

Constructing Multicast Routing Tree for Inter-cloud Data Transmission: An Approximation Algorithmic Perspective

Jun Huang, *Senior Member, IEEE*, Shihao Li, and Qiang Duan, *Senior Member, IEEE*

Abstract—Networking plays a crucial role in cloud computing especially in an inter-cloud environment, where data communications among data centers located at different geographical sites form the foundation of inter-cloud federation. Data transmissions required for inter-cloud federation in the complex inter-cloud networking system are often point-to-multi points, which calls for a more effective and efficient multicast routing algorithm in complex networking systems. In this paper, we investigate the multicast routing problem in the inter-cloud context with K constraints where $K \geq 2$. Unlike most of existing algorithms that are too complex to be applied in practical scenarios, a novel and fast algorithm for establishing multicast routing tree for inter-clouds is proposed. The proposed algorithm leverages an entropy-based process to aggregate all weights into a comprehensive metric, and then uses it to search a multicast tree (MT) on the basis of the shortest path tree (SPT). We conduct complexity analysis and extensive simulations for the proposed algorithm from the approximation perspective. Both analytical and experimental results demonstrate that the algorithm is more efficient than a representative multi-constrained multicast routing algorithm in terms of both speed and accuracy, and thus we believe that the proposed algorithm is applicable to the inter-cloud environment.

Index Terms—Entropy, inter-clouds, multicast tree (MT), routing algorithm, shortest path tree (SPT).

I. INTRODUCTION

CLOUD computing allows a pool of abstracted, virtualized, and scalable computing resources to be delivered on demand as services over the Internet. The past decade has witnessed rapid development in cloud computing technologies and numerous cloud services have been deployed for supporting a wide variety of applications. The traditional cloud service model, where a client utilizes a single cloud data center for access services, is facing some challenges, such as reliability and availability of a single data center, elastic

service provisioning for better service quality, and the potential vendor lock-in issue. Usage of multiple cloud data centers to forms an Inter-cloud environment for further enhancing cloud service provisioning has attracted research interest from both academia and industry.

Networking plays a crucial role in cloud computing especially in an inter-cloud environment, where data communications among data centers located at different geographical sites form the foundation of inter-cloud federation. Due to the very dynamic interactions among data centers for supporting diverse service provisioning, the networking system in an inter-cloud environment form a quite complex system. Data transmissions required for inter-cloud federation in the complex inter-cloud networking system are often point-to-multipoints, for example multimedia content delivery services offered by cloud providers require high performance data transmission from a single data source to multiple destinations. Therefore inter-cloud service provisioning calls for more effective and efficient multicast routing mechanisms in complex networking systems.

Multicast is a networking technology that enables packets to be simultaneously forwarded to multiple destinations from a single data source. Traditional multicast routing techniques implement routing procedures mainly by using the minimum-hop constraint or the shortest path tree (SPT) as the measurement [1]. Compared to traditional network applications, multimedia inter-cloud services bring in more challenging requirements such as mass data traffic, low latency, low packet loss rate, and long duration. Therefore, multicast routing needs to fully consider these new challenging issues.

Constructing multicast trees is a common solution to multicast routing problems. Real-time monitoring and controlling of quality-of-service (QoS) metrics is difficult in the conventional IP-based networks due to the complexity of network control and management. Therefore, multicast algorithms designed for such networking scenarios mainly consider constructing multicast trees with single QoS metric, typically data transmission delay or available bandwidth. Inter-cloud communications between data centers often leverage the software-defined networking technologies, which decouple network control functions from data forwarding operation thus enabling simplified network control with enhanced network manageability. Multicasting algorithms in an inter-cloud networking environment should not only consider more QoS metrics but also be adaptive to the dynamic changes in such QoS metrics. Therefore, the capability of constructing multicast trees with

Manuscript received January 30, 2016; accepted September 20, 2016. This work was supported by the National Natural Science Foundation of China (61309031). Recommended by Associate Editor Jiming Chen. (*Corresponding author: Jun Huang.*)

Citation: J. Huang, S. H. Li, and Q. Duan, "Constructing multicast routing tree for inter-cloud data transmission: an approximation algorithmic perspective," *IEEE/CAA J. of Autom. Sinica*, vol. 5, no. 2, pp. 514–522, Mar. 2018.

J. Huang and S. H. Li. are with School of Computer Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: xiaoniuduan@gmail.com; goliatheden@gmail.com).

Q. Duan is with the Department of Information Science and Technology, Pennsylvania State University, Abington 19001, USA (e-mail: qduan@psu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2017.7510460

multiple QoS constraints in real-time becomes an important requirement for multicast routing algorithms in inter-cloud networks.

In general, there are three different types of the multicast trees: source-based SPT, Steiner tree, and center-based tree. The source-based SPT and the Steiner tree are the two representative ones out of three. A key feature of SPT is its smallest total path weight from the source node to all destination nodes, while Steiner tree minimizes the total weight from the source node to all destination nodes. Since the latter type of tree is known to be NP (non-deterministic polynomial-time)-complete [2], it has attracted considerable research interests during the last decade. Takahashi and Matsuyama [3] proposed a minimum tree-based greedy heuristic to compute a multicast tree whose cost is within twice the cost of the corresponding Steiner tree. Zelikovsky [4] improved this result to about 1.83 times the cost of a Steiner tree. Recently Robins and Zelikovsky [5] presented a new polynomial-time heuristic that achieves the best-known approximation ratio of 1.55 for general graphs. Unfortunately, these works were conducted in the general arbitrarily weighted graphs; and, when viewed from a practical network-studying perspective, the behavior suggested by those arbitrary weights might not occur in real-world networks.

Currently there is a large body of research efforts on seeking an optimized SPT. Bharath-Kumar and Jaffe [6] investigated the minimization of both cost and delay through assuming that the cost and delay functions are the same in a network. Khuller *et al.* developed a simple polynomial-time algorithm in [7] to find a spanning tree that simultaneously approximates a shortest-path tree and a minimum spanning tree. Parsa and others of [8] presented an algorithm for constructing minimum-cost multicast trees which is based on a feasible search optimization strategy that starts with the minimum-delay multicast tree and monotonically decreases the cost by iterative improvements of the delay-bounded multicast tree. Xue [9] extended the idea of [10] for balancing minimum spanning trees and shortest path trees to compute impressive approximations to delay-constrained multicast trees. This algorithm was proven to be able to deliver an outstanding approximation ratio when every edge has the same cost and delay values.

While the above mentioned works mainly focused on constructing an SPT with cost and delay constraints, the case with K constraints ($K \geq 2$) has received limited attention. Multicast routing with K constraints is a challenging issue and deserves thorough investigations since the newly emerged services require the constructed multicast trees satisfy three or even more QoS requirements. We have presented two approximation algorithms for K -constrained multicast routing problem in our recent work [11], but the effectiveness of those algorithms can still be improved. In this paper, we target on resolving this issue based on [11] by designing a more efficient algorithm. Our contributions are summarized as follows.

- 1) We formulate the multi-constrained multicast routing problem and propose a novel and fast algorithm leveraging the entropy technique to address the problem.
- 2) We analyze the performance of the proposed algorithm.

The theoretical results show that the proposed algorithm has lower time complexity than that of a current representative existing algorithm.

3) We conduct intensive numerical experiments to validate the performance of the algorithm. Obtained experimental results indicate that our algorithm outperforms the previous existing algorithm in terms of both speed and accuracy.

The rest of this paper is organized as follows. Section II formally formulates the problems and notations. The proposed algorithm for multi-constrained multicast routing and its relevant analysis are presented in Section III. The experimental results are given in Section IV to examine the efficiency of the proposed algorithm. Section V concludes this paper.

II. PRELIMINARIES

In this section we formulate the problems and define the notions that will be used throughout the paper. Most of definitions and mathematical notations are inherited from [11].

A communication network in an inter-cloud context can be represented by a connected and weighted graph $G(V, E, w_1, \dots, w_K)$, where V is the set of n vertices, E is the set of m edges. Each edge has K independent additive weights¹ w_1, \dots, w_K , in which $w_k(e) \geq 0$ is the k th weight of edge e , $\forall e \in E$, $1 \leq k \leq K$. Let T be a multicast routing tree in $G(V, E)$. Denote $w_k(T)$ as the sum of the k th weights on edges of T , then

Definition 1: Multicast tree (MT). Given a source node $s \in V$ and a set of destination nodes $D \subseteq V$, ($s \notin D$), multicast trees $T(s, D)$ are subgraphs of G whose root node is s and whose leaf nodes form the subsets of D .

For any pair of nodes $u, v \in T$, we use $p(u, v, T)$ to denote the unique path from u to v , and $w_k(p(u, v, T))$ denotes the sum of k th weight of $p(u, v, T)$.

Definition 2: Minimum Steiner tree (MST). When $K = 1$, an MST is a multicast tree in $T(s, D)$ whose weight is minimum among all multicast trees.

Definition 3: Shortest path tree (SPT). When $K = 1$, an SPT is a multicast tree in $T(s, D)$ where $w_1(p(s, v, T))$ is the smallest for every node $v \in D$.

An SPT can be calculated in $O(m + n \log n)$ time, while an MST cannot be computed in polynomial time [2]. This paper mainly focuses on finding an optimal multicast tree with multiple constraints. Specifically, let W_1, \dots, W_K be the K QoS constraints.

Problem 1: Multi-constrained multicast tree (MMT). The problem of MMT is to find a multicast tree in $T(s, D)$ such that $w_k(T) \leq W_k$, $1 \leq k \leq K$.

A multicast tree that satisfies $w_k(T) \leq W_k$, $1 \leq k \leq K$ is said to be a feasible tree. We use $\{T^f\}$ to denote all the feasible trees in $G(V, E)$. For each feasible tree $T_i \in \{T^f\}$, there exists a smallest $\theta_i \in (0, 1]$ such that $w_k(T_i) \leq \theta_i \cdot W_k$, $1 \leq k \leq K$.

Problem 2: Multi-constrained optimal multicast tree (MOMT). The problem of MOMT is to find a multicast tree T_{opt} among all feasible trees $\{T^f\}$ in $G(V, E)$ and the

¹We only consider the additive constraint since the edges with concave constraint can be erased beforehand.

smallest value of θ among all θ_i such that $w_k(T_{\text{opt}}) \leq \theta \cdot W_k$, $1 \leq k \leq K$.

Since MOMT is a combination of the MST problem and the multi-constrained path (MCP) problem, we claim that the problem of MOMT is NP-complete. This is due to the fact that for $D = \{d\}$ the MOMT reduces to the MCP, which is already proved to be NP-complete for $K \geq 2$ additive parameters [12]. On the other hand, for the case $K = 1$, MOMT reduces to the MST, which is also proved to be NP-complete [2]. Therefore, the NP-completeness of MOMT can be seen immediately. The objective of this paper is to find an algorithm to deal with this problem.

Table I lists the frequently used notations in this paper.

TABLE I
FREQUENTLY USED NOTATIONS

| Notations | Meaning |
|-------------------|--|
| K | Number of additive QoS parameters |
| s, D | Source and destinations |
| w_1, \dots, w_K | Edge weights |
| W_1, \dots, W_K | K QoS constraints |
| T | Multicast tree |
| p | Path in the multicast tree |
| MST | Minimum Steiner tree |
| SPT | Shortest path tree |
| MMT | Multi-constrained multicast tree problem |
| MOMT | Multi-constrained optimal multicast tree problem |
| T_{opt} | Optimal multicast tree for MOMT |
| θ | Optimal coefficient for MOMT |

III. ALGORITHM DESCRIPTION AND ANALYSIS

We first present an entropy-based weight aggregation algorithm, then, on the basis of it, we propose an algorithm for establishing the multicast routing tree. Time complexity and approximation analysis are also given for this algorithm in this section.

A. Entropy-based Weight Aggregation

Entropy plays a vital role in multi-criteria decision areas and is widely used for criteria aggregation [13]. Using this technology, the multi-criteria decision problem can be naturally transformed to be a single-criterion decision problem. Here, we adopt the entropy technology to reduce the multi-constrained multicast routing problem. An entropy-based algorithm for weight aggregation, termed EWA, is presented in Algorithm 1.

Algorithm 1 EWA

Input: $G(V, E)$
Output: $G^0(V, E)$

- 1 Initialize $X = [x_{ik}]_{m \times K}$; $f_k^U \leftarrow \max_{1 \leq i \leq m} \{x_{ik}\}$; $f_k^L \leftarrow \min_{1 \leq i \leq m} \{x_{ik}\}$;
- 2 Compute $R = [r_{ik}]_{m \times K}$;
- 3 Calculate $P = [p_{ik}]_{m \times K}$, δ_k , d_k , and α_k ;
- 4 $w^0(e) \leftarrow \sum_{k=1}^K \alpha_k \cdot w_k(e)$;
- 5 **return** $G^0(V, E)$.

In the first step, EWA calculates the matrix X by observing $G(V, E)$ where the element x_{ik} of X denotes the k th weight of the i th edge in $G(V, E)$. In the second step, two parameters are obtained, say f_k^U and f_k^L . Both of them are used in the next step to calculate the matrix R . Since the type of weights on each edge could be either positive or negative [14], we apply f_k^U and f_k^L to calculate different types of weights, that is

$$r_{ik} = \begin{cases} \frac{f_k^L}{x_{ik}}, & x_{ik} \text{ is the negative type weight} \\ \frac{x_{ik}}{f_k^U}, & \text{otherwise.} \end{cases} \quad (1)$$

Obviously $0 \leq r_{ik} \leq 1$, the normalized matrix $P = [p_{ik}]_{m \times K}$ can be computed by

$$p_{ik} = \frac{r_{ik}}{\sum_{i=1}^m r_{ik}}. \quad (2)$$

Then δ_k , d_k , and α_k are calculated by

$$\delta_k = -\frac{1}{\ln m} \cdot \sum_{i=1}^m p_{ik} \ln p_{ik} \quad (3)$$

$$d_k = 1 - \delta_k \quad (4)$$

$$\alpha_k = \frac{d_k}{\sum_{j=1}^K d_j}. \quad (5)$$

With α_k , EWA immediately obtains $w^0(e)$ in Step 5. Then a graph $G^0(V, E)$, which is associated with only one weight $w^0(e)$ of each edge, is returned in the last step.

We now present the algorithm to solve the MOMT problem by using the EWA in the following.

B. Algorithm Description

The proposed algorithm, termed FAMOUS (a fast multiconstrained multicast routing algorithm), is given in Algorithm 2, in which the EWA is called in the first step for transforming the problem to be a simple one. The remaining steps of FAMOUS follow the same philosophy of the shortest path tree, where

$Adj[u]$: the adjacent nodes of u ;

$Plen[u]$: the length of the shortest path from s to u ;

$Tist[u]$: the shortest length from u to the current tree T_m ;

$Parent[u]$: the parent node of u ;

$w^0(e_{s,a})$: the aggregated weight of edge (s, a) .

It is interesting to notice that the obtained multicast tree T_m may be infeasible since the aggregation of the weights leads to some weight which is likely to exceed the constraint. To remedy this situation, we employ a request-filtering process [15] executed in advance to assure that the request, i.e., the constraint is valid. Thus, it is guaranteed that T_m is a feasible multicast tree.

Algorithm 2 FAMOUS

Input: $G(V, E)$, source node s , destinations D .
Output: A multicast tree T_m .

- 1 Call EWA to obtain $G^0(V, E)$;
- 2 Initialize $Adj[u]$, $Plen[u]$, $Tist[u]$ and $Parent[u]$ for node $u \in G^0(V, E)$, clear H ;
- 3 $Plen[s] \leftarrow 0$, $Tist[s] \leftarrow 0$, $Parent[a] \leftarrow \text{null}$;
- 4 **for** $a \in V \setminus \{s\}$ **do**
- 5 **if** $a \in Adj[s]$ **then**
- 6 $Plen[a] \leftarrow w^0(e_{s,a})$; $Tist[a] \leftarrow w^0(e_{s,a})$;
- 7 $Parent[a] \leftarrow s$; $H \leftarrow H \cup \{a\}$;
- 8 **else**
- 9 $Plen[a] \leftarrow \infty$; $Tist[a] \leftarrow \infty$; $Parent[a] \leftarrow \text{null}$;
- 10 **end**
- 11 **while** D is not null **do**
- 12 Select a node $g \in H$ whose
 $plen[g] = \min_{m \in H} \{Plen[m]\}$;
- 13 **if** $g \in D$ **then**
- 14 Add node g to T_m and build the shortest path
 from s to g ; remove g from D ; $Tist[g] \leftarrow 0$;
- 15 **end**
- 16 **for** $h \in Adj[g]$ **do**
- 17 $H \leftarrow H \cup \{h\}$ whose $Plen[h] = \infty$;
- 18 **if** $Plen[h] > Plen[g] + w^0(e_{g,h})$ **then**
- 19 $Plen[h] \leftarrow Plen[g] + w^0(e_{g,h})$;
- 20 $Tist[h] \leftarrow w^0(e_{g,h})$; $Parent[h] \leftarrow g$;
- 21 **end**
- 22 **if** $Plen[h] = Plen[g] + w^0(e_{g,h})$ **and**
 $Tist[h] < Tist[g] + w^0(e_{g,h})$ **then**
- 23 $Tist[h] \leftarrow Tist[g] + w^0(e_{g,h})$;
- 24 $Parent[h] \leftarrow g$;
- 25 **end**
- 26 **end**
- 27 **end**
- 28 **return** T_m ;

Fig. 1 provides an illustrative example of the execution of FAMOUS, where s is the source and $d1$, $d2$, $d3$ are destinations. Fig. 1(a) shows the original graph $G(V, E)$ and the weights associated with each edge. Fig. 1(b) gives the graph $G^0(V, E)$ with the aggregated weight on each edge obtained by EWA. Fig. 1(c) depicts the process of adding the destination node $d1$ to the multicast tree. It seems that Fig. 1(d) shows the situation of adding $d3$ to join the tree. Fig. 1(e) describes how the last destination node $d2$ joins the tree whereby the multicast tree T_m is eventually formed as shown in Fig. 1(f).

It is worth noting that the constraints are not aggregated in the proposed algorithm but are used to examine the feasibility of the obtained solution. We did not consider the case where the weight of the obtained solution violates the corresponding constraint. Investigating the relationship among constraints and their weight configuration are interesting topics for our future work.

C. Analysis

Theorem 1: The time complexity of FAMOUS is $O(km + n \log n)$.

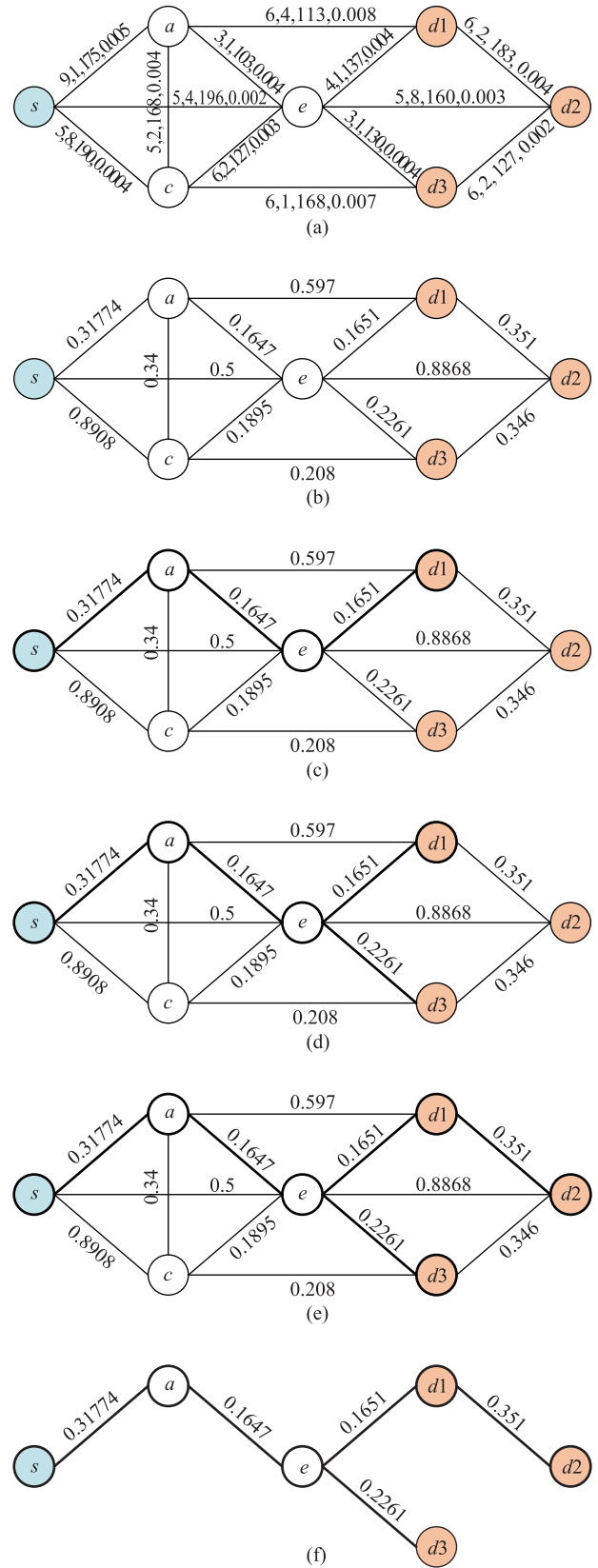


Fig. 1. An illustrative example of FAMOUS execution.

Proof: In Line 1, calling EWA takes $O(Km)$ time to calculate the aggregated weight for each edge. Line 2 initializes the adjacent information for each node that needs $O(n)$ time. The for-loop from Line 4 to Line 10 consumes $O(n)$ for marking

the neighbors of the source node. In the while-loop, selecting an appropriate node from the rest of node set can be done within $O(n \log n)$ time, while the if-statement costs constant time. The relaxation operation from Line 16 to Line 23 requires $O(m)$ time. That is to say, the entire while-loop takes $O(m + n \log n)$. Therefore, the time complexity of FAMOUS is $O(Km + n \log n)$. ■

Theorem 1 indicates that FAMOUS has much lower time complexity compared with a representative algorithm HeurMOMT in [11]. This is due to the facts that the employed EWA is able to simplify the problem efficiently by aggregating the weights and that FAMOUS does not need to check the feasibility of the tree, which lead to a significant reduction in time complexity. For the space complexity of FAMOUS, it is not so hard that we leave it to readers. Next, we analyze the asymptotical approximation performance of FAMOUS.

Theorem 2: FAMOUS finds a ρ -approximation for MOMT, where $\rho = \min_{1 \leq k \leq K} \{1/\alpha_k\}$.

Proof: For the optimal multicast tree T_{opt} , we have

$$w_k(T_{\text{opt}}) \leq \theta \cdot W_k \quad (6)$$

where $1 \leq k \leq K$, that is

$$\sum_{e \in T_{\text{opt}}} w_k(e) \leq \theta \cdot W_k. \quad (7)$$

Assume $W^m = \min_{1 \leq k \leq K} \{W_k\}$, then $\forall k \in [1, K]$, the above inequality can be rewritten as

$$\sum_{e \in T_{\text{opt}}} w_k(e) \leq \theta \cdot W^m. \quad (8)$$

By multiplying α_k on each side and summing the above equation over K weights we can get

$$\sum_{e \in T_{\text{opt}}} \sum_{k=1}^K \alpha_k \cdot w_k(e) \leq \theta \cdot \sum_{k=1}^K \alpha_k \cdot W^m. \quad (9)$$

Since $\sum_{k=1}^K \alpha_k = 1$, we have

$$\sum_{e \in T_{\text{opt}}} w^0(e) \leq \theta \cdot W^m \quad (10)$$

that is

$$w^0(T_{\text{opt}}) \leq \theta \cdot W^m. \quad (11)$$

Also, since the SPT T_m is computed via the weight w^0 , we have $w^0(T_m) \leq w^0(T_{\text{opt}})$. Therefore,

$$w^0(T_m) \leq \theta \cdot W^m \quad (12)$$

this implies

$$\sum_{e \in T_m} w^0(e) \leq \theta \cdot W_k. \quad (13)$$

Since $w^0(e) = \sum_{k=1}^K \alpha_k \cdot w_k(e) \geq \alpha_k \cdot w_k(e)$, $1 \leq k \leq K$, then

$$\sum_{e \in T_m} w_k(e) \leq \rho \cdot \theta \cdot W_k \quad (14)$$

i.e.,

$$w_k(T_m) \leq \rho \cdot \theta \cdot W_k. \quad (15)$$

Theorem 2 implies that the approximation ratio of FAMOUS depends on the coefficient of weight aggregation. From an approximation perspective, the quality of solution should be worse than that of HeurMOMT [11] since there is a trade-off between “optimality” and “simplicity”. However, the approximation ratio of FAMOUS is possibly smaller than that of HeurMOMT as they are decided by different types of parameters. In the next section we shall examine the FAMOUS performance through extensive experiments.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the FAMOUS algorithm by comparing its performance against that of a representative multicast routing algorithm HeurMOMT [11] via extensive experiments. We select HeurMOMT as the baseline for comparison because FAMOUS and HeurMOMT fall in the same category – approximation-based algorithms; thus are comparable in terms of both effectiveness and efficiency. There are exact algorithms for solving MOMT problem by searching the Pareto set, which is typically time-consuming or even impossible as it is readily proved to be NP-complete. Therefore, such algorithms are not applicable to the inter-cloud networking scenario, which requires faster and more efficient multicast algorithms. Our algorithm is developed with an approximation perspective to address this challenge. To the best of our knowledge, the most recent approximation algorithms dedicated to MOMT are HeurMOMT and fully polynomial time approximation scheme (FPTAS) published in [11]. The HeurMOMT algorithm runs faster than the FPTAS algorithm and is catered for inter-cloud networking very well. Therefore, the performance of the proposed algorithm is compared against that of HeurMOMT. Our comparison is conducted in two aspects: the speed and the accuracy of the algorithm.

A. Experiment Setup

We adopt the same experiment settings from [11] in order to compare the performance unbiasedly. A set of random network topologies generated by Waxman [16] model is adopted here: the nodes are randomly placed in a one-by-one square, and the probability of creating a link between node u and node v is $\alpha \cdot e^{-d(u,v)/\beta L}$, where $d(u,v)$ is the distance between u and v , and L is the maximum distance between any two nodes. α and β are set to 0.6 and 0.4 respectively to guarantee that each generated topology is a connected graph. The network size used in the experiment ranges from 100 nodes to 500 nodes, and the number of destinations is set to 5 and 10.

Regarding QoS parameters, we set K to 2, 3, and 4, i.e., each edge is associated with two or three or four weights, which are uniformly generated in a given range $[1, 100]$. The constraints are all set to 1000. In our experiments, the average results are reported by running each test instance one hundred times independently, and all the experiments are run on an IBM P4 2.4 GHz PC with 4 GB memory.

Note that the topologies as well as QoS parameters used in the experiments are generated randomly, which are representative enough to test the speed and accuracy of the proposed algorithm. Essentially in a practical inter-cloud environment, multicast routing usually happens in regular topologies as data

■

centers are interconnected by specific networks, which are much simpler than the randomly generated topologies.

In the proposed algorithm the constraints are not aggregated but are used to examine the feasibility of the obtained solution. We did not consider the case where the weight of the obtained solution violates the corresponding constraint. The constraint values used in the experiments were set to be large enough so that generated solution always satisfies the constraints.

B. Performance Metric

The performance metrics used in the rest of this section for evaluating FAMOUS and HeurMOMT are defined as follows.

Definition 4 (Execution Time): It indicates the average running time of an algorithm by its one hundred independent runs. This metric is used for evaluating time cost performance of an algorithm, i.e., speed.

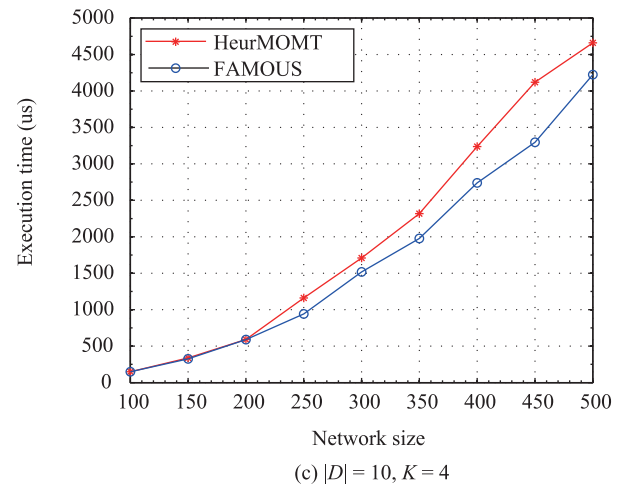
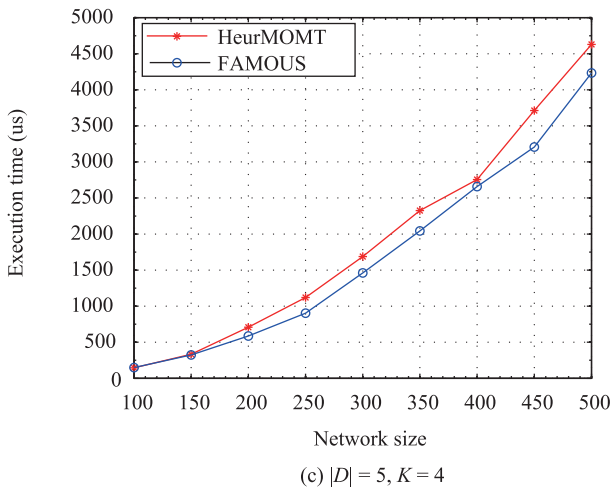
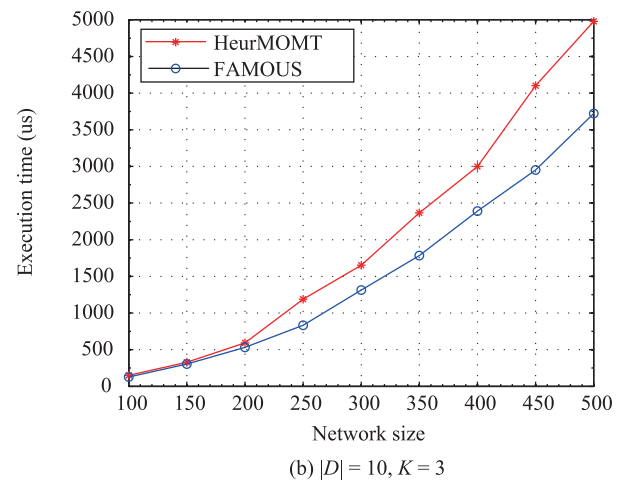
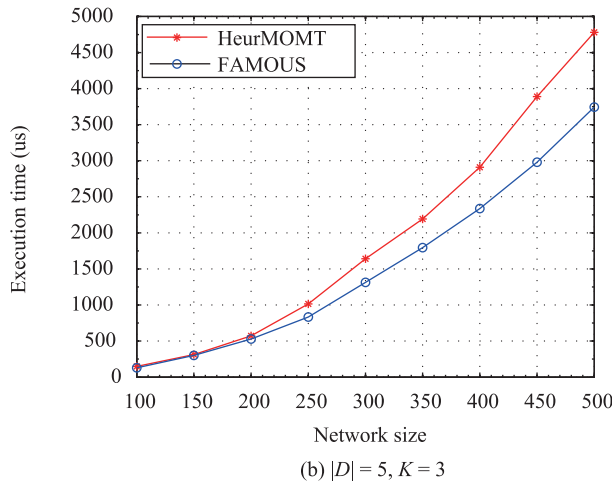
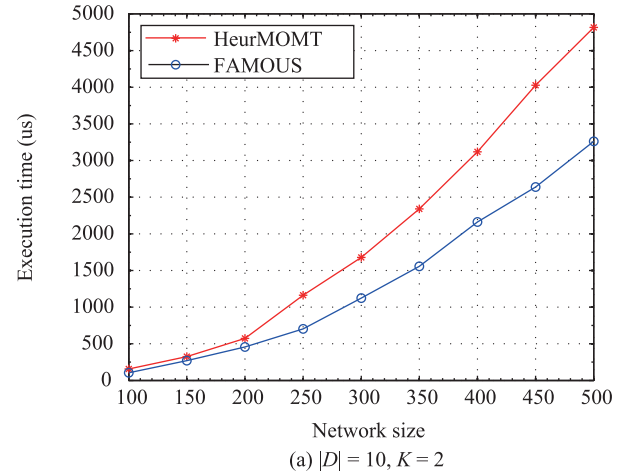
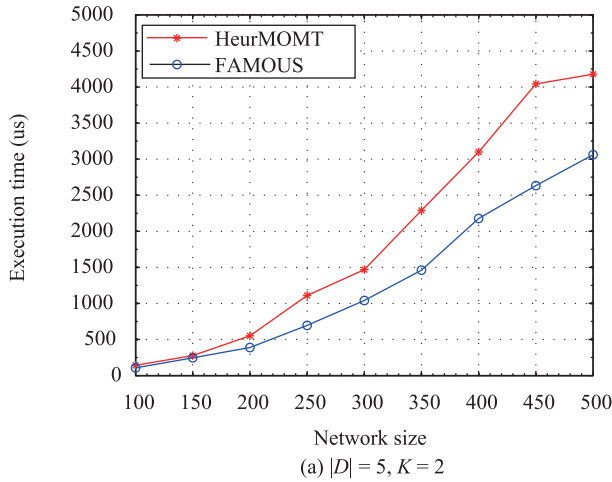


Fig. 2. Execution time comparison.

Fig. 3. Execution time comparison.

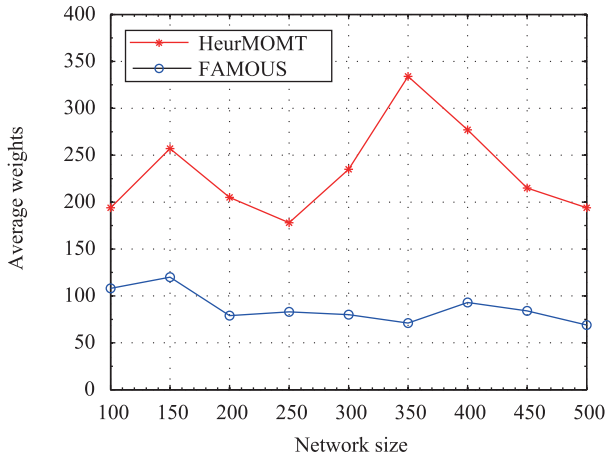
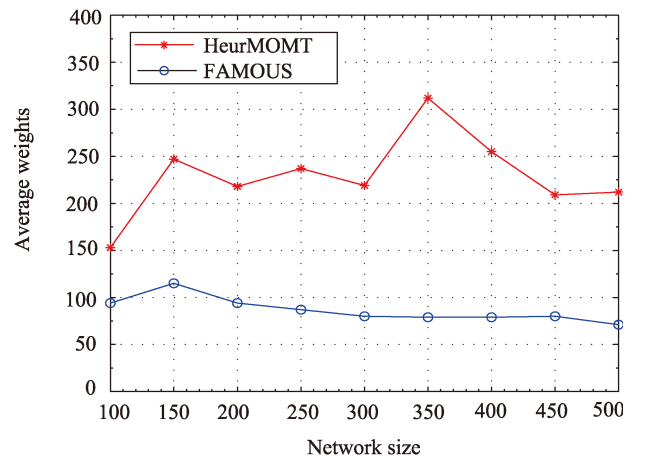
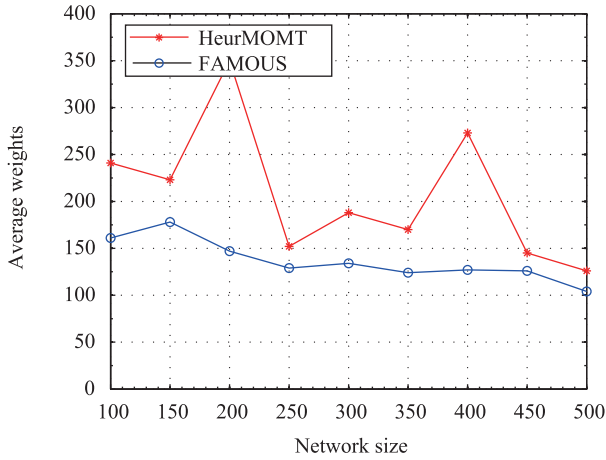
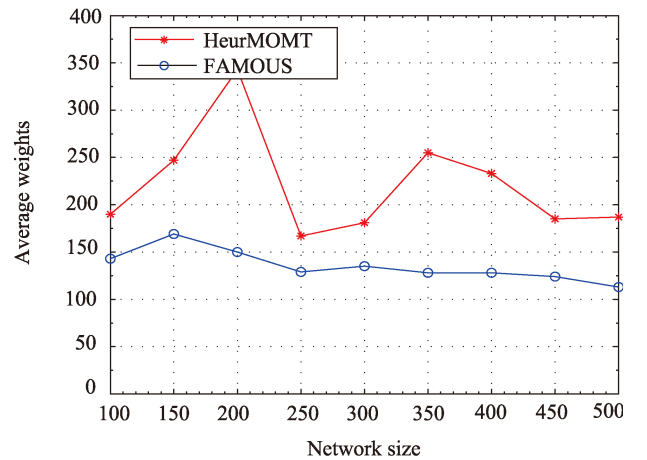
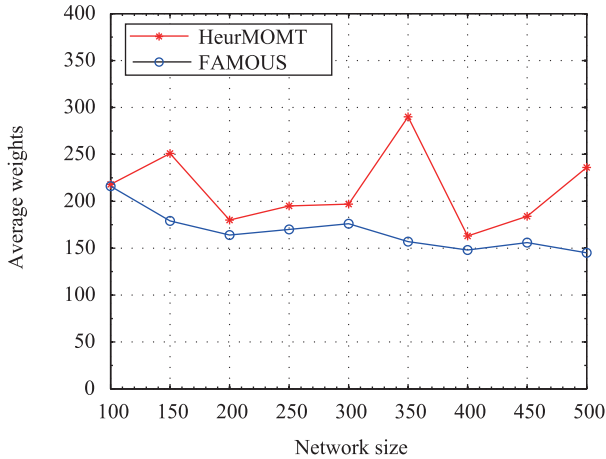
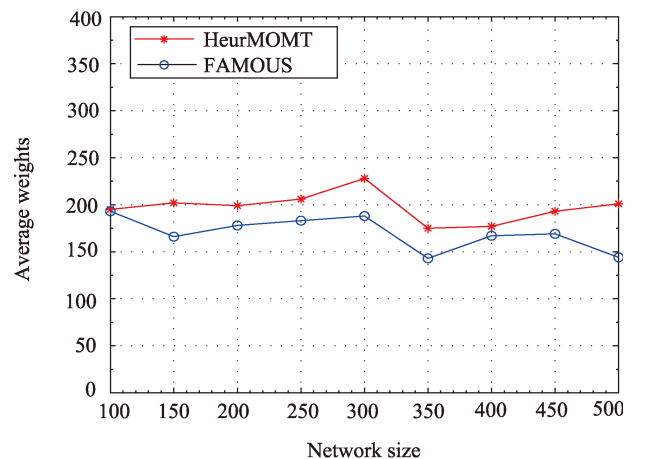
(a) $|D| = 5, K = 2$ (a) $|D| = 10, K = 2$ (b) $|D| = 5, K = 3$ (b) $|D| = 10, K = 3$ (c) $|D| = 5, K = 4$ (c) $|D| = 10, K = 4$

Fig. 4. Average weights comparison.

Definition 5 (Average Weight): It denotes an aggregated weight of a tree returned by the algorithm, that is, for a tree T returned by an algorithm

$$\text{Average weight} = \frac{\left[\sum_{k=1}^K w_k^2(T) \right]^{\frac{1}{2}}}{|D|}.$$

This metric reflects the quality of solution, i.e., the accuracy

Fig. 5. Average weights comparison.

of the algorithm.

C. Results

Figs. 2 and 3 show the execution time of HeurMOMT and FAMOUS with different network sizes when $|D| = 5, 10$, $K = 2, 3, 4$, respectively. It is obvious that the execution time of FAMOUS is lower than that of HeurMOMT, i.e., FAMOUS is more time-efficient. On the other hand, Figs. 4 and 5 plot

TABLE II
RETURNED WEIGHTS COMPARISON WHEN $|D| = 5$

| Network size | HeurMOMT | | | FAMOUS | | |
|--------------|-------------------|-----------------------|---------------------------|-----------------|----------------------|---------------------------|
| | $K = 2$ | $K = 3$ | $K = 4$ | $K = 2$ | $K = 3$ | $K = 4$ |
| 100 | (181, 52) | (112, 136, 144) | (96, 126, 88, 109) | (58, 84) | (70, 115, 80) | (63, 145, 102, 95) |
| 150 | (253, 40) | (130, 123, 111) | (130, 94, 105, 129) | (84, 75) | (94, 98, 101) | (111, 74, 67, 76) |
| 200 | (165, 77) | (161, 193, 216) | (105, 78, 95, 57) | (52, 54) | (76, 92, 69) | (74, 68, 102, 64) |
| 250 | (176, 24) | (119, 58, 70) | (134, 65, 59, 98) | (64, 52) | (85, 73, 57) | (96, 83, 63, 76) |
| 300 | (234, 21) | (111, 75, 89) | (140, 61, 73, 84) | (69, 39) | (67, 60, 82) | (116, 71, 56, 75) |
| 350 | (213, 211) | (139, 64, 52) | (138, 154, 103, 121) | (51, 47) | (65, 91, 38) | (68, 90, 68, 69) |
| 400 | (252, 58) | (260, 54, 52) | (130, 42, 51, 62) | (42, 80) | (54, 71, 84) | (90, 74, 40, 59) |
| 450 | (201, 45) | (129, 59, 42) | (134, 76, 65, 58) | (43, 56) | (66, 71, 62) | (67, 88, 55, 71) |
| 500 | (194, 14) | (114, 40, 27) | (99, 107, 119, 96) | (40, 54) | (78, 49, 35) | (54, 85, 55, 67) |

TABLE III
RETURNED WEIGHTS COMPARISON WHEN $|D| = 10$

| Network size | HeurMOMT | | | FAMOUS | | |
|--------------|-------------------|-----------------------|---------------------------|-----------------|---------------------|--------------------------|
| | $K = 2$ | $K = 3$ | $K = 4$ | $K = 2$ | $K = 3$ | $K = 4$ |
| 100 | (136, 48) | (98, 104, 108) | (78, 99, 96, 99) | (45, 77) | (69, 96, 67) | (62, 112, 98, 91) |
| 150 | (207, 67) | (135, 139, 114) | (106, 83, 91, 97) | (86, 70) | (82, 88, 99) | (96, 73, 72, 71) |
| 200 | (187, 73) | (135, 198, 222) | (130, 77, 95, 69) | (68, 59) | (82, 77, 84) | (88, 75, 92, 77) |
| 250 | (207, 52) | (118, 63, 93) | (132, 81, 70, 94) | (59, 61) | (72, 60, 77) | (96, 85, 64, 92) |
| 300 | (213, 32) | (125, 75, 75) | (144, 89, 92, 92) | (68, 38) | (78, 60, 74) | (98, 80, 97, 71) |
| 350 | (220, 159) | (150, 109, 134) | (112, 90, 61, 58) | (48, 58) | (64, 76, 63) | (68, 87, 54, 56) |
| 400 | (230, 58) | (219, 50, 49) | (126, 55, 59, 77) | (43, 62) | (67, 70, 64) | (71, 65, 71, 91) |
| 450 | (182, 60) | (128, 83, 77) | (135, 81, 71, 56) | (46, 54) | (66, 74, 56) | (80, 80, 60, 81) |
| 500 | (211, 14) | (138, 80, 53) | (87, 104, 95, 80) | (47, 49) | (77, 55, 51) | (66, 74, 54, 66) |

the comparison results regarding the average weights of HeurMOMT and FAMOUS. The results demonstrate that the average weights of FAMOUS is lower than that of HeurMOMT, which means that FAMOUS is able to find a better solution than HeurMOMT.

In order to further evaluate the quality of FAMOUS solutions, we present the individual weight comparisons for two algorithms in Tables II and III. To distinguish the better result, the boldface is exploited to highlight the smaller weight. We can see from these tables that although FAMOUS is not able to excel HeurMOMT in every aspect, FAMOUS generates a more preferable tree than what HeurMOMT does in most cases. This observation matches our previous theoretical analysis. Therefore, we claim, on the basis of the above results, that FAMOUS is superior to HeurMOMT.

V. CONCLUSION

To address the multicast requirement of inter-cloud federation in the complex inter-cloud networking system, in this paper we have proposed a fast multi-constrained multicast routing algorithm with $K \geq 2$ constraints. The proposed algorithm applies the entropy technique to aggregate multiple constraints into a comprehensive metric, dramatically reduces

the complexity of the multi-constrained multicast routing problem, and enables the application of the shortest path tree algorithm to solve the problem. We have presented theoretical analysis on the complexity and approximation of the proposed algorithm and have conducted extensive simulations to evaluate the performance of the algorithm. Both analytical and experimental results have demonstrated that the proposed algorithm is more efficient and superior to a representative multi-constrained multicast routing algorithm in the literature in terms of both speed and accuracy.

As future works, we plan to implement our algorithm in a practical inter-cloud environment and collect the real data trace to further evaluate the performance of the proposed algorithm. Also, the relationship among constraints and their weight configurations are interesting topics that are worth to be investigated.

REFERENCES

- [1] B. Wang and J. C. Hou, "Multicast routing and its QoS extension: problems, algorithms, and protocols," *IEEE Netw.*, vol. 14, no. 1, pp. 22–36, Jan.–Feb. 2000.

- [2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: W. H. Freeman & Co Ltd, 1979.
- [3] H. Takahashi and A. Matsuyama, "An approximate solution for the Steiner problem in graphs," *Math. Japonica*, vol. 24, no. 6, pp. 573–577, Nov. 1980.
- [4] A. Z. Zelikovsky, "An 11/6-approximation algorithm for the network Steiner problem," *Algorithmica*, vol. 9, no. 5, pp. 463–470, May 1993.
- [5] G. Robins and A. Zelikovsky, "Improved Steiner tree approximation in graphs," in *Proc. 11th Annu. ACM-SIAM Symp. Discrete Algorithms*, San Francisco, California, USA, 2000, pp. 770–779.
- [6] K. Bharath-Kumar and J. M. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Trans. Commun.*, vol. 31, no. 3, pp. 343–351, Mar. 1983.
- [7] S. Khuller, B. Raghavachari, and N. Young, "Balancing minimum spanning and shortest path trees," in *Proc. 4th Annu. ACM/SIAM Symp. Discrete Algorithms*, Austin, Texas, USA, 1993, pp. 243–250.
- [8] M. Parsa, Q. Zhu, and J. J. Garcia-Luna-Aceves, "An iterative algorithm for delay-constrained minimum-cost multicasting," *IEEE/ACM Trans. Netw.*, vol. 6, no. 4, pp. 461–474, Aug. 1998.
- [9] G. L. Xue, "Minimum-cost QoS multicast and unicast routing in communication networks," *IEEE Trans. Commun.*, vol. 51, no. 5, pp. 817–824, May 2003.
- [10] F. K. Hwang and D. S. Richards, "Steiner tree problems," *Networks*, vol. 22, no. 1, pp. 55–89, Jan. 1992.
- [11] J. Huang, Y. Tanaka, and Y. Ma, "On approximating a multicast routing tree with multiple quality-of-service constraints," *IEICE Trans. Commun.*, vol. E95.B, no. 6, pp. 2005–2012, Jun. 2012.
- [12] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 7, pp. 1228–1234, Sep. 1996.
- [13] A. Ishizaka and P. Nemery, *Multi-Criteria Decision Analysis: Methods and Software*. New York: Wiley, 2013.
- [14] J. Huang, C. Q. Xu, Q. Duan, Y. Ma, and G. M. Muntean, "Novel end-to-end quality of service provisioning algorithms for multimedia services in virtualization-based future internet," *IEEE Trans. Broadcast.*, vol. 58, no. 4, pp. 569–579, Dec. 2012.
- [15] J. Huang and Y. Tanaka, "A deployable E2E quality-of-service routing algorithm," in *Proc. IEICE Gen. Conf. 2011*, Tokyo, Japan, No. BS-4-8, 2011, pp. S-23–S-24.
- [16] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.



Jun Huang (M'12-SM'16) received the Ph.D. degree (with honor) from the Institute of Network Technology, Beijing University of Posts and Telecommunications, China, in 2012. He is a Full Professor of Computer Science with the Chongqing University of Posts and Telecommunications.

Dr. Huang was a Visiting Scholar in the Global Information and Telecommunication Institute, Waseda University, a Research Fellow in the Electrical and Computer Engineering Department, South Dakota School of Mines and Technology, a Visiting Scholar in the Computer Science Department, the University of Texas at Dallas, and a Guest Professor at the National Institute of Standards and Technology. He received the outstanding service award from ACM RACS 2017, the runner-up of best paper award from ACM SAC 2014, and the best paper award from AsiaFI 2011. He has authored 100+ publications including papers in prestigious journal/conferences such as the *IEEE Network*, the *IEEE Communications Magazine*, the *IEEE Wireless Communications*, the *IEEE Transactions on Broadcasting*, the *IEEE Transactions on Vehicular Technology*, the *IEEE Transactions on Emerging Topics in Computing*, the *IEEE Transactions on Network and Service Management*, the *IEEE Transactions on Sustainable Computing*, the *IEEE Internet of Things Journal*, the *IEEE Transactions on Cloud Computing*, *IWQoS*, *SCC*, *ICCCN*, *GLOBECOM*, *ICC*, *ACM SAC*, and *RACS*. He is an Associate Editor of *IEEE Access* and the *KSI Transactions on Internet and Information Systems*. He guest-edited several special issues on IEEE journals such as the *IEEE Communications Magazine*, the *IEEE Internet of Things Journal*, the *IEEE Access*, etc. He also chaired and co-chaired multiple conferences in the communications and networking areas and organized multiple workshops at major IEEE and ACM events. His current research interests include network optimization and control, machine-to-machine communications, and the internet of things.



Shihao Li received the B.S. degree in computer science from Chongqing University of Posts and Telecommunications, China, in 2014. He is currently working toward M.S. degree at Chongqing University of Posts and Telecommunications. His research interests include cloud computing and NP-hard algorithm design.



Qiang Duan (M'07-SM'17) received the Ph.D. degree in electrical engineering from the University of Mississippi. He holds a B.S. degree in electrical and computer engineering and a M.S. degree in telecommunications and electronic systems. He is an Associate Professor at the Pennsylvania State University Abington College. He has published over 70 journal articles and conference papers and authored 5 book chapters. Dr. Duan is on the editorial boards for more than 10 international research journals and has served on the technical program committees for numerous international conferences including GLOBECOM, ICC, ICCCN, AINA, WCNC. His currently active research interests include future internet architecture, network virtualization, network-as-a-service, software defined network, and cloud computing.