Throughput Maximization of NFV-Enabled Unicasting in Software-Defined Networks

Weifa Liang

Research School of Computer Science
The Australian National University
Canberra, ACT 2601, Australia

Outline

Hardware-Implemented Network Functions and Their Limitations

Modern communication networks rely on hardware middleboxes to implement *Network Functions* such as firewalls, intrusion detection system, traffic compression, and so on, to ensure the security and performance of data transfers.

Limitations of Hardware-implemented Network Functions

Provisioning dedicated hardware for many different types of network functions however is expensive, due to the increased cost of purchasing and managing the hardware. Also, with more and more new developed network functions, it takes quite a while to deliver dedicated midddleboxes for their implementations.

Network Function Virtualization

Network Function Virtualization (NFV) has been proposed to implement network functions as *software components* in servers or data centers.

- Introduces a new dimension for cost savings on hardware
- Enables flexible, faster deployments of new network functions

Next-Generation Networking Paradigm

Along with the NFV technology, Software-Defined Networking (SDN) has been envisioned as another disruptive technology, which separates the control plane from the data plane, easing network management and simplifying network configurations.

In this paper we study the dynamic admissions of *unicast requests* with *policy-enforcement requirements* in an SDN, by implementing the policies through Network Functions Virtualization in data centers. We term such requests as *NFV-enabled unicast requests*.

Challenges

- To admit a given NFV-enabled unicast request, we must determine not only *a routing path* for its data traffic but also *which data centers to include* in the routing path for its NFV implementation
- Since NFV-enabled requests arrive into the system *dynamically and unpredictably*, the response to each incoming request by either admitting or rejecting it is crucial to maximizing the network throughput
- Dynamic nature of resource allocations

Main Contributions

Australian National University

- We formulate novel optimization problems for NFV-enabled unicasting in SDNs, and propose a generic optimization framework for the admissions of NFV-enabled requests with and without end-to-end delay constraints
- We devise efficient algorithms for minimizing the implementation cost of each NFV-enabled request in terms of computing and bandwidth resource consumptions
- We propose an online algorithm with a provable competitive ratio for the problem without the end-to-end delay constraint, and a fast online algorithm otherwise
- We finally evaluate the performance of the proposed algorithms through experimental simulations. Experimental results demonstrate that the proposed algorithms are promising and outperform existing heuristics.

System Model

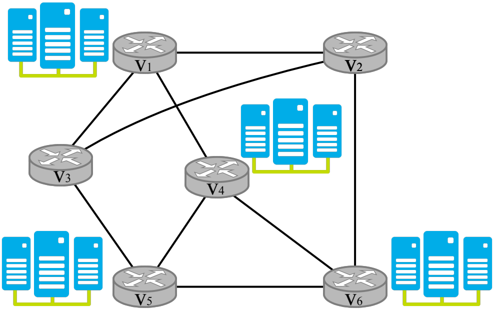


Figure : An SDN G with a set $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ of SDN-enabled switch nodes and a subset $V_S = \{v_1, v_4, v_5, v_6\}$ ($V_S \subseteq V$) of switch nodes attached with data centers (servers) that can implement network functions

User Requests

An NFV-enabled unicast request

As requests dynamically arrives in the system, denote by the kth request as $\rho_k = (s_k, t_k; SC_k, b_k)$, where

- s_k: source node
- t_k: destination node
- SC_k $(= \langle SC_{k,1}, SC_{k,2}, \ldots SC_{k,l} \rangle)$
  a sequence of network functions that each packet must pass through in the chain one by one in order
- b_k: bandwidth requirement.

Roles of an SDN switch in Admitting a Request

If a switch has not any data center attached, the switch serves only as a *routing switch*.

Otherwise, the switch can serve two roles:

- One serves as a regular *routing switch*
- Another serves not only as a routing switch but also as its attached data center for implementing some (or all) of the network functions in SC_k.

Implementation Costs

Assuming that c_e and c_v represent the costs of using one unit of bandwidth resource at link $e \in E$ and computing resource at data center $v \in V_S$, respectively, the *implementation cost* of an admitted request is the sum of costs of computing and bandwidth resource consumptions for the request.

Given an SDN $G = (V, E)$, a subset $V_S \subset V$ of switches attached with data centers each of which has unlimited computing resource, and a unicast request $\rho_k = (s_k, t_k; b_k, SC_k)$, we assume that the network operator of G charges each admitted unicast request on a pay-as-you-go basis, and it focuses on minimizing the implementation cost of each admitted requests.

Problem Definitions

Definition 1

The *online NFV-enabled unicasting problem* in an SDN $G = (V, E)$ with a set V_S of data centers is to admit as many NFV-enabled unicast requests as possible without the knowledge of future request arrivals, subject to the bandwidth capacity constraint at each link in G.

Definition 2

The *online delay-aware NFV-enabled unicasting problem* in an SDN $G = (V, E)$ with a set V_S of data centers is to admit as many NFV-enabled unicast requests as possible without the knowledge of future request arrivals, while meeting the end-to-end delay requirement of each admitted request, subject to the bandwidth capacity constraint at each link in G.

We propose a novel optimization framework that efficiently reduces each of the problems into a problem of finding a (delay-constrained) shortest path in an auxiliary acyclic graph $G'_k = (V'_k, E'_k)$ of each request ρ_k.

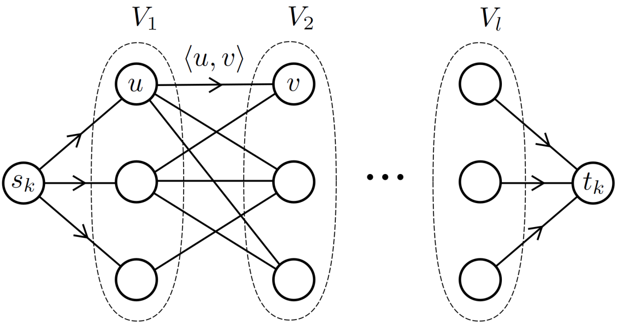


Figure : An auxiliary graph G'_k, where $V_1, \ldots, V_l$ represent the sets of candidate nodes for each service layer in the service chain with $|SC_k| = \ell$

Construct an auxiliary directed acyclic graph $G'_k = (V'_k, E'_k)$ for each request ρ_k, where the cost (or weight) assignment to each node or edge will be jointly determined by the current workload (or the utilization ratio) and the capacity of the node (or edge) dynamically.

For each edge $\langle u, v\rangle \in E$ in G, we have

$$c_{u,v}(k) = B(u,v)(\beta^{1-\frac{B_{u,v}(k)}{B(u,v)}} - 1) \text{ if } \langle u, v\rangle \in E, \qquad (1)$$

where $e = \langle u, v\rangle$, $B_e(k) = B_e(k-1) - b_k$ is the residual bandwidth at link e when ρ_k arrives with $B_e(0) = B_e$, and parameter β is a constant.

The weight of each edge $e' = \langle u', v'\rangle \in E'_k$ of G'_k is defined as *the normalized cost* of the shortest path in G from u' to v'.

Pseudo-code of the Online Algorithm

Australian National University

Input: An SDN $G = (V, E)$ with a set V_S of data centers for implementing network functions, a sequence of unicast requests $\rho_k = (s_k, t_k; b_k, SC_k)$ arrived one by one.

Output: a solution to maximize the throughput, by either admitting or rejecting each incoming request ρ_k. If ρ_k is admitted, a routing walk for it will be delivered.

1: Let G be the subgraph of $G = (V, E)$ after the removal of its edges with residual bandwidth less than $b_k \cdot \ell$ where $\ell = |SC_k|$;

2: Assign each edge in G with the normalized weight (or normalized cost) $w_e(k)$ which is defined in Eq. (1) divided by its bandwidth capacity;

3: Compute all pairs shortest paths in G;

4: Construct $G'_k = (V'_k, E'_k; w_e(k), d_v(k), d_e(k))$, by assigning each edge with the normalized length of the shortest path in G;

5: Find a shortest path P'_{s_k,t_k} in G'_k from s_k to t_k if it exists; otherwise reject ρ_k;

6: A routing walk P_{s_k,t_k} in G is then derived from P'_{s_k,t_k}, determine whether P_{s_k,t_k} should be accepted by the admission policy σ, if not, request ρ_k is rejected.

Algorithm Analysis

Theorem

Given an SDN $G = (V, E)$ *with bandwidth capacity* B_e *for each link* $e \in E$*, a sequence of NFV-enabled unicast requests with the* k*th unicast request* ρ_k *being represented by a quadruple* $(s_k, D_k; b_k, SC_k)$*, and admission control threshold* $\sigma = \ell_{\max} \cdot (|V| - 1)$*, where* $\ell_{\max} = \max\{|SC_{k'}| \mid 1 \leq k' \leq k\}$*, the proposed algorithm has a competitive ratio of* $O(\log |V|)$ *for the online NFV-enabled unicasting problem without the end-to-end delay constraint. The algorithm takes* $O(k(|V|^3 + \ell_{\max}|V|^2))$ *time if the request sequence contains* k *requests.*

Online algorithm with the end-to-end delay constraint

Given a network G and an online request ρ_k, we construct an auxiliary graph $G'_k = (V'_k, E'_k; w_e(k), d_v(k), d_e(k))$.

The only difference between the two online algorithms for dynamic admissions of requests with and without end-to-end delays is the delay constraint.

We now assign each edge an extra metric - the delay, which is the sum of the delays on the edges in $P_{u,v}$, i.e., $d_e(k) = \sum_{e' \in P_{u,v}} d_{e'}$.

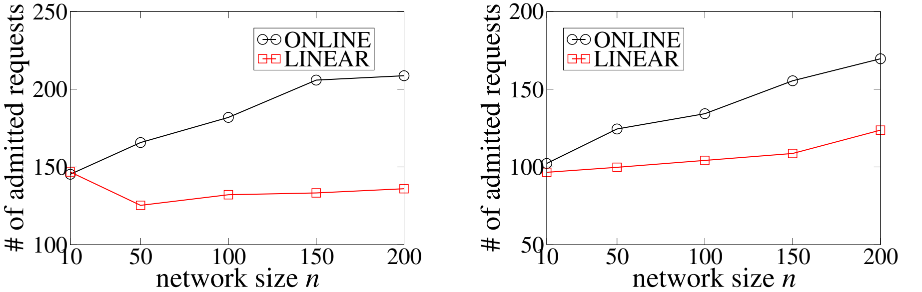
Having constructed G'_k with a cost $w_e(k)$ and a delay $d_e(k)$ for each edge e, we then find a delay-constrained shortest path in G'_k from s_k to t_k for request ρ_k, using the algorithm due to Juttner *et al.*

Experimental environmental setting

Australian National University

- The commonly used GT-ITM tool was used to generate network topologies.
- The bandwidth of each link, the number of data centers, the number of VMs in the generated networks are adopted from, the delay of a link, and the types and computing demands of network functions were adopted from prior studies.
- Each NFV-enabled unicast request was generated by randomly picking two nodes as the source s_k and destination t_k of request ρ_k, respectively, and assigning it a bandwidth demand and delay requirement consistent with existing studies.
- The running time was obtained based on a machine with a 4 GHz Intel i7 Quad-core CPU and 32 GiB RAM.

Performance evaluation of the proposed algorithms

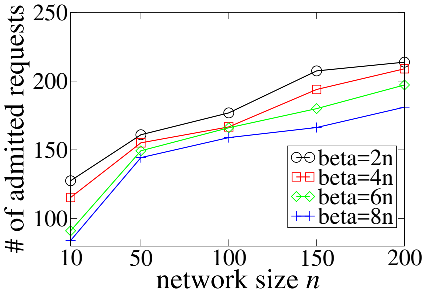


(a) Admission of requests without delay constraints

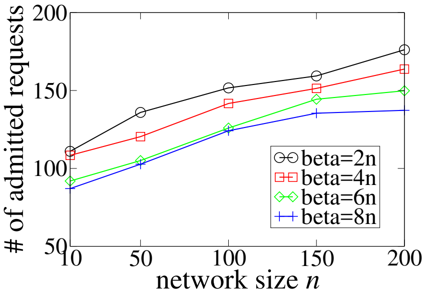
(b) Admission of requests with delay constraints

Figure : The performance of online algorithms `ONLINE`, `ONLINE_DELAY`, and `LINEAR`

Impact of parameters on algorithm performance

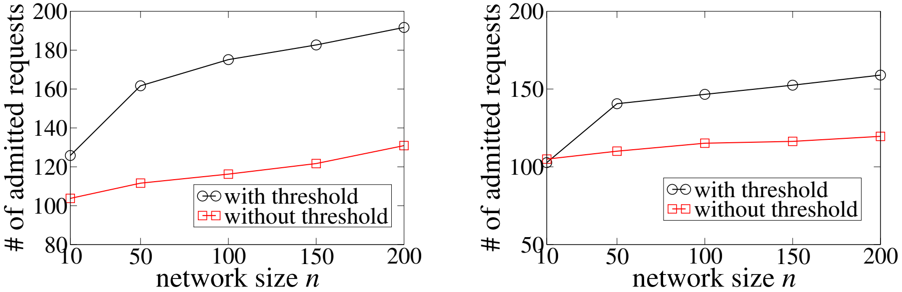


(a) ONLINE



(b) ONLINE_DELAY

Figure : The number of requests admitted by online algorithms ONLINE and ONLINE_DELAY with different values of β



(a) ONLINE (b) ONLINE_DELAY

Figure : The number of requests admitted by online algorithms ONLINE and ONLINE_DELAY with and without the admission control threshold σ

Conclusions

- We studied NFV-enabled unicasting in a Software-Defined Network (SDN) with and without end-to-end delay constraints, for which we proposed a generic optimization framework.
- We then investigated dynamic admissions of NFV-enabled unicast requests without the knowledge of their future arrivals with the objective of maximizing the network throughput, and devised efficient online algorithms for it.
- We finally evaluated the performance of the proposed algorithms through experimental simulations. Simulation results demonstrate that the proposed algorithms are promising, and outperform other heuristics.