Betweenness Centrality Based Software Defined Routing: Observation from Practical Internet Datasets

KAI WANG, Harbin Institute of Technology, Weihai, China
WEI QUAN, Beijing Jiaotong University, Beijing, China
NAN CHENG, Xidian University, Xi'an, China
MINGYUAN LIU and YU LIU, Beijing Jiaotong University, Beijing, China
H. ANTHONY CHAN, Caritas Institute of Higher Education, Hong Kong, China

Software-defined networking (SDN) enables routing control to program in the logically centralized controllers. It is expected to improve the routing efficiency even in highly dynamic situations. In this article, we make an in-depth observation of practical Internet datasets and investigate the relationship between *betweenness centrality* and *network throughput*. Furthermore, we propose a new routing observation factor, *differential ratio of betweenness centrality* (DRBC), to denote the varying amplitude of betweenness centrality to node degree. We reveal an interesting phenomenon that DRBC is proportional to the routing efficiency when the maximum betweenness centrality varies in a small range. Based on this, a DRBC-based routing scheme is proposed to improve routing efficiency. The experimental results verify that DRBC-based routing can improve the network throughput and accelerate the routing optimization.

CCS Concepts: • **Networks → Control path algorithms**; **Network performance analysis**; *Network design principles*; *Network reliability*;

Additional Key Words and Phrases: Betweenness centrality, routing adjustment, internet datasets, software defined networking

ACM Reference format:
Kai Wang, Wei Quan, Nan Cheng, Mingyuan Liu, Yu Liu, and H. Anthony Chan. 2019. Betweenness Centrality Based Software Defined Routing: Observation from Practical Internet Datasets. *ACM Trans. Internet Technol.* 19, 4, Article 50 (October 2019), 19 pages.
https://doi.org/10.1145/3355605

This work is supported by Shandong Provincial Natural Science Foundation, China (No. ZR2017BF018), National Natural Science Foundation of China (No. 61702439, 61602030), China Postdoctoral Science Foundation (No. 2019T120732), National Key R&D Program (No. 2017YFE0121300).
Authors' addresses: K. Wang, Harbin Institute of Technology, Weihai, China; email: dr.wangkai@hit.edu.cn; W. Quan (Corresponding author), M. Liu, and Y. Liu, Beijing Jiaotong University, Beijing, China; emails: {weiquan, 18111025, 15125028}@bjtu.edu.cn; N. Cheng, Xidian University, Xi'an, China; email: dr.nan.cheng@ieee.org; H. A. Chan, Caritas Institute of Higher Education, Hong Kong, China; email: h.anthony.chan@gmail.com.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1533-5399/2019/10-ART50 $15.00
https://doi.org/10.1145/3355605

1 INTRODUCTION

The Internet has grown to be a huge system, with nearly 65,000 Autonomous Systems (AS) as of August 2019. An AS, the basic unit of the Internet, is further composed of a group of networks operated by the network operators. As the number of ASes increases quickly, the scale of the Internet becomes very large. The Internet must sustainably perform its primary function, routing packets. The reported data show that the number of ASes increases by about 45 K every year [8]. Besides, the practical ASes have a range of variation in size. However, the Internet is not static due to the failure of existing nodes, or the appearance of new ones. In such a huge network, updating operations make the network topology change constantly, which further aggravates the complexity of the Internet. Therefore, how to accurately characterize and model the dynamic attributes of the Internet is very interesting and full of challenges [7, 17].

Network capacity is dependent on the network topology, routing, and node processing capacity, which mainly depends on the computing and forwarding rate. Internet-like complex networks can be measured by many metrics, such as betweenness centrality and clustering coefficiency. In practical Internet, routing protocols are important on specifying how nodes communicate with each other. Each AS has a single defined routing policy. Once a change occurs anywhere in the Internet, the information about this event must diffuse to all ASes, which have to process it to compute new routes distributively and quickly [3]. To achieve this task, different strategies have been put forward, such as congestion control based on routing strategy [22], deflection routing strategy [12], local dynamic routing strategy [35], global dynamic routing strategy [21], and so on. However, most of them are still at the research stage and have not been put into practice.

Particularly, shortest-path algorithms are widely used in existing routing protocols [30, 46]. A survey makes a comprehensive overview on routing optimization for the Internet traffic engineering [34]. In such algorithms, each node attempts to route packets to their destinations over paths of minimum distance and respond to topological changes periodically to adjust routing decisions when traffic changes. Shortest-path routing algorithms have served remarkably well in the network environment where the network topology changes slightly [42]. However, in a dynamic network environment, shortest-path routing algorithms, particularly those that attempt to adapt to traffic conditions, frequently exhibit oscillatory behaviors and cause performance degradation. In some super large-scale and heterogeneous networks like the Internet, the widely used shortest path strategy can easily cause transmission congestion due to the fast-growing network data when the demands are greater than the supply. To this end, many advanced routing algorithms had been well researched and thoroughly investigated [19, 37].

Software Defined Networking (SDN) provides an open platform for developing and managing the novel and advanced routing policies and network enforcement [26, 27]. SDN is a novel networking paradigm that promises to dramatically simplify network management and enable rapid network innovation and evolution [24, 29, 33]. Different from the rigid design of traditional networks, SDN is featured by decoupling the forwarding hardware from control software. The network intelligence can be logically centralized in the software-based controllers in the control plane, and network devices become simple packet forwarding devices in the data plane that can be programmed via an open interface [44]. In this way, SDN promotes to flexibly design an intelligent control engine to achieve better Internet services. Although the field of SDN is quite recent, it is growing at a very fast pace. SDN has been widely extended to many promising areas, such as satellite network and vehicular network [13, 28, 40, 45].

Leveraging the power and flexibility of SDN, there are many researches on using topological measurements to analyze and improve the performance of SDN. For examples, Yoon et al. considered using centrality measure for scalable traffic sampling to decide the traffic sampling rates at the selected switches [41]. This scheme aims to enhance the intrusion detection performance in terms of malicious traffic flows. Hegr et al. introduced a novel metric, Quality of Alternative Paths centrality (QAP), to quantify node surroundings and indicate more robust paths [14]. Rueda et al. analyzed the critical parts of physical topology and selected the best controllers placement in SDN for improving the network robustness to targeted attacks [31]. Kim et al. proposed a logically isolated networking scheme to integrate distributed cloud resources to dynamic and on-demand virtual networking over software-defined wide area network (SD-WAN) [16]. However, these works focus on different technical issues in SDN, such as traffic sampling [41], path robustness [14], attack avoidance [31], and virtual networking [16]. To the best of the authors' knowledge, there is little work focusing on investigating the novel data-driven network patterns to ameliorate the SDN controller to make efficient routing decisions. *Therefore, it becomes very necessary to explore the novel objective network patterns based on the practical Internet datasets, which have a potential to assist to make proper routing decisions as well as reduce the cost of global routing management.*

This work is motivated by one of the most recent works [23], which surveys the large volumes of router reachability data and proposes a useful indicator for understanding the dependence of the AS-level Internet on individual routers. The authors finally quantify the resulting impact of each router outage on global Border Gateway Protocol (BGP) reachability. In this work, we focus on investigating the relationship between the betweenness centrality and routing throughput based on the practical network topology datasets. We propose a new routing observation factor, differential ratio of betweenness centrality (DRBC), and reveal an interesting phenomenon that DRBC is proportional to the routing efficiency when the maximum betweenness centrality varies in a small range. Some primary tests are finally introduced to verify the availability in the routing optimization. The main contributions are summarized as follows:

- We first define a new routing observation factor, differential ratio of betweenness centrality (DRBC, ∇_B), which is represented by making a *Difference* for the betweenness centrality. According to the theory of complex networks, we formulate the models for the DRBC metric and network throughput, respectively.
- We conduct extensive experiments to analyze the relation between DRBC and network throughput based on the practical network topology dataset from CAIDA [4] and try to find whether there is any law the two metrics follow. Interestingly, we find a hidden indicator that the DRBC metric is positively proportional to network throughput on condition: The maximum betweenness centrality of a network graph keeps constant or changes very slightly. This law can be used to indicate routing efficiency intuitively and guide to make routing adjustment.
- We take an application case to show the DRBC-based routing optimization adjustment. We propose a DRBC-based software defined routing scheme, which guides how to adjust routing path to improve the network throughput based on the DRBC metric. Some primary tests verify DRBC can be applied to control routing in the software defined network controllers and DRBC-based routing can accelerate the routing optimization.

This article is organized as follows: Section 2 lists related works briefly. Section 3 introduces the theoretical modeling and analysis. Section 4 describes the big network datasets used in this work. Section 5 presents the data observations and analysis. Section 6 introduces a brief application case of routing control and shows its efficiency. We conclude this article in Section 7.

2 RELATED WORKS

An increasing number of scholars have devoted themselves to the advanced and efficient routing in complex networks to improve the network efficiency and scalability. Kirst et al. proposed a dynamic information routing mechanism to route information on top of collective dynamical

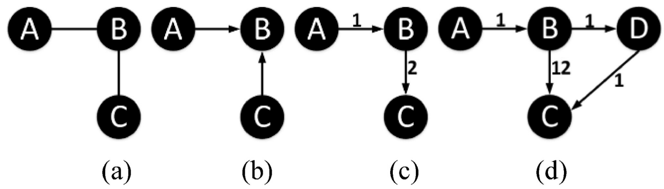

Fig. 1. Illustrations of graph samples.

reference states to achieve flexible information routing in complex networks [7]. Ling et al. proposed a global dynamic routing strategy for network systems based on the information of the queue length of nodes to improve the traffic capacity [21]. Lin et al. proposed three advanced algorithms to improve the efficiency of local routing strategies [19]. Kawamoto et al. proposed a new efficient heuristic algorithm to balance the network traffic by achieving the minimization of the maximum betweenness [15]. Yan et al. considered the possible congestion in the nodes along actual paths that can be resolved by redistributing traffic load in central nodes to other non-central nodes [39]. Holterbach et al. presented a fast-reroute framework enabling routers to restore connectivity in a few seconds [32]. However, all above works focus on designing the routing policies based on the heuristic experience or a certain consideration. It is lack of a general analysis from the practical observation of Internet datasets.

However, routing strategies and technologies have been well investigated in SDN [1, 11]. Zeng et al. first evaluated the performance of a mainstream software-defined routing platform named RouteFlow [43]. Destounis et al. proposed two routing control policies for SDN controllers to minimize the time-average routing cost while respecting a network reconfiguration budget [9]. Lin et al. proposed a simulated annealing-based QoS-aware routing (SAQR) algorithm that can adaptively adjust weights of delay, loss rate, and bandwidth in a cost function to find the best fit path [20]. Kharkongor et al. proposed an efficient routing protocol that considers the energy consumption of heterogeneous devices in software-defined IoT [5]. Lee et al. proposed a segment routing algorithm for SDN that considers the balance of traffic load and reduces the extra cost of packet header size [18]. However, all above works do not consider the data-driven routing optimization leveraging on the power of practical Internet data.

As the amount of network data increases, big data analysis is expected to efficiently guide the routing decision [6, 36]. To this end, this work is driven by the practical Internet datasets and finds a hidden indicator for routing efficiency through a series of experimental analysis. A data-driven software defined routing scheme is also proposed to improve the network performance and reduce the routing management cost.

3 MODELING AND ANALYSIS

To analyze the practical Internet topology data, we first should comprehend some basic models. The central points of the complex network have often been identified using graph-based theoretical centrality measures.

We model the Internet as a graph $G = (V, E)$ where $V = \{v_1, v_2, v_3 \ldots\}$ is the set of vertices and $E = \{e_1, e_2, e_3 \ldots\}$ is the set of edges (unordered pairs of vertices). Let e_{uv} denote the edge that connects node u and node v ($u, v \in V$). If for each edge e_{uv}, $(u, v) = (v, u)$, then the graph is called an undirected graph. If $(u, v) \neq (v, u)$, then the graph is a directed graph. A path is a sequence of vertices, such that there is an edge in E between all consecutive pairs of vertices. A geodesic between u and v is a path containing a fewest possible number of vertices; the number of edges of a geodesic between u and v is called a distance $d(u, v)$. In Figure 1, (a) is an undirected graph and (b) is a directed graph. In Figure 1(a), the distance between B and C is one hop, therefore $d(B,C) = d(C,B) = 1$. As Figure 1(b) is a directed graph, thus $d(C,B) = 1$, but $d(B,C) = +\infty$ (that is, the path from B to C is unreachable).

A node degree is the most fundamental measure of a node in a network. Here, the degree of a node i is simply defined to be the number k_i of its existing edges. Thus, as shown in Figure 1(a), $k_A = 1, k_B = 2, k_C = 1$. In Figure 1(b), $k_{A_{out}} = 1$, $k_{B_{in}} = 2$, $k_{C_{out}} = 1$. Furthermore, let $\tilde{k}$ denote the average node degree; it is the average value of all node degrees k_i. It is clear that

$$\tilde{k} = \frac{1}{N}\sum_{i \in V} k_{i_{(in,out)}}, \tag{1}$$

where N is the total number of nodes in the node set V.

Thus, in Figure 1(a) and (b), the average degree is 4/3.

Weighted graphs (namely, edges) have weight values in a graph and are very common in complex networks. For each edge e_{ij}, we define its weight as w_{ij}. For example, in Figure 1(c), $w_{AB} = 1, w_{BC} = 2$. In practice, the weights are correlated with node degrees, edges, and other metrics, respectively. In this article, w_{ij} is defined as follows:

$$w_{ij} = {k_i}^{\alpha} \times {k_j}^{\beta}, \tag{2}$$

where α and β are the variable parameters to indicate the correlation coefficient between the degree of nodes i, j and the weight of edge e_{ij}. For example, if α is bigger than β, the weight of edge e_{ij} will be easier affected by the degree of node i than node j.

Assuming a path from the node i to node j is $P_{ij} = \{v_1, v_2, \ldots, v_n\}$ and v_x is a node included in the path. $L(P_{ij})^{(\alpha,\beta)}$ represents the sum of the weights of a path from node i to node j with variable parameters α and β. The equation is as follows:

$$L(P_{ij})^{(\alpha,\beta)} = \sum_{m=1}^{n-1} w_{m,m+1} = \sum_{m=1}^{n-1} \left(k_m^{\alpha} \times k_{m+1}^{\beta}\right). \tag{3}$$

For any two nodes, we can obtain many P_{ij}. We define the node set that makes $L(P_{ij})^{(\alpha,\beta)}$ minimum as the shortest path. On the Internet, we always want to send packets at minimum cost, i.e., making $L(P_{ij})^{(\alpha,\beta)}$ as small as we possibly can. As shown in Figure 1(d), the shortest path from A to C is the path $\{A, B, D, C\}$.

Next, we present the theoretical analysis of the relationship between network throughput and betweenness centrality.

Differential Ratio of Betweenness Centrality: Betweenness centrality is used in complex networks to estimate the importance of a node. The betweenness of node i is as follows:

$$B(i) = \sum_{o \neq q} \frac{\sigma_{oq}(i)}{\sigma_{oq}}, \tag{4}$$

where σ_{oq} is the number of all shortest paths from a node o to node q and $\sigma_{oq}(i)$ is the number of shortest paths from a node o to node q going through i.

Specifically, in a scale-free network, different nodes have different degrees. We divide nodes into several groups according to their degree. All nodes with the same degree will belong to the same group. Let B^k denote the average betweenness of nodes with the same degree. Then, the average betweenness of nodes with a degree k is defined as follows:

$$\tilde{B}^k = \frac{1}{N_k}\sum_{i \in V_k} B(i), \tag{5}$$

where N_k is the number of the nodes in the set V_k, $B(i)$ is the betweenness of the node i.

Based on this, we can present the parameter of the average differential ratio of betweenness centrality, denoted by ∇_B. It shows the increasing rate of betweenness.

The equation for ∇_B is:

$$\nabla_B = \frac{1}{N^2} \sum_{k \in V_k, k' \in V_{k'}} \frac{\tilde{B}^{k'} - \tilde{B}^k}{k' - k}, \tag{6}$$

where v_k is a node with degree of k, v'_k is a node with degree of k', N is the total number of nodes in the node set V, and ∇_B is the average differential ratio of betweenness centrality.

Network throughput: For the node set V_k in which each node has a degree k, the network throughput is denoted by the average amount of packets with the following equation:

$$\tilde{W}^k(t) = \frac{1}{N_k} \sum_{i \in V_k} W_i(t), \tag{7}$$

where N_k is the number of the nodes in the set V_k, $W_i(t)$ is the number of current packets in the node i, and $\sum_{i \in V_k} W_i(t)$ is the number of total packets in the set V_k.

Let λ^k denote the average packet generation rate of nodes whose degree is k. It is a measure of the packet generation rate by the host nodes and is defined as follows:

$$\lambda^k = \frac{1}{N_k} \sum_{i \in V_k} \lambda_i, \tag{8}$$

where λ_i is the packet generation rate of the node i.

When the network is about to reach a congestion, the increasing rate of packets in the network is zero, and the packet generation rate is the same as that of network throughput. We have

$$\frac{d}{dt} W(t) = \rho \lambda N_k - \frac{W(t)}{\tilde{\tau}^*(t)} = 0, \tag{9}$$

where ρ is the proportion of host node.

On this condition, we can use λ^k to denote the network throughput. It is easy to get:

$$\rho \sum_{k \in K} \lambda^k N_k = \frac{\sum_{k \in K} \tilde{W}^{k*}(t) N_k}{\tilde{\tau}^*(t)}, \tag{10}$$

where K is the set of all node degrees and $\tilde{W}^{k*}(t)$ is the average amount of network throughput of nodes in the set N_k when the network arrives at a critical congestion state.

Besides, average transmission time depends on average transmission distance and forwarding time. It is defined by:

$$\tilde{\tau}^*(t) \approx \tilde{D} + \tilde{D} \sum_{k \in K} N_k \tilde{B}^k \frac{\tilde{W}^{k*}(t)}{R^k}, \tag{11}$$

where R^k is the forwarding rate of the nodes with a degree of k, D is the transmission distance, and $\tilde{D}$ is the average value of D.

When the network arrives at a critical congestion state, the queue is at its max length:

$$\tilde{W}^{k*} \approx L^k, \tag{12}$$

where L^k is the max queue length of nodes in V_k.

According to Equations (9), (10), (11), and (12), it is easy to get:

$$\sum_{k \in K} \lambda^k N_k = \frac{\sum_{k \in K} L^k N_k}{\rho \tilde{D} \left(1 + \sum_{k \in K} N_k \tilde{B}^k \frac{L^k}{R^k}\right)}. \tag{13}$$

Table 1. Internet Topological Data Statistics (2007–2015)

Data	N	E	K_max	K_ave	d	C	r
2007	26,955	74,499	2,753	5.528	3.79	0.261	−0.198
2008	30,018	82,630	2,632	5.505	3.786	0.267	−0.216
2009	33,017	94,073	2,591	5.698	3.812	0.257	−0.225
2010	36,111	10,2310	2,939	5.666	3.836	0.252	−0.215
2011	39,703	122,580	3,330	6.175	3.818	0.265	−0.218
2012	42,847	138,306	3,703	6.456	3.825	0.258	−0.213
2013	45,427	159,049	4,137	7.002	3.805	0.266	−0.205
2014	46,085	175,538	4,306	7.618	3.763	0.279	−0.223
2015	52,351	204,401	4,765	7.809	3.799	0.272	−0.233

The network throughput of nodes in V_k is defined by λ^k:

$$\lambda^k N_k \propto \frac{L^k N_k}{\rho \tilde{D}\left(1 + \sum_{k \in K} N_k \tilde{B}^k \frac{L^k}{R^k}\right)}. \tag{14}$$

The network throughput of nodes λ_c depends on the minimum of λ^k, approximately:

$$\lambda_c \propto \min_{k \in K} \frac{R^k L^k}{\rho \tilde{D}(R^k + N_k \tilde{B}^k L^k)}. \tag{15}$$

Network throughput is an important evaluation parameter for the routing strategy. As is well known, given a network topology, the better the routing strategy is, the greater the network throughput is.

Now, we can calculate the λ_c for each network graph. Based on the above theoretical model in Equation (15), we can formulate the network throughput dependent with betweenness centrality. We believe that a significant task is to optimize the network throughput based on network metrics including betweenness centrality and differential ratio of betweenness centrality. In the following sections, we will focus on exploring some useful laws of improving network throughput based on the observations of practical Internet datasets.

4 NETWORK DATASET DESCRIPTIONS

The Internet has many special features, such as super-large scale, isomerous, distributed deployment, so there is underlying difficulty in obtaining accurate and complete Internet topologies. Alternatively, partial network datasets can be collected and built by leveraging some efficient active probing techniques [41]. These datasets will stimulate the researchers to make novel findings, assumptions, and analyses for the practical Internet.

In this article, we adopt the network topology data from CAIDA project [4]. It is the most famous project for measuring network topology. The CAIDA network topology data are collected by 30 measure points distributed extensively. The topology is approved in the scientific community.

We selected the real-world topology data from 2007 to 2015 as our raw materials. Then, we use the *Networkx* tool in Python to analyze these data to build the experimental dataset [25]. To further observe the varying pattern of Internet topology data during these years, we make a study of some key parameters of Internet topological data. Table 1 shows the summary information of dataset collections. Figure 2 shows the varying number of edges, nodes, and maximum degrees in Internet topological data during the years 2007 to 2015. Figure 3 shows the varying number of average degree and average shortest path. In this table, N is the total number of nodes, E is the number

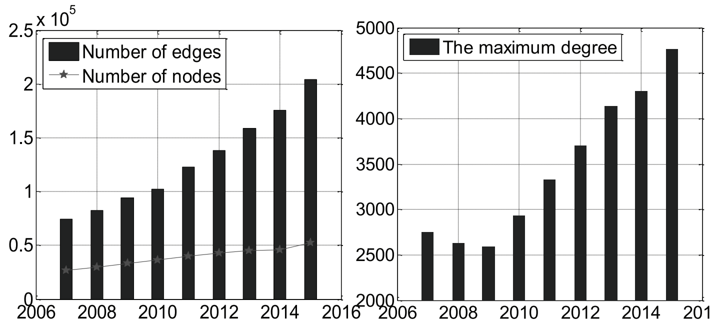

Fig. 2. Internet topological data: (a) number of edges and nodes and (b) maximum degree.

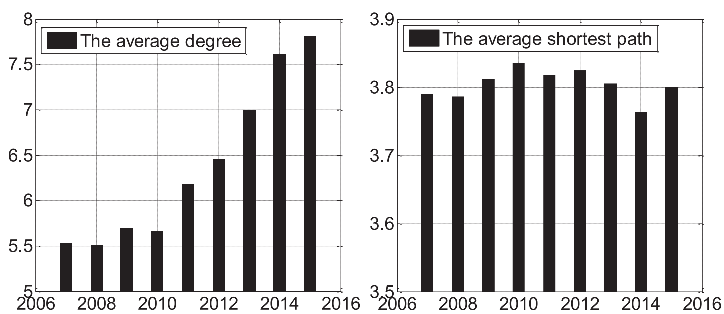

Fig. 3. Internet topological data: (a) average degree and (b) average shortest path.

of edges, K_{max} is the max degree of nodes, K_{ave} is the average degree of nodes, d is the average shortest path length, C is the average clustering metric, and r is the average degree correlation. It is not difficult to find the features of the Internet:

- The size of the Internet grows continuously. We can see a clear growth of the number of network nodes and edges at the AS level, shown in Figure 2(a).
- The maximum node degree is relatively stable during 2007–2009. However, it increases linearly from 2010–2015, as shown in Figure 2(b).
- Similarly, the average node degree in the AS level has little change during 2007–2010. However, it increases greatly from 5.6 to 7.8 during 2010–2015, shown in Figure 3(a).
- The average shortest path length presents a steady trend. In Figure 3(b), it maintains in a relatively small range of (3.76, 3.84).

All above observations indicate that the Internet has a small world effect. Although the scale of the Internet continues to expand, the changing amplitude of the average shortest path length is not obvious. Except for the above observations, there are many hidden inherent rules to be explored. As introduced in Section 3, the betweenness centrality is used to indicate the importance of a node in a network. Therefore, one straightforward question is whether there is any potential relationship between the betweenness centrality and the routing efficiency. Furthermore, how does the changing rate of betweenness centrality, i.e., DRBC, affect the network throughput? The answer should be from the in-depth observation of practical Internet Datasets. In the following section, we will make a further in-depth investigation on these Internet datasets.

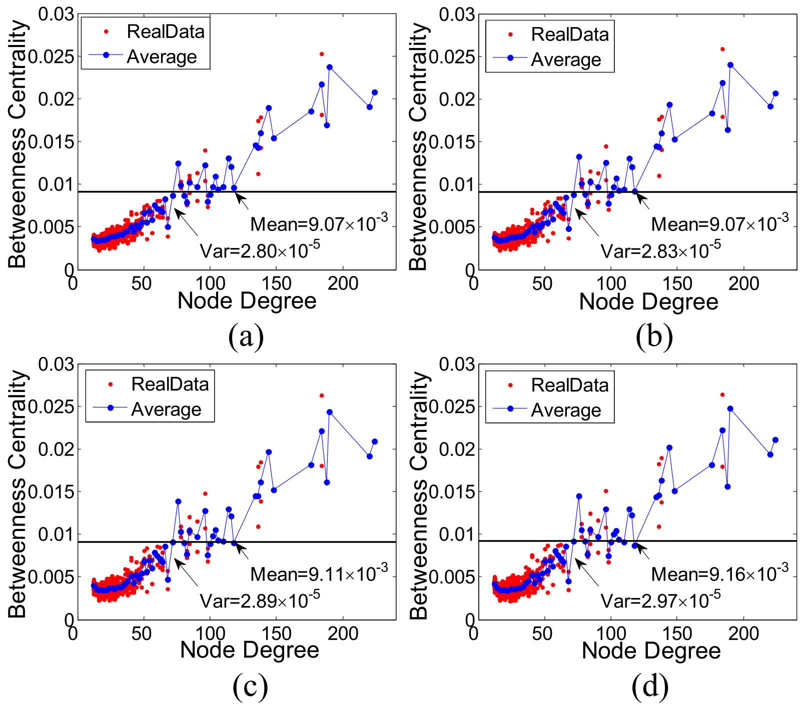

Fig. 4. Betweenness Centrality: (a) $\alpha = 0.1$, $\beta = 0.6$. (b) $\alpha = 0.1$, $\beta = 0.7$. (c) $\alpha = 0.1$, $\beta = 0.8$. (d) $\alpha = 0.1$, $\beta = 0.9$.

5 DATA PROCESS AND ANALYSIS

This section presents our experimental analysis. To simplify the experiment process, we select a set of samples in the datasets. We choose the data samples following a principle [2]: The distribution of degrees of the selected networks approximately satisfies:

$$p(k) = k^{-3}, \tag{16}$$

where $p(k)$ is appearance probability of nodes with a degree of k.

To answer the questions proposed in Section 4, we conduct two separate experimental observations to reveal the relationship of betweenness centrality, node degree, DRBC metric, and network throughput.

Experimental observation 1: *Relationship of betweenness centrality and node degree*

According to Equation (2), the weight of an edge is correlated with the degrees of nodes i and j. The variable parameters, α and β, directly affect the shortest path routing policy. To this end, we first select different values for α and β. In this experiment, we perform four sets of tests, in each of which α is same but β is different. With the knowledge of a pair of α and β, we add weight to the directed graph according to Equation (2). Then, we use the shortest path algorithm to calculate the betweenness centrality of nodes. Finally, we obtain a set of data for betweenness centrality in different cases and analyze the relationship of the betweenness centrality and node degree.

Figures 4–7 show the relationship between degree and betweenness of the four cases. As we can see in the figures, there are more nodes of which the degree is below about 50, satisfying the distribution proposed at the beginning of this part. The two numbers in each figure are the mean and variation of the average data. These figures show the results have almost similar mean and

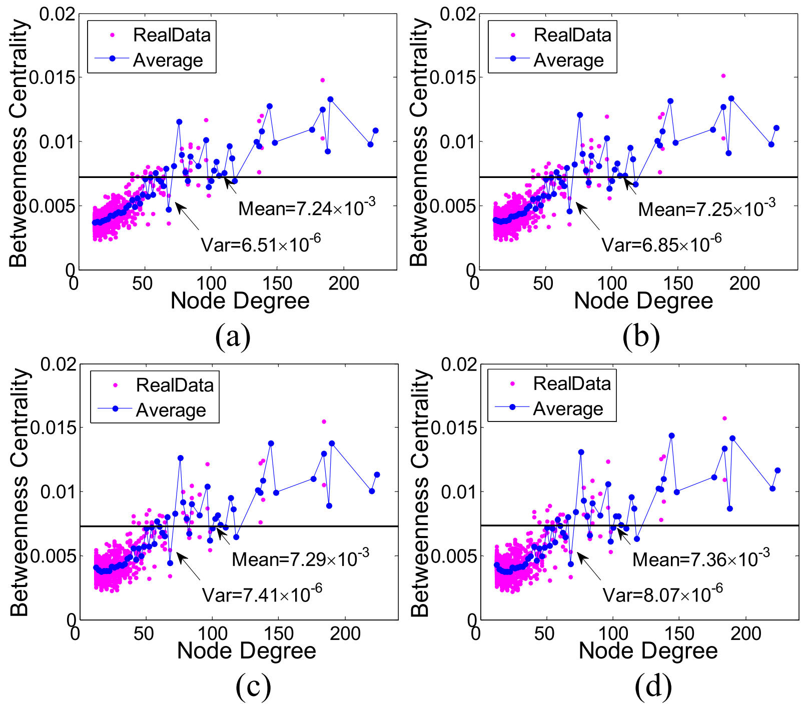

Fig. 5. Betweenness Centrality: (a) $\alpha = 0.2$, $\beta = 0.6$. (b) $\alpha = 0.2$, $\beta = 0.7$. (c) $\alpha = 0.2$, $\beta = 0.8$. (d) $\alpha = 0.2$, $\beta = 0.9$.

variation in the same set, but have quite a different mean and variation compared with the figures in other sets. Tables 2–5 present the detailed experimental results, including the max betweenness. ∇_B is the average differential ratio of Betweenness, and λ_c is the Critical packet generation rate, which represents the network throughput [19]. From the results in these four tables, we can see that the max betweenness in the same table has jitter in a very small range.

Experimental observation 2: *Relationship between DRBC metric and network throughput*

In this experiment, we mainly observe the relationship between the DRBC metric and the network throughput. We need to calculate the DRBC metric. Equation (6) defines the DRBC parameter. In practice, we need to select a topology dataset and give a routing strategy. In our experiment, we first add weight on the directed graph and use the normal shortest path (Dijkstra) algorithm to calculate the betweenness centrality of nodes. Then, we can further get the knowledge of the average differential ratio of betweenness centrality according to observing the relationship of betweenness centrality and node degree.

The algorithm for the average differential ratio of betweenness centrality is summarized based on Equation (6). Similar to Experiment 1, this experiment also has four observations where both the max betweenness centrality and the parameter α vary. It is noteworthy that in each case, the max betweenness centrality should be a constant or only has a very small change; it is because a change of the max betweenness centrality may bring a drastic and unpredicted jitter to the network structure. Then, we can get the average differential ratio of betweenness and network throughput and plot the corresponding figures of the relationship between them. Figure 8 shows the relationship between ∇_B and λ_c. We can see the greater ∇_B is, the greater the network throughput will be

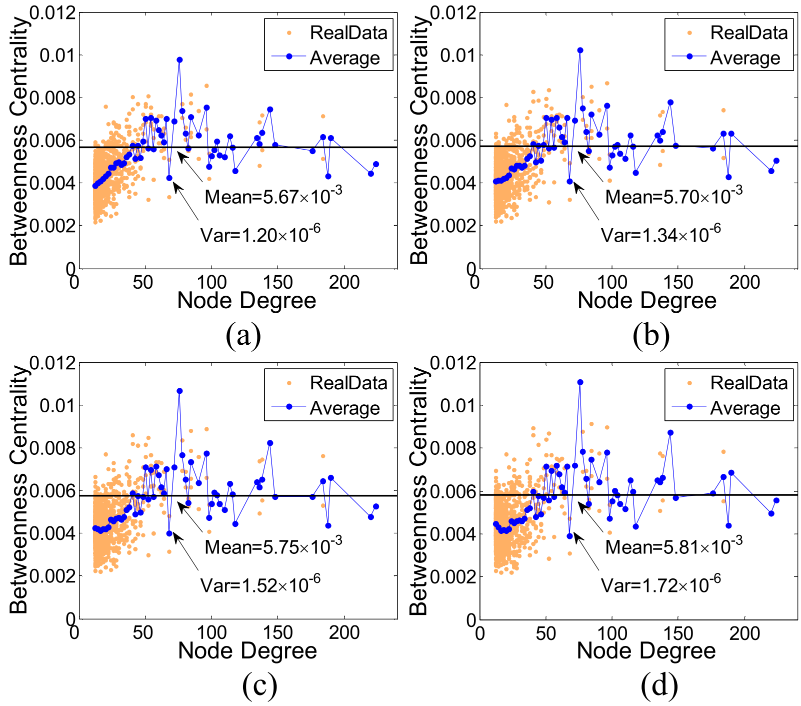

Fig. 6. Betweenness Centrality: (a) $\alpha = 0.3$, $\beta = 0.6$. (b) $\alpha = 0.3$, $\beta = 0.7$. (c) $\alpha = 0.3$, $\beta = 0.8$. (d) $\alpha = 0.3$, $\beta = 0.9$.

in four cases of $\alpha = 0.1$, 0.2, 0.3 and 0.4. The values of max betweenness are kept as 0.024, 0.013, 0.01, and 0.008 for four cases with $\alpha = 0.1$, 0.2, 0.3, and 0.4.

Based on this observation, we reveal a hidden but interesting indicator that the DRBC metric is positively proportional to network throughput. However, it requires a precondition: The maximum betweenness centrality of a network graph keeps as a constant (or with a very small change). This law can indicate routing efficiency intuitively and guide to make routing adjustment. In other words, when the maximum betweenness centrality keeps as a constant, we can use the DRBC metric to indicate the routing efficiency. For example, we can keep the maximum betweenness centrality unchanged and improve the routing efficiency and total network throughput by adjusting the DRBC metric. In the following section, we will show how to use this finding to guide routing adjustment to improve the routing efficiency.

6 APPLICATION CASE: DRBC-BASED SOFTWARE DEFINED ROUTING

In this section, we introduce an application case to show the DRBC-based software defined routing. In this application, a centralized software defined controller is employed to manage and update the routing policies [10]. Since the controller has the global network information, such as network topology and network status, it can calculate the routing policies quickly and efficiently, then distribute these updated policies to connected routers or switches. Figure 9 shows the topology of the practical test network, which includes 15 nodes numbered by 0–14. Random links are built among these nodes, as shown in the figure. There is no link between some pair of nodes. Based on the above analysis, to improve the network throughput, the network routing selection should increase the DRBC metric as much as possible; that is, two conditions need to be met: (1) the DRBC

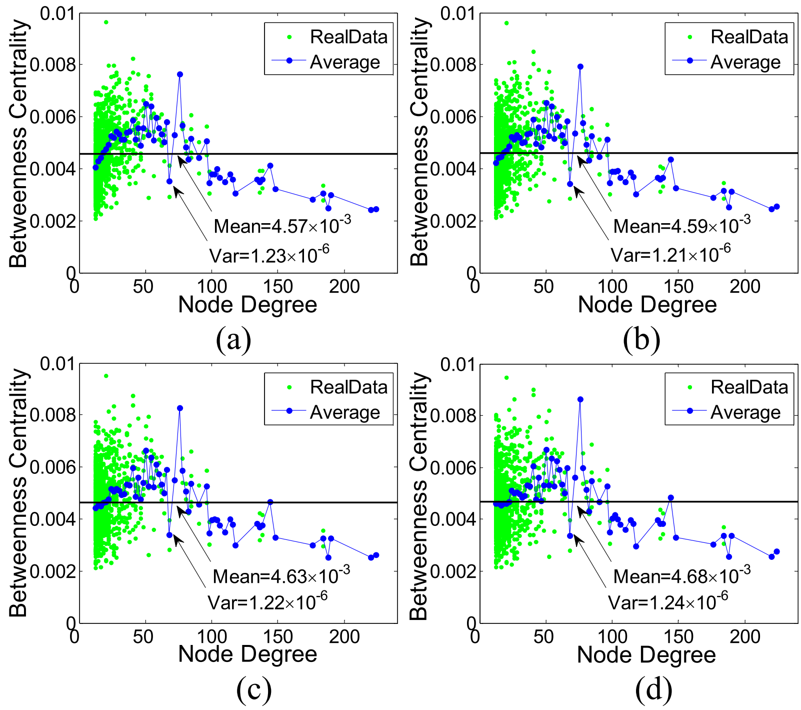

Fig. 7. Betweenness Centrality: (a) $\alpha = 0.4$, $\beta = 0.6$. (b) $\alpha = 0.4$, $\beta = 0.7$. (c) $\alpha = 0.4$, $\beta = 0.8$. (d) $\alpha = 0.4$, $\beta = 0.9$.

Table 2. The Detailed Result Data ($\alpha = 0.1$)

α	β	Max betweenness	∇_B	λ_c
0.1	0.4	0.024	0.73	139
0.1	0.5	0.024	0.66	135
0.1	0.6	0.024	0.57	130
0.1	0.7	0.024	0.49	125
0.1	0.8	0.024	0.46	121
0.1	0.9	0.025	0.45	115

value should be nonnegative; (2) the DRBC value should keep stable. Therefore, the DRBC-based routing adjustment process is summarized as follows:

- First, we should set initial values for α and β; for example, $\alpha = 0.3$ and $\beta = 0.6$. Then, we calculate the average betweenness centrality for different node degree. Figure 10 shows the relationship between average betweenness centrality and node degree. As the baseline, we use the optimal shortest path based routing algorithm [30]. In this figure, we can see the curve is not linear, which means a great improvement can be achieved through the DRBC-based software defined routing. When the curve is approximately linear, the average gradient becomes the biggest when the maximum of betweenness centrality is a certain value. In this case, the average DRBC will become the optimal.
- In step 2, we try to increase the average betweenness of nodes whose degree is 3. We look for the nodes with degree 3 in this test network, which are nodes 1, 5, and 7. Then, we

Table 3. The Detailed Result Data ($\alpha = 0.2$)

α	β	Max betweenness	∇_B	λ_c
0.2	0.4	0.013	0.27	144
0.2	0.5	0.013	0.18	131
0.2	0.6	0.013	0.13	120
0.2	0.7	0.013	0.06	111
0.2	0.8	0.014	0.03	103
0.2	0.9	0.014	0.01	98

Table 4. The Detailed Result Data ($\alpha = 0.3$)

α	β	Max betweenness	∇_B	λ_c
0.3	0.4	0.009	−0.06	111
0.3	0.5	0.009	−0.1	101
0.3	0.6	0.01	−0.15	94
0.3	0.7	0.01	−0.19	86
0.3	0.8	0.01	−0.24	81
0.3	0.9	0.011	−0.28	76

Table 5. The Detailed Result Data ($\alpha = 0.4$)

α	β	Max betweenness	∇_B	λ_c
0.4	0.4	0.007	−0.19	89
0.4	0.5	0.007	−0.24	82
0.4	0.6	0.008	−0.27	76
0.4	0.7	0.008	−0.3	70
0.4	0.8	0.008	−0.37	63
0.4	0.9	0.009	−0.37	63

decrease the weight of edges around nodes 1, 5, and 7. As a result, the average betweenness of nodes with degree 3 does not increase, but the average betweenness of nodes with degree 2 decreases. After this routing adjust, the curve has tended to be more linear. Figure 11 shows the results after the step 2 adjustment.

- In step 3, we try to decrease the average betweenness of nodes with degree 5 to increase the DRBC metric. We look for the nodes with degree 5 in this test network, which are nodes 2, 6, 9, 11, 12, and 13. Then, we increase the weight of edges around these nodes. Figure 12 shows the results after the step 3 adjustment. In this figure, we can see the curve becomes more linear than that before the second routing adjust.
- In step 4, we try to decrease the average betweenness of nodes whose degree is 4. We look for the nodes with degree 4 in this test network, which are nodes 0, 4, 8, and 10. Then, we increase the weight of edges around these nodes. Figure 13 shows the results after the step 4 adjustment. We can see that the curve has approached exactly linear.

After these four adjust steps, this DRBC-based software defined routing tends to be stable. We finally obtain an approximately optimized value for the DRBC metric, as shown in Figure 13, where the DRBC value is nonnegative and also tends to be stable in four steps.

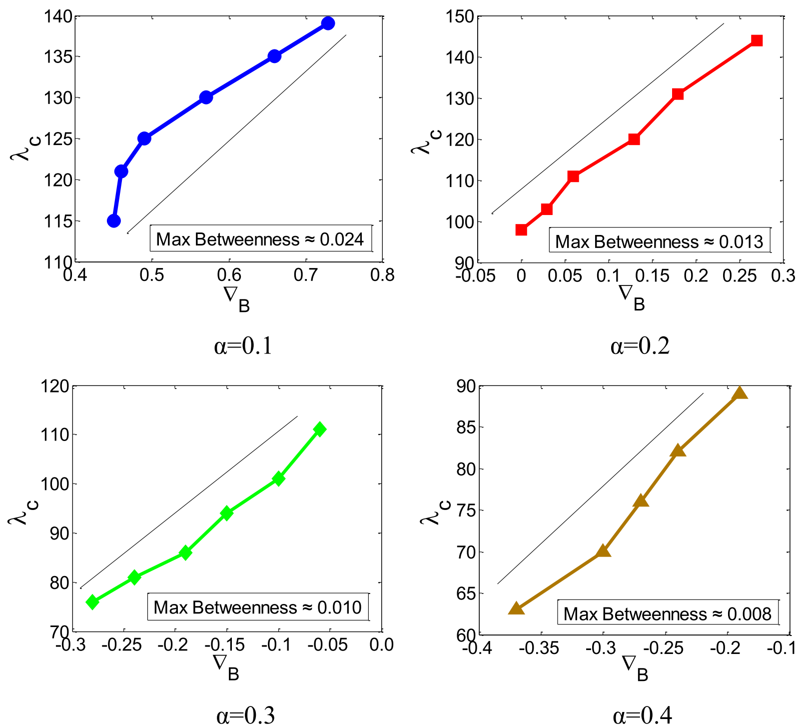

Fig. 8. Relationship between ∇_B and λ_c.

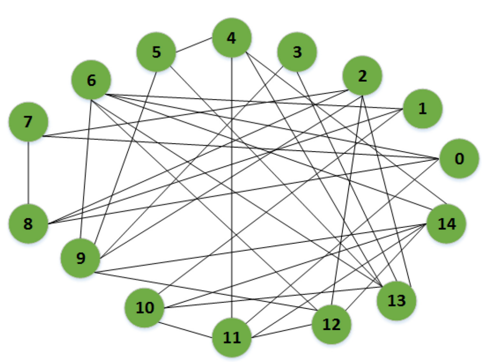

Fig. 9. The smaller experiment network topology with 15 nodes.

We compare the network throughput λ_C during the routing adjustment based on the Equation (15). Figure 14 shows the varying of network throughput λ_C in the four adjust steps. We can see the network throughput increases with each routing adjustment. The network throughput increases step-by-step (from 68 to more than 110) during the DRBC-based software defined routing, which verifies DRBC can well indicate the routing efficiency and guide the routing optimization in the next generation Internet system.

It is noted that this experiment shows how to improve the routing efficiency according to the DRBC-based routing adjustment. Therefore, we focus on comparing the performance without and

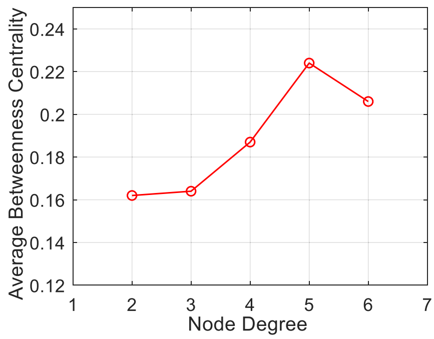

Fig. 10. Betweenness centrality during the routing adjust (baseline).

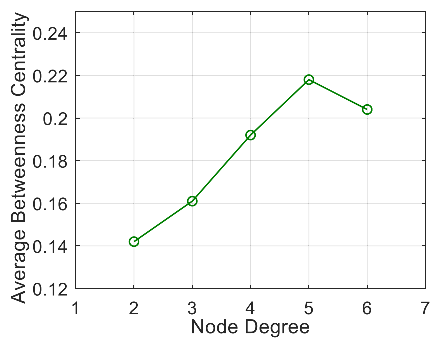

Fig. 11. Betweenness centrality during the routing adjust (step 2).

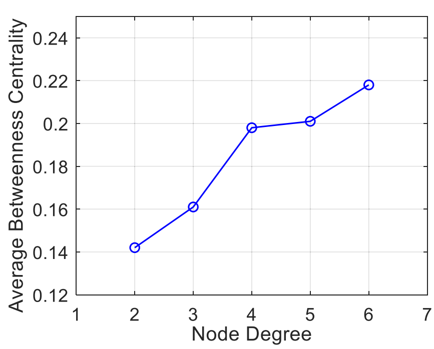

Fig. 12. Betweenness centrality during the routing adjust (step 3).

with DRBC-based routing enhancement. Based on this, we use the shortest-path-based routing algorithm [30] as the baseline.

To further verify the efficiency of the DRBC-based software defined routing, we test its performance in two larger size of the networks. The two large network topologies include 25 nodes and 35 nodes, respectively, which are shown in Figure 15. We also compare the network throughput in the different routing adjustment steps. The results are shown in Figure 16. We can see the network throughput increases step-by-step in both cases during the DRBC-based software defined routing. It shows that DRBC can guide the routing optimization in larger-size networks. Although we test the performance in only three different network sizes, we believe the DRBC-based soft-

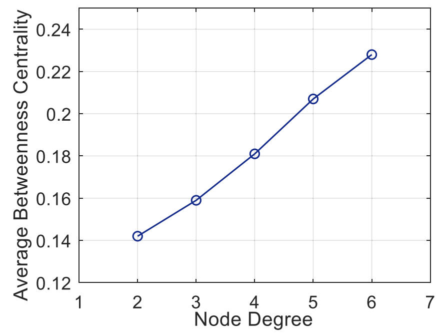

Fig. 13. Betweenness centrality during the routing adjust (step 4).

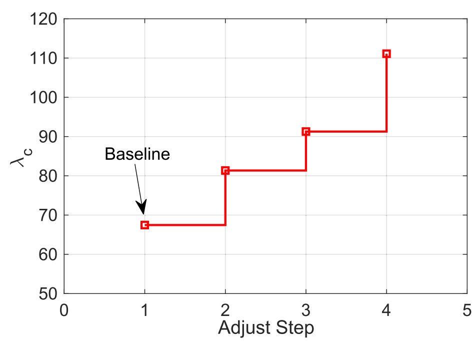

Fig. 14. Network throughput during the four adjust steps (with 15 nodes).

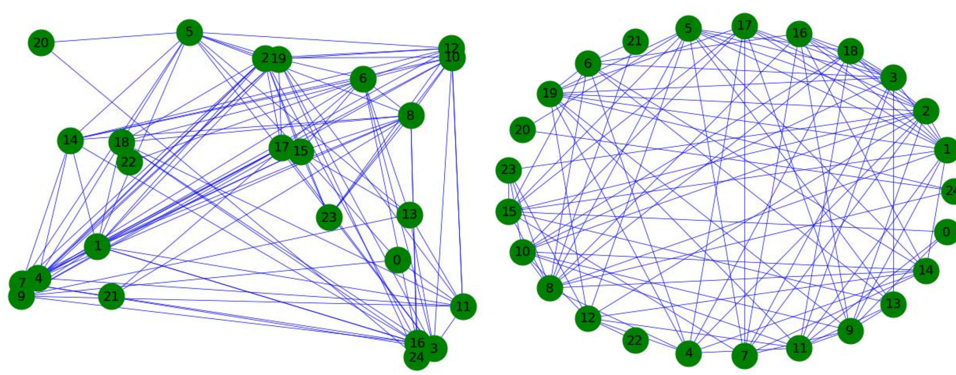

Fig. 15. The larger experiment network topology with 25 nodes and 35 nodes.

ware defined routing can also work efficiently in other size of networks. Besides, it is noteworthy that we only show a routing adjustment scheme to optimize the DRBC metric to improve the network throughput in this application case. The proposed method can be easily extended and applied to more application cases, such as using different routing adjustment and improving network reliability.

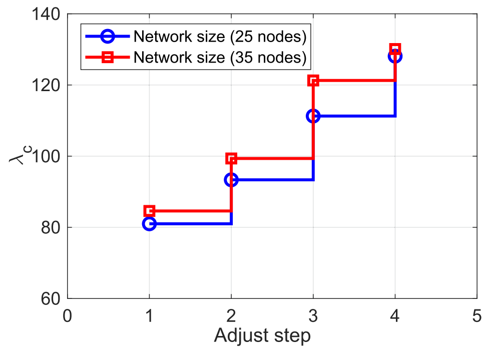

Fig. 16. Network throughput during the four adjust steps (with 25 and 35 nodes).

7 CONCLUSION

This article is motivated by exploring the hidden laws for the software defined routing based on the practical Internet data. An in-depth observation is made to investigate the relationship among betweenness centrality, node degree, and network throughput. In particular, a new metric, differential ratio of betweenness centrality, is proposed to indicate the routing efficiency. It can denote the varying amplitude of betweenness centrality to node degree. Experimental results show that the metric is proportional to the routing efficiency if the maximum of betweenness centrality keeps constant or changes in a small range. This interesting law can be used to improve the routing efficiency in the next generation Internet and reduce the overhead of routing management. In future work, we will further analyze the advanced learning methods [38] to improve the SDN routing management efficiency.

REFERENCES

[1] Z. N. Abdullah, I. Ahmad, and I. Hussain. 2019. Segment routing in software defined networks: A survey. *IEEE Commun. Surv. Tutor.* 21, 1 (2019), 464–486. DOI: https://doi.org/10.1109/COMST.2018.2869754

[2] A. L. Barab and R. Albert. 2002. Statistical mechanics of complex networks. *Rev. Mod. Phys.* 7 (2002), 47–97.

[3] M. Boguna, F. Papadopoulos, and D. Krioukov. 2010. Sustaining the internet with hyperbolic mapping. *Nat. Commun.* 9 (2010) Issue 07.

[4] CAIDA. (2018). Retrieved from: http://www.caida.org/home.

[5] B. E. Carpenter. 2016. A SDN controller with energy efficient routing in the internet of things (IoT). *Procedia Comput. Sci.* 89 (2016), 218–227.

[6] N. Cheng, F. Lyu, J. Chen, W. Xu, H. Zhou, S. Zhang, and X. Shen. 2018. Big data driven vehicular networks. *IEEE Netw.* (2018), 1–8. DOI: https://doi.org/10.1109/MNET.2018.1700460

[7] K. Christoph, T. Marc, and B. Demian. 2016. Dynamic information routing in complex networks. *Nat. Commun.* 7, 12 (2016). DOI: 10.1038/ncomms11061

[8] CIDR-Report. (2019). Retrieved from: http://www.cidr-report.org/as2.0/.

[9] A. Destounis, S. Paris, L. Maggi, G. S. Paschos, and J. Leguay. 2018. Minimum cost SDN routing with reconfiguration frequency constraints. *IEEE/ACM Trans. Netw.* 26, 4 (Aug. 2018), 1577–1590. DOI: https://doi.org/10.1109/TNET.2018.2845463

[10] X. Gao, B. Wang, and W. Deng. 2017. Software defined routing system. In *Wireless Algorithms, Systems, and Applications.* Springer International Publishing, 617–628.

[11] J. W. Guck, A. V. Bemten, M. Reisslein, and W. Kellerer. 2018. Unicast QoS routing algorithms for SDN: A comprehensive survey and performance evaluation. *IEEE Commun. Surv. Tutor.* 20 (2018), 388–415. Issue 1. DOI: https://doi.org/10.1109/COMST.2017.2749760

[12] S. Haeri and L. Trajkovic. 2014. Deflection routing in complex networks. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'14).* 2217–2220.

[13] Z. He, J. Cao, and X. Liu. 2016. SDVN: Enabling rapid network innovation for heterogeneous vehicular communication. *IEEE Netw.* 30, 4 (July 2016), 10–15. DOI: https://doi.org/10.1109/MNET.2016.7513858

[14] T. Hegr and L. Bohac. 2014. Impact of nodal centrality measures to robustness in software-defined networking. *Adv. Electr. Electron. Eng.* 12, 4 (2014), 252–259.

[15] H. Kawamoto and A. Igarashi. 2012. Efficient packet routing strategy in complex networks. *Phys. A: Stat. Mech. Appl.* 391, 3 (2012), 895–904.

[16] D. Kim, Y. H. Kim, K. H. Kim, et al. 2017. Cloud-centric and logically isolated virtual network environment based on software-defined wide area network. *Sustainability* 9, 12 (2017), 2382.

[17] D. Krioukov. 2016. Clustering implies geometry in networks. *Phys. Rev. Lett.* 116 (May 2016), 208302. Issue 20. DOI: https://doi.org/10.1103/PhysRevLett.116.208302

[18] M. Lee and J. Sheu. 2016. An efficient routing algorithm based on segment routing in software-defined networking. *Comput. Netw.* 103 (2016), 44–55.

[19] B. Lin, B. Chen, Y. Gao, et al. 2016. Advanced algorithms for local routing strategy on complex networks. *PLOS One* 11 (July 2016), 1–17.

[20] C. Lin, K. Wang, and G. Deng. 2017. A QoS-aware routing in SDN hybrid networks. *Procedia Comput. Sci.* 110 (2017), 242–249.

[21] X. Ling, M. Hu, R. Jiang, and Q. Wu. 2010. Global dynamic routing for scale-free networks. *Phys. Rev. E* 81 (Jan. 2010), 016113. Issue 1.

[22] W. Liu and B. Liu. 2014. Congestion control in complex network based on local routing strategy. *Acta Phys. Sin.* 63, 24 (2014), 248901.

[23] M. Luckie and R. Beverly. 2017. The impact of router outages on the as-level internet. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'17)*. ACM, New York, NY, 488–501.

[24] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, et al. 2008. OpenFlow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.* 38, 2 (Mar. 2008), 69–74.

[25] Networkx. (2018). Retrieved from: http://networkx.github.io.

[26] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti. 2014. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Commun. Surv. Tutor.* 16, 3 (July 2014), 1617–1634. DOI: https://doi.org/10.1109/SURV.2014.012214.00180

[27] Z. Qazi, C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu. 2013. SIMPLE-fying middlebox policy enforcement using SDN. *SIGCOMM Comput. Commun. Rev.* 43, 4 (Aug. 2013), 27–38.

[28] W. Quan, N. Cheng, M. Qin, H. Zhang, H. Chan, and X. Shen. 2019. Adaptive transmission control for software defined vehicular networks. *IEEE Wirel. Commun. Lett.* 8, 3 (2019), 653–656. DOI: https://doi.org/10.1109/LWC.2018.2879514

[29] W. Quan, Y. Liu, H. Zhang, and S. Yu. 2017. Enhancing crowd collaborations for software defined vehicular networks. *IEEE Commun. Mag.* 55, 8 (2017), 80–86. DOI: https://doi.org/10.1109/MCOM.2017.1601162

[30] K. G. Ramakrishnan and M. A. Rodrigues. 2001. Optimal routing in shortest-path data networks. *Bell Labs Tech. J.* 6, 1 (Jan. 2001), 117–138.

[31] D. Rueda, E. Calle, and L. Marzo. 2017. Improving the robustness to targeted attacks in software defined networks. In *Proceedings of the 13th International Conference on Design of Reliable Communication Networks (DRCN'17)*. 1–8.

[32] H. Thomas, V. Stefano, D. Alberto, and V. Laurent. 2017. SWIFT: Predictive fast reroute (SIGCOMM'17). 460–473.

[33] K. Wang, H. Yin, W. Quan, and G. Min. 2018. Enabling collaborative edge computing for software defined vehicular networks. *IEEE Netw.* 32, 5 (2018), 112–117. DOI: https://doi.org/10.1109/MNET.2018.1700364

[34] N. Wang, K. H. Ho, G. Pavlou, and M. Howarth. 2008. An overview of routing optimization for internet traffic engineering. *IEEE Commun. Surv. Tutor.* 10, 1 (year 2008), 36–56. DOI: https://doi.org/10.1109/COMST.2008.4483669

[35] W. Wang, C. Yin, G. Yan, and B. Wang. 2006. Integrating local static and dynamic information for routing traffic. *Phys. Rev. E* 74 (Jul 2006), 016101. Issue 1.

[36] J. Wu, M. Dong, K. Ota, J. Li, and Z. Guan. 2018. Big data analysis-based secure cluster management for optimized control plane in software-defined networks. *IEEE Trans. Netw. Serv. Manag.* 15, 1 (Mar. 2018), 27–38. DOI: https://doi.org/10.1109/TNSM.2018.2799000

[37] J. Wu, C. K. Tse, and F. C. M. Lau. 2014. Effective routing algorithms based on node usage probability from a complex network perspective. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'14)*. 2209–2212. DOI: https://doi.org/10.1109/ISCAS.2014.6865608

[38] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu. 2019. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. *IEEE Commun. Surv. Tutor.* 21, 1 (2019), 393–430. DOI: https://doi.org/10.1109/COMST.2018.2866942

[39] G. Yan, T. Zhou, B. Hu, Z. Fu, and B. Wang. 2006. Efficient routing on complex networks. *Phys. Rev. E* 73 (Apr. 2006), 046108. Issue 4. DOI: https://doi.org/10.1103/PhysRevE.73.046108

[40] Q. Ye, W. Zhuang, S. Zhang, A. Jin, X. Shen, and X. Li. 2018. Dynamic radio resource slicing for a two-tier heterogeneous wireless network. *IEEE Trans. Vehic. Technol.* 67, 10 (Oct. 2018), 9896–9910.

[41] S. Yoon, T. Ha, S. Kim, and H. Lim. 2017. Scalable traffic sampling using centrality measure on software-defined networks. *IEEE Commun. Mag.* 55, 7 (July 2017), 43–49.

[42] W. T. Zaumen and A. J. Garcia-Luna. 1991. Dynamics of distributed shortest-path routing algorithms. *SIGCOMM Comput. Commun. Rev.* 21, 4 (Aug. 1991), 31–42. DOI: https://doi.org/10.1145/115994.115997

[43] P. Zeng, K. Nguyen, Y. Shen, and S. Yamada. 2014. On the resilience of software defined routing platform. In *Proceedings of the 16th Asia-Pacific Network Operations and Management Symposium.* 1–4. DOI: https://doi.org/10.1109/APNOMS.2014.6996605

[44] H. Zhang, W. Quan, H. Chao, and C. Qiao. 2016. Smart identifier network: A collaborative architecture for the future internet. *IEEE Netw.* 30, 3 (May 2016), 46–51. DOI: https://doi.org/10.1109/MNET.2016.7474343

[45] S. Zhang, W. Quan, J. Li, W. Shi, P. Yang, and X. S. Shen. 2018. Air-ground integrated vehicular network slicing with content pushing and caching. *IEEE J. Select. Areas Commun.* 36 (2018). Issue 10. Retrieved from: https://arxiv.org/abs/1806.03860.

[46] S. Zhu and G. M. Huang. 1998. A new parallel and distributed shortest path algorithm for hierarchically clustered data networks. *IEEE Trans. Parallel Distrib Syst.* 9, 9 (Sept. 1998), 841–855.

Received January 2019; revised June 2019; accepted August 2019