Online NFV-Enabled Multicasting in Mobile Edge Cloud Networks

Yu Ma[†], Weifa Liang[†*], and Jie Wu[‡]

† Research School of Computer Science
The Australian National University

‡ Dept of Computer Science
Temple University, USA

Outline

Background

Mobile devices, including smart phones and tablets, gain increasing popularity as communication tools of users for business, social networking, and personal entertainment. Meanwhile, more and more computation-intensive mobile applications with advanced features are developed for the convenience and experience of users. However, executing computation-intensive applications on mobile devices is heavily constrained by limited resources imposed on mobile devices.

Limitations of Cloud Computing

Although offloading computation-intensive tasks to clouds with rich computing resource can leverage the capabilities of mobile devices significantly, it results in inevitable long communication delays, as clouds usually are located far away from mobile users, this degrades user experience especially for services with stringent delay requirements.

Mobile Edge Computing

Australian National University

Mobile Edge Computing (MEC) provides adequate computation and storage resources to mobile users at the network edge. It has been proposed to respond to delay-sensitive user applications, in which servers (cloudlets) are co-located with access points (APs) for the access of mobile users. Thus, the quality of service and energy consumption on mobile devices can be greatly improved.

Network Function Virtualization (NFV) has been proposed to implement network functions as software components in servers/cloudlets.

- Introduce a new dimension for cost savings on hardware middleboxes
- Enable flexible, faster deployments of network functions

Challenges

- Computation and bandwidth resources constraints on the MEC network.
- To steer the data traffic of a request to go through each network function in its service function chain correctly.
- To make a decision to instantiate a new VNF instance or make use of an existing VNF instance to minimize the operational cost.
- Dynamic NFV-enabled multicast request admissions.

Contributions

Australian National University

- We study the NFV-enabled multicast request admissions in MEC networks with the aim to minimize the admission cost of a single NFV-enabled multicast request admission, or maximize the network throughput through admitting a sequence of NFV-enabled multicast requests dynamically, by taking both resource capacity constraints on cloudlets and links, and the service chain of each request into consideration.
- We first propose an approximation algorithm for the cost minimization of a single NFV-enabled multicast request admission.
- We then devise an online algorithm with a provable competitive ratio for dynamic NFV-enabled multicast request admissions.
- We finally evaluate the performance of the proposed algorithms through experimental studies. The simulation results reveal that the proposed algorithms are very promising.

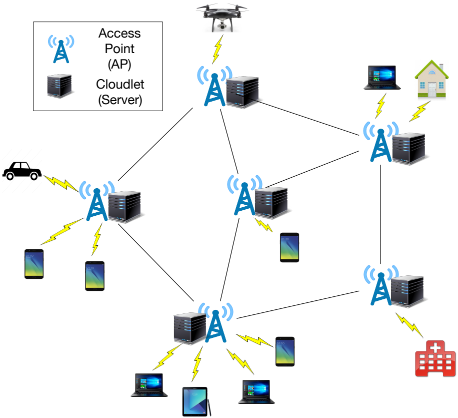


Figure: An MEC network $G = (V, E)$, where V is a set of *access points* (APs) located at different locations, each AP has a co-located cloudlet, E is the set of *links* between APs.

NFV-enabled multicast requests

An NFV-enabled multicast request

User requests arrive dynamically, denote by the jth request as $r_j = (s_j, D_j, SFC_j, \rho_j)$, where

- s_j: the source node,
- D_j: the set of destination nodes,
- SFC_j: type of Service Function Chain required,
- ρ_j: the packet rate.

Each VNF instance has a maximum processing capacity.

If the residual processing capacity of a VNF instance is sufficient to process the data traffic of a request, this VNF instance can be shared by the request. Otherwise, a new VNF instance needs to be instantiated in a cloudlet with sufficient residual computing resource for the admission of the request.

NFV-enabled Multicast Request Implementation

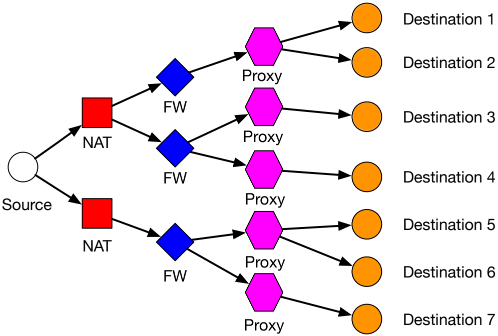


Figure: An example of an NFV-enabled multicast request with a service function chain consists of three network functions, Network Address Translation (NAT), Firewall (FW), and Proxy. Data traffic of the multicast request is transferred from the source node *Source* to a set of 7 destination nodes. Each packet of the request must pass through an instance of network function in its service function chain.

Implementation Cost

- *Instantiation cost*: the instantiation of a VNF instance of network function $f^{(k)}$ in a cloudlet v incurs the instantiation cost $c_{ins}(f^{(k)}, v)$.
- *Processing cost*: the processing of data packet traffic of a request r_j at a VNF instance of $f^{(k)}$ in cloudlet v has the computing resource usage cost $\rho_j \cdot c_{proc}(f^{(k)}, v)$.
- *Communication cost*: the routing cost along a pseudo-multicast tree $T(j)$ in the network from the source node s_j to the destination nodes in D_j, is $\rho_j \cdot c_{bw}(T_j) = \rho_j \cdot \sum_{e \in T(j)} c_e$, where c_e is the unit transmission cost on link $e \in E$.

Definition 1

Given an MEC network $G = (V, E)$ with a set V of cloudlets, each $v \in V$ has computing capacity C_v, each link $e \in E$ has bandwidth capacity B_e. Given an NFV-enabled multicast request $r_j = (s_j, D_j, SFC_j, \rho_j)$, *the NFV-enabled multicasting problem* in G is to find a pseudo-multicast tree for r_j to route its data traffic from the source node s_j to each destination node in D_j while each packet of its data traffic must pass through each VNF instance in its service function chain SFC_j, such that its implementation cost is minimized, subject to the computing and bandwidth capacities on both cloudlets and links in G.

Problem Definition

Australian National University

Definition 2

Given an MEC network $G = (V, E)$ with a set V of cloudlets, each $v \in V$ has computing capacity C_v, each link $e \in E$ has bandwidth capacity B_e. There is a sequence of NFV-enabled multicast requests $r_1, r_2, \ldots r_j$ arriving one by one without the knowledge of future arrivals, *the online multicasting throughput maximization problem* in G is to maximize the number of requests admitted, subject to resource capacities on both cloudlets and links in G.

Approximation Algorithm for the NFV-enabled Multicasting Problem

Two important challenges:

- Given a multicast request, whether it should be admitted or rejected is determined by the availability of its demanded resources in the network;
- Which cloudlets should be selected to implement which network functions in its service function chain, and whether new VNF instances to be instantiated or existing VNF instances can be shared.

Algorithm Overview

We solve the cost minimization problem by reducing it into a multicast tree problem in an auxiliary acyclic graph for the NFV-enabled multicast request.

If there is a multicast tree in the auxiliary graph for the request, there will be sufficient resources in the network to meet the resource demands of the request, and a pseudo-multicast tree for the request can be derived from the multicast tree.

Auxiliary Graph Construction

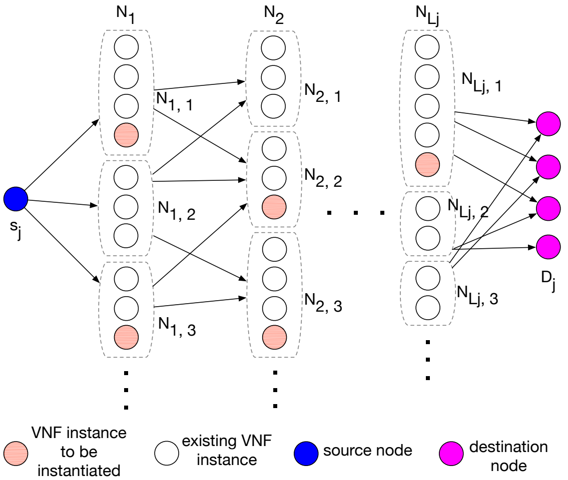


Figure: An auxiliary graph G'_j for NFV-enabled multicast request r_j.

An auxiliary graph G'_j with $L_j + 2$ layers for NFV-enabled multicast request r_j, where $L_j = |SFC_j|$

Layer 0 is the source node s_j.

Layer $L_j + 1$ consists of destination nodes in D_j.

Each layer l, with $1 \leq l \leq L_j$, consists of VNF instances of type $\lambda(j, l)$ that can be employed to process data traffic of request r_j in each cloudlet $v \in V$.

If there is sufficient residual computing resource in a cloudlet, a new VNF instance of that type can be instantiated.

Find a Pseudo-Multicast Tree

Australian National University

Find a directed multicast tree $T'(j)$ in G'_j rooted at s_j and spanning all nodes in D_j, such that the weighted sum of edges in $T'(j)$ is minimized.

A classic Directed Steiner Tree problem.

If a multicast tree $T'(j)$ in G'_j exists, a pseudo-multicast tree $T(j)$ in G rooted at s_j and spanning all nodes in D_j can then be derived, where *a pseudo-multicast tree* in fact is a graph in which nodes and links can appear multiple times.

Approximation Algorithm for the NFV-enabled Multicasting Problem

Australian National University

Input: An MEC network $G = (V, E)$ with a set V of cloudlets, a multicast request $r_j = (s_j, D_j, SFC_j, \rho_j)$.

Output: Admit or reject request r_j. If r_j is admitted, a pseudo-multicast tree $T(j)$ in G is delivered.

Construct auxiliary graph G'_j from G, and assign a cost weight on each edge;
Find an approximate multicast tree $T'(j)$ in G'_j rooted at s_j and spanning all nodes in D_j, by applying any exisiting approxmation algorithm;
if $T'(j)$ exists **then**
    A pseudo-multicast tree $T(j)$ in G is derived, by replacing each edge in $T'(j)$ with a shortest path in G;
    Update residual resource capacities of links, cloudlets, and VNF instances in G;
else
    Reject request r

Australian National University

Theorem

Given an MEC network $G = (V, E)$ with a set V of APs with each attaching with a cloudlet, and a multicast request $r_j = (s_j, D_j, SFC_j, \rho_j)$, there is an approximation algorithm, `Algorithm` *1, for the cost minimization problem with an approximation ratio of $|D_j|^\epsilon$, which takes $O((L_j \cdot |V|)^{\frac{1}{\epsilon}}|D_j|^{\frac{2}{\epsilon}} + |V|^3)$ time, where L_j is the length of service function chain SFC_j of request r_j, and ϵ is a constant with $0 < \epsilon \leq 1$.*

When admiting an NFV-enabled multicast request, if a specific resource has been highly utilized, it should be less used in future, otherwise this will result in a higher cost. On the other hand, if a specific resource has rarely been used, it should be encouraged to use by assigning it a lower cost. To capture the resource usage of request r_j, we use an exponential function to model the cost $W_{v,i}^{(k)}(j)$ of processing its packets at a VNF instance $i \in F_v^{(k)}$ of its SFC as follows,

$$W_{v,i}^{(k)}(j) = \mu^{(k)}(\alpha^{1-\frac{\mu_{v,i}^{(k)}(j)}{\mu^{(k)}}} - 1), \tag{1}$$

where α (> 1) is a tuning parameter to be decided later, and $1 - \frac{\mu_{v,i}^{(k)}(j)}{\mu^{(k)}}$ is the processing capacity utilization ratio in VNF instance i when request r_j is considered, and $\mu^{(k)}$ is processing capacity of the VNF instance.

The cost $W_v(j)$ of instantiating a new VNF instance at cloudlet $v \in V$ and the cost $W_e(j)$ of using bandwidth resource at link $e \in B$ can be defined similarly, that is

$$W_v(j) = C_v(\beta^{1-\frac{C_v(j)}{C_v}} - 1), \tag{2}$$

$$W_e(j) = B_e(\gamma^{1-\frac{B_e(j)}{B_e}} - 1), \tag{3}$$

where β (> 1) and γ (> 1) are tuning parameters to be decided later, and $1 - \frac{C_v(j)}{C_v}$ and $1 - \frac{B_e(j)}{B_e}$ are the resource utilization ratios in cloudlet v and in link e, respectively, when request r_j is considered.

Australian National University

The normalized usage cost of each VNF instance $i \in F_v^{(k)}$ in cloudlet v for request r_j is defined as

$$\omega_{v,i}^{(k)}(j) = W_{v,i}^{(j)}(j)/\mu^{(k)} = \alpha^{1-\frac{\mu_{v,i}^{(k)}(j)}{\mu^{(k)}}} - 1. \quad (4)$$

Similarly, the normalized usage costs $\omega_v(j)$ at each cloudlet $v \in V$ and $\omega_e(j)$ at each link $e \in E$ for request r_j are defined as follows,

$$\omega_v(j) = W_v(j)/C_v = \beta^{1-\frac{C_v(j)}{C_v}} - 1, \quad (5)$$

$$\omega_e(j) = W_e(j)/B_e = \gamma^{1-\frac{B_e(j)}{B_e}} - 1. \quad (6)$$

Australian National University

For each NFV-enabled multicast request r_j, we construct an auxiliary graph similar as the one for the NFV-enabled multicasting problem.

The difference is the weight assigned to each directed edge in the auxiliary graph is the sum of three normalized usage costs.

we adopt the following admission control policy:

If (i) the sum of normalized usage costs of the VNF instances in its service function chain of request r_j is greater than σ_1, i.e., $\sum_{v \in V} \sum_{l=1}^{L_j} \sum_{i \in F_v^{(\lambda(j,l))}} \omega_{v,i}^{(\lambda(j,l))}(j) > \sigma_1$, where $L_j = |SFC_j|$;

or (ii) the sum of normalized usage costs of VNF instantiation for request r_j is greater than σ_2, $\sum_{v \in V} \omega_v(j) > \sigma_2$;

or (iii) the sum of normalized usage costs of links for request r_j is greater than σ_3, $\sum_{e \in E} \omega_e(j) > \sigma_3$,

request r_j will be rejected,

where σ_1, σ_2, σ_3 are the admission control thresholds of resource usages in VNF instances, cloudlets, and links, respectively, where $\sigma_1 = \sigma_2 = \sigma_3 = n$, and $n = |V|$. The detailed algorithm is given in `Algorithm` 2.

Online Algorithm for the Online Multicasting Maximization Problem

Input: An MEC network G and a sequence of multicast requests.

Output: A solution to maximize the network throughput. If r_j admitted, a routing multicast tree for r_j will be delivered.

while request r_j arrives **do**

  Construct the auxiliary graph G'_j, assign weight to each edge in E'_j;

  Find an approximate multicast tree $T'(j)$ in G'_j rooted at s_j and spanning all nodes in D_j,

  if $T'(j)$ does not exist **then**

    Reject multicast request r_j;

  else

    Determine whether r_j should be accepted by the admission control policy;

    if r_j is admitted **then**

      A pseudo-multicast tree $T(j)$ in G is derived, by replacing each edge in $T'(j)$ with its corresponding set of edges in G;

      Update residual resource capacities of VNF instances, links and cloudlets in G;

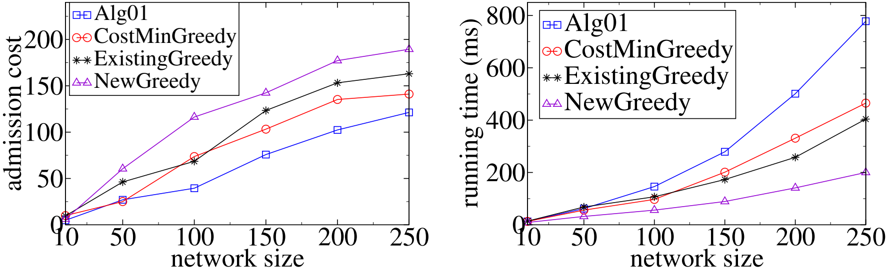
Australian National University

Theorem

Given an MEC network $G = (V, E)$ with a set V of APs in which each $v \in V$ is attached with a cloudlet with computing capacity C_v, each link $e \in E$ has bandwidth capacity B_e, there is an online algorithm, `Algorithm` *2, with competitive ratio of $O(\log n)$ for the online multicasting throughput maximization problem, and the algorithm takes $O((L_j \cdot |V|)^{\frac{1}{\epsilon}} |D_j|^{\frac{2}{\epsilon}})$ time to admit each request r_j when $\alpha = \beta = \gamma = O(n)$, where $n = |V|$, $L_j = |SFC_j|$, and ϵ is a constant with $0 < \epsilon \leq 1$.*

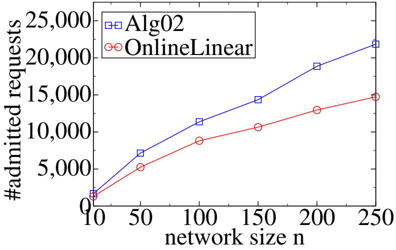
Experimental Environmental Setting

- The commonly used GT-ITM tool was used to generate network topologies.
- The computing capacity of each cloudlet, the number of cloudlets in the generated networks, the types and computing demands of network functions, and the cost of VNF instance implementation and link usage were adopted from prior studies.
- Each user request was generated by randomly picking an AP node as its source node, a set of AP nodes as its destinations, and assigning it a service function chain demand and packet rate consistent with existing studies.
- The running time was obtained based on a machine with a 3.4 GHz Intel i7 Quad-core CPU and 16 GB RAM.

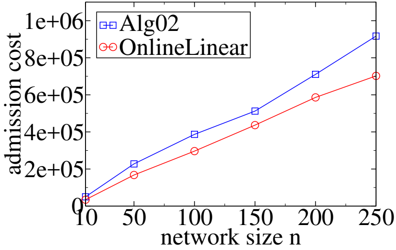


(a) The admission cost (b) The running time

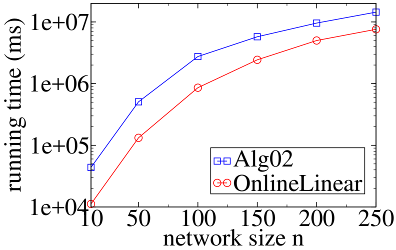
Figure: Performance of `Algorithm 1`, `CostMinGreedy`, `ExistingGreedy`, and `NewGreedy`.



(a) The network throughput

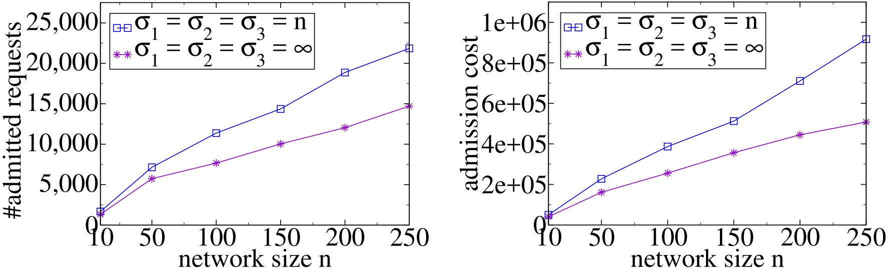


(b) The admission cost



(c) The running time

Figure: Performance of `Algorithm 2` and `OnlineLinear`



(a) The network throughput (b) The admission cost

Figure: Impact of the admission control policy on the performance of `Algorithm` 2.

Conclusions

- We studied the online NFV-enabled multicast request admissions in a mobile edge cloud network.
- We first proposed an approximation algorithm for finding a minimum cost multicast tree for a single request.
- We then devised an online algorithm with a provable competitive ratio for the online throughput maximization problem where NFV-enabled multicast requests arrive one by one without the knowledge of future request arrivals.
- We finally evaluated the performance of the proposed algorithms through experimental simulations. Experimental results demonstrate that the proposed algorithms are promising, and outperform their theoretical counterparts.

Q&A