_n - c_n) - 0] && \text{(9a)} \\ &\textit{subject to} \quad && c_n \le p_n \le R && \text{(9b)} \end{aligned}$$

We denote p'_n as the solution to Problem 3. It is easy to derive $p'_n = (R + c_n)/2$, indicating that if only one EC node is available for negotiation, the NFV provider has to share the same reward with this EC node. Otherwise, it may reject the agreement, leading to the NFV provider gaining nothing.

4.2 One-to-Many Unit Price Bargaining

We further consider a general model with multiple EC nodes. In this case, the NFV provider needs to bargain with every node $n \in N$, on the unit price p_n (hence a one-to-one bargaining), and thus the entire bargaining problem becomes a one-to-many bargaining, consisting of $|N|$ coupled one-to-one bargainings. Accordingly, the one-to-many bargaining solution contains $|N|$ agreement or disagreement outcomes, each associated with a one-to-one bargaining (between the NFV provider and a certain EC node).

4.2.1 Utility Function

In the previous one-to-one bargaining, we use the potential gains $(R - p_n)$ and $(p_n - c_n)$ to model the utility of the NFV provider and an EC node n, respectively. Here, we still assume the NFV provider's utility is a linear function. This assumption is reasonable because the NFV provider bargains with each EC node and the agreement on one node will not impact the agreement on others (since this node may not be chosen by the NFV provider in the placement step given the delay constraint).However, on the EC-node side, the linear utility function is no longer suitable. In the one-to-one bargaining, node n is the only choice for the NFV provider. Upon agreement on p_n, the NFV provider will purchase all resource units he requires from n.

In the one-to-many bargaining, the NFV provider needs to negotiate with all the nodes, and then make his placement decision. That is, except node n, the NFV provider faces many other options when renting resources. Although we take the delay constraint into consideration, the main objective of the placement problem is the cost minimization. Generally, the lower p_n is, the more resource units the NFV provider should rent from n. Thus, n's net profit is related to its unit price p_n and the amount of resources it eventually sells out. According to the law of diminishing marginal utility in economics, we use the iso-elastic utility to capture a risk-averse EC node's utility when bargaining with the NFV provider. The exact function we adopt is the natural logarithm, which describes that, an EC node's marginal happiness decreases when its unit price increases due to its worry about its final sale amount.

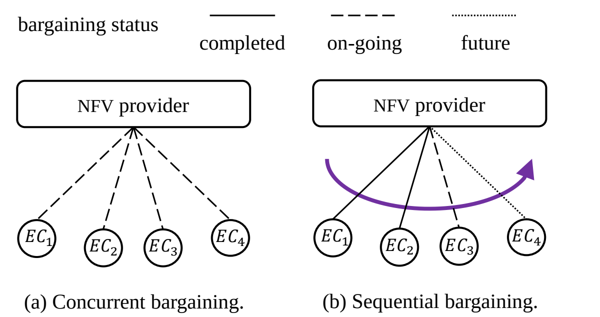


Figure 3: Illustration of different one-to-many bargaining protocols.

4.2.2 Bargaining Protocol

An important issue arising naturally in a one-to-many bargaining is the bargaining dynamics (called bargaining protocol), namely, how the NFV provider bargains with multiple EC nodes. In the following sections, we will focus on two different bargaining protocols, as is shown in Fig. 3: (a) concurrent bargaining, where the NFV provider bargains with all EC nodes simultaneously, and (b) sequential bargaining, where the NFV provider bargains with all EC nodes sequentially in a predefined order. We also elaborate on how a predefined order will affect the NFV provider's utility, in order to find the optimal bargaining order(s). Note that we adopt a simple model in this paper, *i.e.*, all EC nodes do not form coalition, although whether or not there exists coalition among EC nodes is another important factor, which can be extremely complex in some scenarios.

5 NBS Under Concurrent Bargaining

Now we consider the case where the NFV provider bargains with multiple EC nodes simultaneously. First, we consider how an EC node n will respond to the price p_n offered by the NFV provider. We will use β_n to denote the probability that node n would agree upon the price p. We can conclude that node n will definitely accept the price when $p_n \ge p'_n$, as this is an offer no less than what it can gain even in the one-to-one bargaining. Given $c_n \le p_n < p'_n$, the willingness of n to accept the price can be basically characterized by the ratio of current unit profit $(p_n - c_n)$ to the maximum possible unit profit $(p'_n - c_n)$. As we mentioned before, if $p_n < c_n$, then node n has no incentive to provide resources to the NFV provider.

Thus, β_n follows a uniform distribution, which can be characterized by the Eq. (10).

$$\beta_n(p_n) = \begin{cases} 0 & p_n < c_n \\ \frac{2(p_n - c_n)}{p'_n - c_n} & c_n \le p_n < p'_n \\ 1 & p \ge p'_n \end{cases} \tag{10}$$

Now, we discuss the utility of each participant in this concurrent one-to-many bargaining. Without loss of generality, we consider the bargaining between the NFV provider and an EC node n for p_n. We analyze their utilities under the outcomes of agreement and disagreement, respectively.

5.0.1 Agreement

If the NFV provider and node n agree on the price p_n, then their individual unit profit is $R - p_n$ and $p_n - c_n$, respectively. Let U_n^1 represent the NFV provider's utility obtained from reaching an agreement with node n, and V_n^1 represent the corresponding utility obtained by node n. Then, $U_n^1 = R - p_n$ and $V_n^1 = ln(1 + p_n - c_n)$.

5.0.2 Disagreement

If the NFV provider and node n fail to reach an agreement, n's utility is still 0, while the disagreement point for the NFV provider changes since it has the opportunity to bargain with other nodes. As long as a node n', other than n, accepts this price, then the NFV provider still holds a chance to successfully deploy his service chain. Thus, the minimum probability that other EC nodes are willing to accept this price can be expressed as $\min_{n' \neq n} \{\beta_{n'}(p_n)\}$. Let U_n^0 and V_n^0 represent the expected utilities of the NFV provider and node i at the disagreement point, then $U_n^0 = (R - p_n) \min_{n' \neq n} \{\beta_{n'}(p_n)\}$ and $V_n^0 = 0$.

5.0.3 NBS

Similarly, we can solve the following maximum problem to find the Nash bargaining solution.

Problem 4a (ONE-TO-MANY BARGAIN: OMB_{CON}).

$$\textit{maximize} \quad (U_n^1 - U_n^0)(V_n^1 - V_n^0) \tag{11a}$$

$$\textit{subject to} \quad c_n \le p_n \le R \tag{11b}$$

By solving Problem 4a for $\forall n \in N$, we will obtain the unit price profile of all EC nodes. However, objective function (11) is so complex that it is infeasible to express its NBS in a symbolic manner. Therefore, we use numerical analysis to find the optimal unit prices for the NFV provider and all EC nodes in the market. As numerical results will later show in Section 8, we find that the optimal unit price is influenced by many factors, *e.g.* unit cost, market demand, .

6 NBS Under Sequential Bargaining

Lots of previous works on the sequential one-to-many bargaining use the accumulated utility to analyze the bargaining evolution. That is, applied in our scenario, the NFV provider's utility under the disagreement at stage n is the accumulated utility obtained via the NBS in the previous $n-1$ stages. However, as we stress above, there is no promise that the NFV provider must purchase resources from an EC node even if they agree on a certain unit price. Thus, the accumulated utility is not applicable here since the decisions made with different nodes should not impact each other. We start from a simple condition where the bargaining order is predefined. Then, we introduce a new definition for the NFV provider, termed as *payoff*. On this basis, we further consider how to coordinate the order to optimize the NFV provider's payoff after sequentially negotiating with all EC nodes.

6.1 Sequential Bargaining With a Predefined Order

Assume the NFV provider will bargain with all EC nodes sequentially based on their indexes. Now, we consider the NFV provider has negotiated with the first $(n-1)$ nodes and is bargaining with n-th node. We analyze their utilities under the outcomes of agreement and disagreement, respectively.

6.1.1 Agreement

If the NFV provider and node n agree on the price p_n, then their individual unit profit is $R - p_n$ and $p_n - c_n$, respectively. Then, $U_n^1 = R - p_n$ and $V_n^1 = ln(1 + p_n - c_n)$.

6.1.2 Disagreement

If the NFV provider and node n fail to reach an agreement, n's utility is still 0, while the disagreement point for the NFV provider can be larger than 0 since there remain some nodes to bargain with. Thus, the NFV provider can calculate the probability $\gamma_{-n}(p_n)$ that, among all unbargained nodes, there is at least one node accepting the price, as is shown in Eq. (12).

$$\gamma_{-n}(p_n) = 1 - \prod\nolimits_{n'=n+1}^{|N|} (1 - \beta_{n'}(p_n)) \tag{12}$$

Let U_n^0 and V_n^0 represent the expected utilities of the NFV provider and node n at the disagreement point, then $U_n^0 = (R - p_n)\gamma_{-n}(p_n)$ and $V_n^0 = 0$.

6.1.3 NBS

Similarly, we can solve the following problem to find i's optimal unit price according to the given order.

Problem 4b (ONE-TO-MANY BARGAIN: OMB_{SEQ}).

$$\textit{maximize} \quad (U_n^1 - U_n^0)(V_n^1 - V_n^0) \tag{13a}$$

$$\textit{subject to} \quad c_n \le p_n \le R \tag{13b}$$

Thus, given a predefined order, each EC node has its optimal unit price p_n^* (which exists, but is hard to be

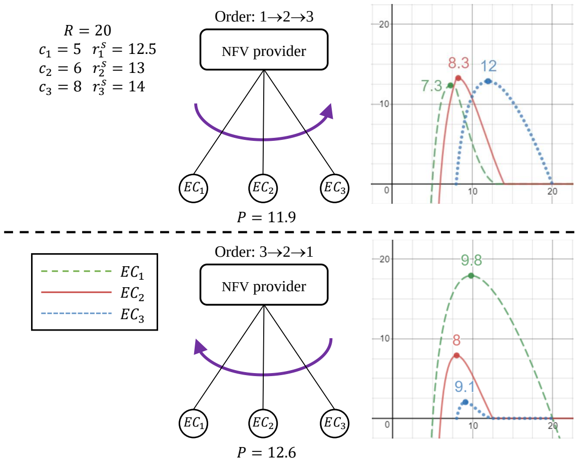


Figure 4: Influence of bargaining order on NBS and NFV provider's payoff.

expressed symbolically). The optimal unit price profile of all nodes forms the NBS of this sequential one-to-many bargaining.

6.2 Order Optimization in Sequential Bargaining

In this part, we study the one-to-many bargaining with endogenous order, where the bargaining order is selected by the NFV provider to maximize his payoff. First, we define the NFV provider's payoff, *i.e.*, his total utility after bargaining with all EC nodes. We also illustrate the influence of the bargaining order on the NFV provider's payoff through two examples. Then, we formulate the NFV provider's optimal bargaining ordering (OBO) problem and find its solution is quite intuitive, *i.e.*, bargaining with nodes in an decreasing order based on their unit resource cost.

6.2.1 NFV Provider's Payoff

We will apply the accumulated price reduction (compared with single one-to-one bargaining outcome p'_n) as the NFV provider's payoff to measure how good a certain one-to-many bargaining outcome is. Let P define the NFV provider's payoff, then $P = \sum_{n=1}^{|N|}(p'_n - p^*_n)$. Based on this definition, we present two examples in Fig. 4 to illustrate that the bargaining order can significantly affect the bargaining solutions and the NFV provider's payoff.

6.2.2 OBO Problem

We use $L = \{l_1, \cdots, l_{|N|}\}$ to denote the bargaining order, *i.e.*, the NVF provider bargains with node l_n at step n. Define $\mathcal{L}$ as the set of all possible bargaining orders, which contains $|N|!$ different L. The NFV provider's objective is to find the optimal bargaining order L^* which maximizes its payoff. Thus, the OBO problem is formulated in the below.

Problem 5 (Optimal Bargaining Ordering: OBO).

$$argmax_{L\in\mathcal{L}} \quad P$$

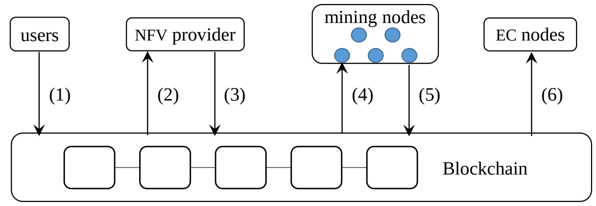


Figure 5: Complete workflow of the CITA-based system.

To solve Problem 5, we may apply the exhaustive search to compute the NFV provider's payoff for each $L \in \mathcal{L}$ and determine L^* accordingly. However, the computational complexity of this method is high. In the following, we prove an important structural property for the one-to-many bargaining, which allows us to find the solution to Problem 4 fast and intuitively.

6.2.3 OBO Solution

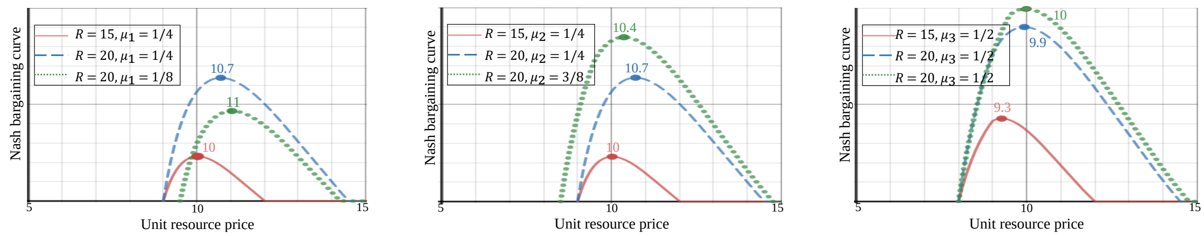
According to our observation, the NFV provider's optimal solution is to bargain with all EC nodes in a descending order in terms of their individual unit resource cost. This solution is quite intuitive as well as reasonable. As the agreed price is definitely higher for a node with a higher cost, if the NFV provider bargains with such a node in the early step, his disagreement point with this node is improved imperceptibly, since he would hold more confidence even if the offer is rejected by this node, because there still exist many other nodes with lower unit cost willing to agree on this price.

Theorem 2. *Given a set of EC nodes N with different unit resource cost c_n for $\forall n \in N$, the optimal bargaining order for the NFV provider is $L^* = \{l_1, \cdots, l_{|N|}\}$, where $\forall i \leq j$, $c_{l_i} \geq c_{l_j}$.*

A: ssume that the optimal solution L contains $i > j$,such that $c_{l_i} > c_{l_j}$. Then we only switch the bargaining order for nodes l_i and l_j, hence obtaining a new sequence L'. Define Δ as the difference between the payoff in the order of L and that in the order of L', then

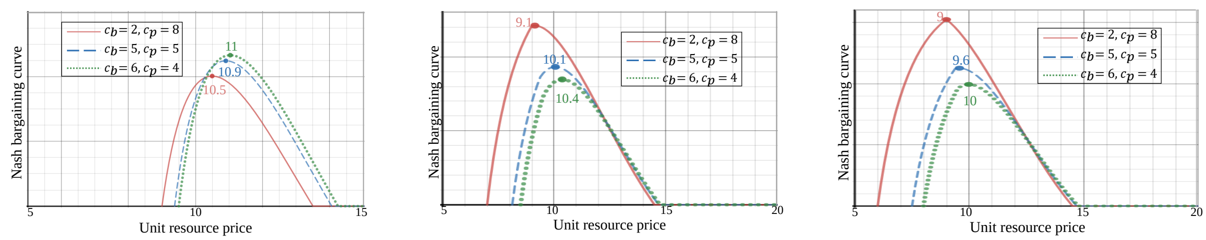
$$\Delta = P - P' = \sum_{n=1}^{|N|}(p'_n - p^*_n) - \sum_{n=1}^{|N|}(p'_n - p^{*\prime}_n)$$
$$= \sum_{n=1}^{|N|}(p^*_n - p^{*\prime}_n) = \sum_{n=l_i}^{l_j}(p^*_n - p^{*\prime}_n). \tag{14}$$

Since switching the order of nodes l_i and l_j won't affect the agreed prices for nodes that the NFV provider bargains with before l_i and after l_j. With the help of MATLAB, we could easily confirm that $\Delta < 0$ always holds. That is, by switching the order of nodes l_i and l_j, we obtain a solution L' better than L and thus L cannot be optimal. Based on the previous analysis, we can conclude that the optimal bargaining ordering cannot contain any $i > j$ such that $c_{l_i} < c_{l_j}$. Thus, L^* must be a decreasing order in terms of unit costs of each EC node.



(a) Node 1's bargaining result. (b) Node 2's bargaining result. (c) Node 3's bargaining result.

Figure 6: Concurrent bargaining given ($M = 3, c_b = 6, c_p = 4$).



(a) Node 1's bargaining result. (b) Node 2's bargaining result. (c) Node 3's bargaining result.

Figure 7: Concurrent bargaining given ($M = 3, R = 20, \mu_1 = 1/8, \mu_2 = 3/8, \mu_3 = 1/2$).

7 CITA-based System

In this section, we introduce the design and implementation of our EC-based NFV market. We apply a blockchain-based system that integrates with the existing edge computing architecture for EC nodes and the scheduler of the NFV provider. The system takes requests to deploy service chains from the NFV provider, and our proposed deployment algorithm runs as the smart contract on the blockchain to determine the optimal instance numbers and the most suitable EC nodes from all candidates. The selected EC nodes receive and enforce the placement decision, while posting the transaction to the blockchain for recording. Our system is implemented through CITA [2], an enterprise-oriented blockchain framework that supports smart contract design and execution.

7.0.1 Entities in the System

Besides three basic entities, *i.e.*, users, EC nodes and the NFV provider, the forth entity is mining nodes, acting as two roles in our designed system. Since each deployment transaction should be processed by all mining nodes, they act as the NFV provider's schedulers, *i.e.*, running the smart contract that implements our algorithms and hence making the deployment decision. Only if the majority of the mining nodes reach agreement on the deployment decision, shall the VNF placement come into force in the selected EC nodes. Thus, the mining nodes also act as monitors to ensure the neutrality of the deployment decision and fairly protect both the NFV provider's and all EC nodes' interests.

7.0.2 Workflow of the System

The complete workflow is shown in Fig. 5: (1) users send service chain requests to blockchain; (2) smart contract takes the request as an input, and invokes NFV provider; (3) NFV provider sends a deployment transaction to the blockchain; (4) mining nodes process the transaction by executing the same smart contract and make a deployment decision based on the majority agreement; (5) the deployment result is recorded in the blockchain as an audit.

8 Evaluation

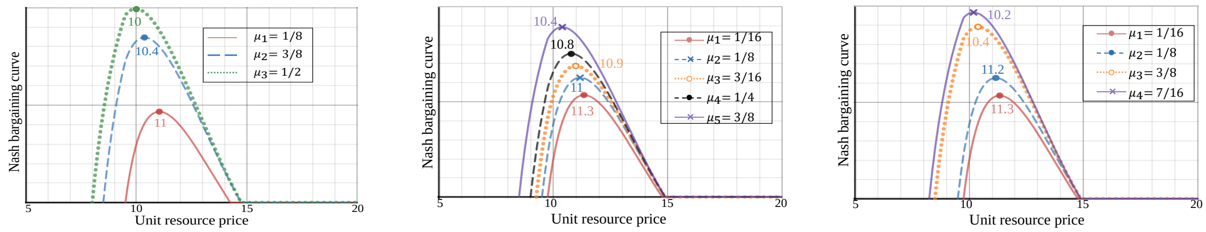
Numerical examples are provided to examine how the optimal unit resource prices can be decided between the NFV provider and all EC nodes through Nash bargaining and how the requested service chain is deployed using our proposed algorithms. This section consists of three parts. The first part focuses on price bargaining. We compare these two bargaining protocols and discuss the suitable scenarios for both of them. In the second part, we show the performance of our deployment algorithms by illustrating the cost reduction of real-world user cases. In the last part, we design different scenarios to run on our system in order to test its applicability.

8.1 Price Bargaining

The bargained price of each EC node is influenced not only by the bargaining protocols, but also by some other parameters. We show how bargaining results change given different parameter values under both protocols, respectively. Then we give comparisons between these two protocols and figure out the selections based on application scenarios.

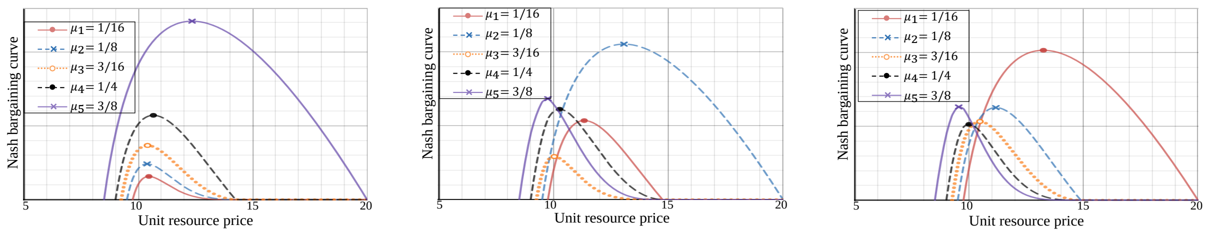
8.1.1 Concurrent Bargaining

In the following, we consider four parameters, *i.e.*, the unit resource reward R, the user distribution μ_n, the



(a) $M = 3$, ($\mu_1 = 1/8, \mu_2 = 3/8, \mu_3 = 1/2$). (b) $M = 4$, ($\mu_1 = 1/16$, $\mu_2 = 1/8, \mu_3 = 3/8, \mu_4 = 7/16$). (c) $M = 5$, ($\mu_1 = 1/16, \mu_2 = 1/8$, $\mu_3 = 3/16, \mu_4 = 1/4, \mu_5 = 3/8$).

Figure 8: Concurrent bargaining given ($R = 20, c_b = 6, c_p = 4$).



(a) Optimal bargaining order: $1, 2, 3, 4, 5$. (b) Random bargaining order: $3, 5, 4, 1, 2$. (c) Worst bargaining order: $5, 4, 3, 2, 1$.

Figure 9: Sequential bargaining given ($M = 5, R = 20, c_b = 6, c_p = 4$).

ratio between base cost c_b and premium cost c_p, as well as the number of all EC nodes $|N|$.

Influence of R and μ_n: We start with a small edge computing network. We use the fixed parameter set $(M, c_b, c_p) = (3, 6, 4)$ and vary the values for R and μ_n where $n \in N$. The corresponding Nash bargaining curves for all nodes are shown in Fig. 6. By comparing the red lines ($R = 15$) and the blue dashes ($R = 20$) in Fig. 6(a), we can conclude that, if the user distribution keeps unchanged, the more unit reward the NFV provider obtains, the higher unit price he is willing to pay to each node. Then we keep $R = 20$ while changing the user distribution to ($\mu_1 = 1/8, \mu_2 = 3/8, \mu_3 = 1/2$). We figure out that any node's optimal unit price is negatively related to the user distribution since the expected unit cost for each node gets higher if it has fewer subscribers. This result is consistent with our theoretical analysis.

Influence of cost ratio: To see how the ratio between c_b and c_p will influence the concurrent bargaining results, we fix the parameter set (M, R) as $(3, 20)$ and further assume the user distribution (μ_1, μ_2, μ_3) as ($\mu_1 = 1/8, \mu_2 = 3/8, \mu_3 = 1/2$). Based on the results shown in Fig. 7, we can easily draw the conclusion that, for each EC node, the more premium is charged to non-subscribers, the higher unit cost it has, consequently leading to a higher unit price.

Influence of node number: Fig. 8 gives bargaining results under three settings where the node number changes. We keep the user distribution $(1/8, 3/8)$ in each setting, and we find the unit price is nearly invariable for nodes accounting for the same traffic percentage. Thus, each node's NBS under the concurrent bargaining is mainly determined by its own unit cost instead of the total node number in the market.

8.1.2 Sequential Bargaining

We focus on the bargaining order. Given 5 EC nodes under the fixed user distribution set ($\mu_1 = 1/16, \mu_2 = 1/8, \mu_3 = 3/16, \mu_4 = 1/4, \mu_5 = 3/8$), we compare 3 different bargaining orders using the same values of R, c_b and c_p. Fig. 9 shows that the cost-decreasing bargaining order is better than a random bargaining solution, while the cost-increasing bargaining order holds the worst performance, in terms of the NFV provider's payoff. This result further confirms our theoretical analysis above is correct.

8.1.3 Concurrent Vs. Sequential

By the comparison of Fig. 8(c) and Fig. 9(c), we find that, in this 5-node example, the concurrent NBS is worse than the worst sequential NBS. Thurs, it is better to apply the sequential bargaining protocol if the cost efficiency is highly desired by a certain NFV provider. However, sequential bargaining takes more time for an NFV provider to reach agreement with all nodes. Thus, if the node number is large and the bargaining time is limited, then the concurrent bargaining protocol is recommended.

8.2 Service Chain Deployment

We first implement the proposed algorithms with Eclipse 4.6 in Java. To demonstrate the performance of our algorithms, we evaluate them over three network models, including M_1 (3 EC nodes), M_2 (5 EC nodes), M_3 (10 EC nodes). We compare OMC and MMC with a benchmark algorithm BA, which deploys the required service chain in each EC node. The performance is measured by using a metric called cost deduction ratio (CDR), ρ. Assume the total cost generated by OMC, MMC and BA is C_1, C_2 and C, respectively, then CDR of OMC is defined as $\rho_1 = (C - C_1)/C$ and CDR of MMC is defined as $\rho_2 =$

Protocol	User number	M_1	M_2	M_3
Concurrent	30	0.194	0.271	0.403
	40	0.221	0.296	0.451
	100	0.332	0.399	0.581
Sequential	30	0.201	0.283	0.415
	40	0.231	0.299	0.463
	100	0.347	0.425	0.601

Table 2 Average ρ_1 with a loose delay requirement

Protocol	Latency (ms)	M_1	M_2	M_3
Concurrent	10	0.071	0.097	0.102
	30	0.113	0.146	0.196
	80	0.221	0.296	0.451
Sequential	10	0.072	0.099	0.113
	30	0.130	0.155	0.204
	80	0.231	0.299	0.463

Table 3 Average ρ_2 given the fixed user number of 40

$(C - C_2)/C$. Table 2 shows the performance of OMC if we assume all users have such a loose delay requirement that a single instance is sufficient. Over each network model, we use different number of users. For each user number, we randomly generate 25 different user distribution sets and the average results are presented under the both bargaining protocols. It is obvious that, the number of available EC nodes plays an important role on the cost deduction due to the significantly reduced instance number. The user number also matters since the more users exist, the higher operating cost it incurs. Then, we analyze the results highlighting the impact of the user latency requirement. As is shown in Table 3, the lower latency bound users require, the more instances the NFV provider needs to install, therefore, the less cost deduction it leads to. Besides, from these two tables, we can further conclude that the sequential bargaining usually gives more cost reduction because the unit prices it yields are better in general, compared with the concurrent bargaining.

8.3 Tests on the Designed System

We further conduct all simulations in Section 8.2 directly using our designed system in order to show its practicality. We record the corresponding time duration of each placement transaction from when it is issued until its decision is made by the system. All time duration ranges from 67s to 5.1min and the average value is around 2.7min. The result is acceptable from an NFV provider's perspective. Thus, we can conclude that our system is applicable in the real world.

9 Related Work

9.1 Network Function Virtualization

The research directions in field of NFV focus on the service chain deployment problem, of which solutions can be divided into two different categories. Approaches [3–7] in the first category solve the problem in two stages : VNFs are initially mapped to substrate nodes then the traffic is steered through chains. They usually suffer from high efficiency loss especially for large input graphs which are hard to map. The second category including [8–10] applies joint optimization solutions, where VNFs are placed and traffic is distributed over them in oneshot. Our solution can be divided into the second category. However, we solve this problem in a multi-node EC network while most previous works consider conventional CC systems.

9.2 Edge Computing

Computation offloading happens in both CC [11, 12] and EC [13, 14], which concerns what/when/how to offload users' workload from their devices to the edge servers or the cloud. One common use case on the EC exploitation is for IoT purposes [15–17]. This paper focuses on computing happening in the edge network, and is distinguished from other works by considering each EC node as a self-interested resource supplier instead of being managed and coordinated in a centralized way [18].

9.3 Blockchain-based System

A blockchain-based system consists of a network of computing nodes, sharing a common data structure (the blockchain) with consensus about the state of this structure. The most prominent examples of such systems are Bitcoin [19] as well as Ethereum [20], a combination of blockchain and smart contract [21]. We also implement our EC-based NFV market in a blockchain platform called CITA [2], an enterprise-oriented blockchain framework that supports smart contract design and execution.

10 Conclusion

In this paper, we consider a cost-minimized service chain deployment problem for an NFV provider in a multi-node edge computing network. We take the user-node subscription and delay requirement into account and then propose two DP-based solutions. Further, we come up with an efficient pricing mechanism for the NFV provider and each EC node to reach an agreement on the unit resource price. We also design a blockchain-based system to execute our proposed pricing mechanism and placement algorithms. Simulation results are provided to demonstrate the efficiency of the proposed algorithms and the applicability of the designed system.

11 Acknowledgements

This research was supported in part by NSF grants CNS 2128378, CNS 2107014, CNS 1824440, CNS 1828363, and CNS 1757533.

References

[1] Nash bargaining solution. https://en.wikipedia.org/wiki/Bargaining_problem#Nash_bargaining_solution.

[2] Cita. https://github.com/cryptape/cita.

[3] Miao Li, Qixia Zhang, and Fangming Liu. Finedge: A dynamic cost-efficient edge resource management platform for nfv network. In *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*. IEEE, 2020.

[4] Hendrik Moens and Filip De Turck. Vnf-p: A model for efficient placement of virtualized network functions. In *10th International Conference on Network and Service Management (CNSM) and Workshop*, 2014.

[5] Omar Houidi, Oussama Soualah, Wajdi Louati, and Djamal Zeghlache. Dynamic vnf forwarding graph extension algorithms. *IEEE Transactions on Network and Service Management*, 2020.

[6] Ali Mohammadkhan, Sheida Ghapani, Guyue Liu, Wei Zhang, KK Ramakrishnan, and Timothy Wood. Virtual function placement and traffic steering in flexible and dynamic software defined networks. In *The 21st IEEE International Workshop on Local and Metropolitan Area Networks*, 2015.

[7] Xin Li and Chen Qian. The virtual network function placement problem. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2015.

[8] Zahra Abbasi, Meral Shirazipour, Attila Takacs, et al. An optimization case in support of next generation nfv deployment. In *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15)*, 2015.

[9] Luhan Wang, Zhaoming Lu, Xiangming Wen, Raymond Knopp, and Rohit Gupta. Joint optimization of service function chaining and resource allocation in network function virtualization. *IEEE Access*, 2016.

[10] Marouen Mechtri, Chaima Ghribi, and Djamal Zeghlache. A scalable algorithm for the placement of service function chains. *IEEE Transactions on Network and Service Management*, 2016.

[11] Niroshinie Fernando, Seng W Loke, and Wenny Rahayu. Mobile cloud computing: A survey. *Future generation computer systems*, 2013.

[12] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 2009.

[13] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. Mobile edge computing: Survey and research outlook. *arXiv preprint arXiv:1701.01090*, 2017.

[14] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 2016.

[15] Yaodong Huang, Jiarui Zhang, Jun Duan, Bin Xiao, Fan Ye, and Yuanyuan Yang. Resource allocation and consensus on edge blockchain in pervasive edge computing environments. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019.

[16] Suhan Jiang, Xinyi Li, and Jie Wu. Hierarchical edge-cloud computing for mobile blockchain mining game. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019.

[17] Xiang Sun and Nirwan Ansari. Edgeiot: Mobile edge computing for the internet of things. *IEEE Communications Magazine*, 2016.

[18] Weisong Shi and Schahram Dustdar. The promise of edge computing. *Computer*, 2016.

[19] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.

[20] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 2014.

[21] Nick Szabo. Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought,(16)*, 1996.