Computers, Materials & Continua
DOI:10.32604/cmc.2020.013355
Article

Tech Science Press

Efficient Routing Protection Algorithm in Large-Scale Networks

Haijun Geng[1,2,*], Han Zhang[3] and Yangyang Zhang[4]

[1]School of Software Engineering, Shanxi University, Taiyuan, 030006, China
[2]2Institute of Big Data Science and Industry, Shanxi University, Taiyuan, 030006, China
[3]School of Cyber Space and Technology, Beihang University, Beijing, 100191, China
[4]College of Engineering Northeastern University, Boston, 02115, USA
[*]Corresponding Author: Haijun Geng. Email: ghj123025449@163.com
Received: 04 August 2020; Accepted: 20 September 2020

Abstract: With an increasing urgent demand for fast recovery routing mechanisms in large-scale networks, minimizing network disruption caused by network failure has become critical. However, a large number of relevant studies have shown that network failures occur on the Internet inevitably and frequently. The current routing protocols deployed on the Internet adopt the reconvergence mechanism to cope with network failures. During the reconvergence process, the packets may be lost because of inconsistent routing information, which reduces the network's availability greatly and affects the Internet service provider's (ISP's) service quality and reputation seriously. Therefore, improving network availability has become an urgent problem. As such, the Internet Engineering Task Force suggests the use of downstream path criterion (DC) to address all single-link failure scenarios. However, existing methods for implementing DC schemes are time consuming, require a large amount of router CPU resources, and may deteriorate router capability. Thus, the computation overhead introduced by existing DC schemes is significant, especially in large-scale networks. Therefore, this study proposes an efficient intra-domain routing protection algorithm (ERPA) in large-scale networks. Theoretical analysis indicates that the time complexity of ERPA is less than that of constructing a shortest path tree. Experimental results show that ERPA can reduce the computation overhead significantly compared with the existing algorithms while offering the same network availability as DC.

Keywords: Large-scale network; shortest path tree; time complexity; network failure; real-time and mission-critical applications

1 Introduction

In recent years, the Internet has become a widely used platform for various network applications. With the rapid development of the Internet, several real-time and mission-critical applications, such as VoIP and video and online games, are deployed [1]. These applications are susceptible to network delay and interruption, which stress strict requirements on network availability [2]. Moreover, even a few seconds

of network discontinuity would have an adverse effect on these applications [3]. However, network failures are common on the Internet. The IP networks need to reconverge when network failure occurs. The convergence time for the currently deployed intra-domain routing protocol is the order of seconds when network components are invalid. During this period, several packets may be dropped because of the inconsistent routing information [4]. The Internet service provider (ISP) has strong motivations to enhance the network survivability when network failures occur [5]. Therefore, improving the Internet routing availability has become an urgent problem that needs to be solved.

To improve the availability of intra-domain routing, the routing protection scheme is usually applied in the academia and industry [6]. The routing protection aims to prove fast convergence when network failures have been detected [7]. Existing routing protection schemes can be divided into two sub-categories depending on whether or not special cooperation between routers is required for packet forwarding. Cooperation-free schemes compute multiple next-hops for each destination, and each router selects an appropriate next-hop for standard packet forwarding independently, where care must be taken such that the induced forwarding paths are loop-free. The benefit is that they can provide not only redundant backup links but also other features, such as load balancing and high throughput. The other sub-category of schemes computes for a link to protect a multihop repair path that is agreed by all routers on that path. Thus, special cooperation mechanisms have to be used to reroute packets along that path.

In this work, we focus on the first type of scheme. We also confine our work in link-state routing networks. Most ISPs prefer link-state routing instead of distance-vector routing in their intra-domain system because of its merits, such as fast convergence and good support for metrics. Layer2 networks also incorporate link-state routing into their network architecture, such as the standardized transparent interconnection of lots of links. Furthermore, during topology changes caused by link or node failure, millisecond-level fast convergence, which poses stringent performance requirement to route computation, is preferred.

Among all of the hop-by-hop routing protection schemes, downstream path criterion (DC) [8] has been favored by the industry because of its simplicity in coping with all the single-link failure scenarios. All of the existing implementation algorithms about DC are time consuming and require a large amount of router CPU resources in large-scale networks.

However, the deployment of DC in real ISP networks is difficult because of the substantial computational overhead. Therefore, an efficient DC-based algorithm is required to be easily deployed in ISP. Thus, a lightweight IPFRR scheme is desired to provide cost-efficient routing protection effectively. Therefore, this study investigates the application of incremental shortest path first algorithm to reduce the computational overhead of the DC implementation. In particular, our contributions can be summarized as follows:

- We propose an efficient intra-domain routing protection algorithm (ERPA) in large-scale networks.
- Theoretical analysis indicates that the computation complexity of ERPA is less than that of constructing a shortest path tree (SPT).
- Theoretical analysis indicates that ERPA can provide the same network availability as DC.
- In terms of computation overhead and network availability, the theoretical analysis and experimental results are consistent.

2 Related Works

Nowadays, network failures have become routine events rather than exceptions [9]. Many schemes for enhancing robustness against network failures have been proposed. Existing approaches fall in either one of the two categories: reactive and proactive approaches. The former studies the reduction of convergence time of routing protocol after the occurrence of failures, whereas the latter addresses the pre-calculating backup paths before the failures. Reactive approaches are applicable to all kinds of network failure scenarios

regardless whether they are single or multiple failures. However, reactive approaches are subject to the risk of routing flap. Therefore, proactive approaches are preferred by the academia and industry. The idea of multitopology configuration method is that each router saves multiple configurations, and each configuration can protect some links to adapt to different link failures. However, the more configurations the router keeps, the more overhead it will introduce. Path splicing [10] is a classical multitopology configuration method that calculates multiple SPTs by adjusting link weights. Each spanning tree corresponds to a routing path, and packets can be forwarded among multiple spanning trees. However, a routing loop may be observed in path splitting, thereby degrading network performance. Dispath [11], which can protect all possible single fault cases in the network, is proposed in the literature. By constructing a directed acyclic graph with three disjoint edges [12], any two links in the network can be protected from failure. This study [13] investigates and proves that single- and double-fault protection algorithms are not restricted by network topology. Among the hop-by-hop proactive schemes, DC has been favored by the industry because of its simplicity in coping with all the single-link failure scenarios. However, all of the existing DC-based implementation algorithms are time consuming and require a large amount of router CPU resources. Authors propose an efficient algorithm called TBFH [14], which provides greater path diversity than ECMP with a very low overhead. In particular, TBFH computes the two best first hop disjoint paths efficiently. We also propose a SPT-based multipath routing algorithm called DMPA [15]. DMPA guarantees the loop-freeness of the induced routing path by maintaining a partial order of the routers underpinning it implicitly. The time complexity of DMPA does not depend on the degree of the calculating router. However, the network availability of TBFH and DMPA is lower than that of the DC. Unlike the aforementioned studies, our main concerns include computational efficiency and network availability, which are critical for the algorithm. Based on the existing work on this research area, we propose an algorithm whose complexity is less than that of constructing a SPT and without degrading the network availability for the first time.

3 Network Model and Problem Description

3.1 Network Model

In this section, we will first show the network model and then describe the key problems that need to be solved in this work. For ease of reading, some of the symbols used in this paper are summarized in Tab. 1. A network can be expressed as a undirected graph $G = (V, E)$, where V and E denote the set of nodes and the set of edges in the network, respectively. For any node $v \notin V$, we use $N(v)$ to denote all the neighbors of node v; $spt(v)$ represents a SPT rooted at v; and $D(v,x)$ is the descendant of node v in $spt(v)$. Each link (i,j) in the network has a weight $w(i,j)$ and a failure probability $r(i,j)$. For any node pair x and y, we use $cost(x,y)$ to indicate the shortest cost from node x to node y; $dn(x,y)$ is the default next-hop from node x to node y; and $bn(x,y)$ is the backup next-hop set from node x to node y. We will use a simple example to explain the aforementioned concepts. For example, Fig. 1 presents a SPT rooted at node c, $N(c) = \{a,b\}$, $D(c,a) = \{a,h,d,g\}$, $D(c,b) = \{b,e,f\}$, $cost(c,d) = 7$, $cost(c,g) = 6$, $dn(c,g) = a$, and $dn(c,f) = b$.

The currently deployed intra-domain routing protocols (e.g., OSPF and IS-IS) only employ the shortest paths to forward packets. Thus, they need to reconverge when network component failures occur. The packets may be dropped because of invalid routing information. These protocols never exploit the inherent diversity of Internet topology and cannot handle network failure flexibly. Therefore, DC has been proposed to cope with all the single-link failure scenarios. The DC can be expressed as follows:

DC: For packets forwarded to a destination d, node $c(c \neq d)$ can forward them to any of its neighboring node x when $cost(x,d) < cost(c,d)$, and no forwarding loop will exist in the induced forwarding path.

Table 1: Symbols

$G = (V, E)$	Network topology
$N(v)$	Neighbor set of v
$spt(c)$	Shortest path tree rooted at c
$D(spt(c))$	Descendants of x in $spt(c)$
$w(i,j)$	Weight of link (i,j)
$cost(x,y)$	Shortest cost from x to y
$dn(x,y)$	Default next-hop from x to y
$bn(x,y)$	Backup next-hop set form x to y

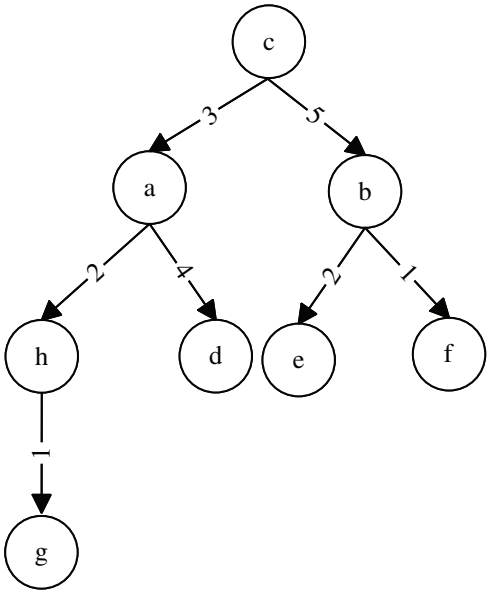

Figure 1: SPT rooted at node c

To implement DC rule at node c, node c should obtain the values of $cost(c,d)$ and $cost(x,d)$. The value of $cost(c,d)$ can be achieved easily via $spt(c)$. However, to obtain $cost(x,d)$, node c needs to compute a SPT for each of its neighbor. Computational complexity increases with the number of network node degree, which is particularly high when a node has a high degree in large-scale networks. Therefore, implementing the DC rule in real ISP networks is not considered a scalable method. For the actual deployment on the Internet, a DC-based scheme should introduce a small additional burden on the current deployed routing protocol. This paper is dedicated to finding an efficient DC-based scheme that is suitable for an ISP network. In particular, we focus on addressing the following problems:

Given a computing node c and its SPT $spt(c)$, we can find an efficient DC-based algorithmic technique in large-scale networks, and the algorithm conforms to the two following conditions:

1. The time complexity of the algorithm is less than that of constructing a SPT.
2. It can provide the same network availability with DC.

4 ERPA and its Performance

4.1 Algorithm

ERPA will be discussed in detail to solve the above problem in this section. The problem can be presented to compute $cost(x,d)$ in the $spt(c)$. We first provide two theorems before formally describing the details of ERPA. The two following theorems describe how to compute the backup next-hop set that satisfies the DC Rule. Moreover, the computation overhead can be reduced dramatically by lessening the times of the operation.

Theorem 1: Given a computing node c and $spt(c)$, for any node $x \in N(c)$, $sptnew(c,x)$ is the new SPT rooted at node c when the weight of link (c,x) is changed to 0. For any node $v(v \neq c, v \neq x)$, if $v \notin D(spt(c),x)$ and $\in D(sptnew(c),x)$, then we can obtain $cost(x,v) < cost(c,v)$.

Proof: Assuming that $dn(c,v) = y, y \neq x$ in the $spt(c)$, we have $cost(c,v) = cost(c,y) + cost(y,v)$.

Given that $v \in D(sptnew(c),x)$, we can obtain $costnew(c,v) = cost(c,x) + cost(x,v)$., where $costnew(c,v)$ is the cost from node c to node v in the $sptnew(c,x)$. Because $cost(c,x) = 0$ in the $sptnew(c,x)$, we can get $costnew(c,v) = cost(x,v)$ (1). According to that $v \notin D(spt(c),x)$and $v \in D(sptnew(c),x)$, we can obtain $costnew(c,v) < cost(c,v)$(2). Combining Eqs. (1) and (2), we have $cost(x,v) < cost(c,v)$.

Theorem 2: Given a computing node c and $spt(c)$, for any node $x \in N(c)$, $sptnew(c,x)$ is the new SPT rooted at node c when the weight of link (c,x) is changed to 0. For any node $v(v \neq c, v \neq x)$, if $v \notin D(spt(c),x)$ and $v \in D(sptnew(c),x)$, then we can obtain $bn(c,v) = bn(c,v) \cup \{x\}$.

Proof: As seen in Theorem 1 and DC rule, node x is a viable backup next-hop from node c to node v; therefore, we can obtain $bn(c,v) = bn(c,v) \cup \{x\}$.

We will use an example to explain **Theorems 1 and 2**. Fig. 1 shows a SPT rooted at node c , whereas Figs. 2 and 3 represent the new SPT when links (c,a) and (c,b) are changed to 0, respectively. Because $e \notin D(spt(c),a)$, $e \in D(sptnew(c),a)$, node a can be a viable backup next-hop from c to e. Given that $d \notin D(spt(c),b)$, $d \in D(sptnew(c),b)$, node b can be a viable backup next-hop from c to d. Considering that $g \notin D(spt(c),b)$, $g \in D(spt(c),b)$, node b can be a viable backup next-hop from c to g.

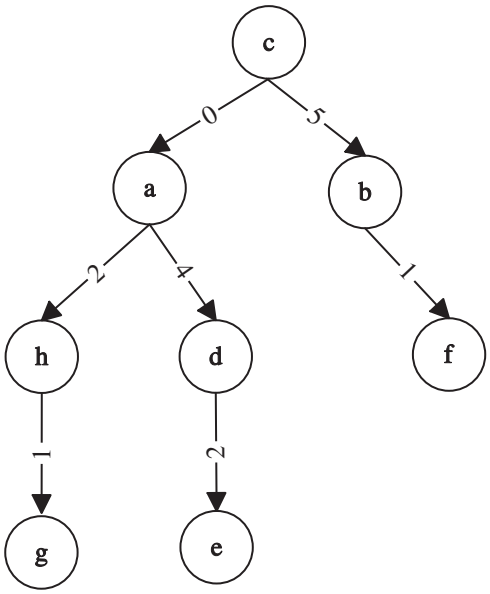

Figure 2: SPT rooted at node c when the weight of link (c,a) is changed to 0

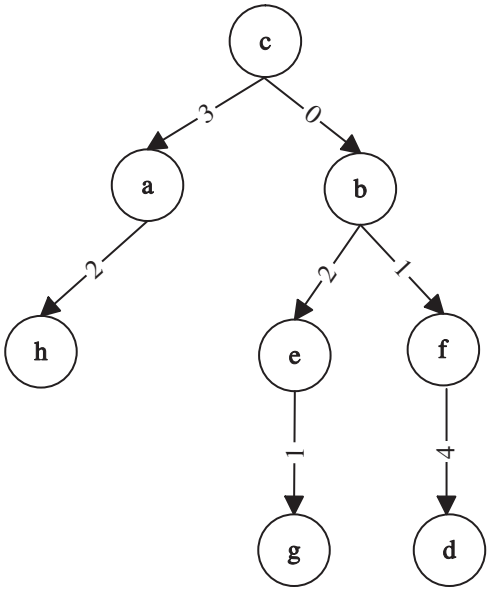

Figure 3: SPT rooted at node c when the weight of link (c, b) is changed to 0

According to the above discussions, ERPA is proposed to compute the backup next-hop set that satisfies the DC rule. The inputs of ERPA include the network topology $G = (V, E)$ and $spt(c)$, and the output is the backup next-hop set from node c to all the other nodes in the network. First, for each neighbor x of c, the weight of the link (c, v) is changed into 0 (lines 2–3), and then a new SPT is built by employing i-SPF (line 4). For any node $v(v \neq c, v \neq x)$, if $v \notin D(spt(c), x)$ and $v \in D(sptnew(c), x)$, then the node x can be a viable backup next-hop from c to v (lines 5–9). At last, the weight of the link (c, x) is adjusted to its original value (line 10).

Algorithm ERPA

Input:

$G = (V, E)$ and $spt(c)$

Output:

$v \in V$ $bn(c, v)$

1: **For** $v \in N(c)$ **do**
2: $weight \leftarrow w(c, x)$
3: $w(c, x) \leftarrow 0$
4: employ i-SPF to construct $sptnew(c)$
5: **For** $v \notin V$ **do**
6: **If** $v \in D(spt(c), x)$and $v \in D(sptnew(c), x)$**then**
7: $bn(c, v) = bn(c, v) \cup \{x\}$.
8: **EndIf**
9: **EndFor**
10: $w(c, x) \leftarrow weight$

4.2 Algorithm Performance and Discussion

In this section, we will show the performance of the algorithm, including the time complexity and the number of backup next-hop computed by ERPA. Theorem 3 suggests that the computational complexity of ERPA is less than that of constructing a SPT. In Theorem 4, ERPA can compute all the backup next-hop sets that satisfy the DC Rule. We will describe Theorems 3 and 4 in detail and prove their correctness.

Theorem 3: Computational complexity of ERPA is less than $O(|E|\lg|V|)$.

Proof: To compute all the backup next-hop set from node c to other nodes in the network. ERPA needs to run the i-SPF algorithm k times, where k is the number of neighbors of node c. Let N_i and M_i indicate the number of nodes that must adjust their costs or parents and the number of edges attached to these nodes when the weight of link $(c,i), i \in N(c)$ is changed to 0, respectively. Therefore, the computational complexity of ERPA is

$$\sum_{i=1}^{k} M_i * lgN_i \leq \lg|V| \sum_{i=1}^{k} M_i = O(|E| \lg|V|) \tag{1}$$

Considering $N_i < |V|$, the computational complexity of ERPA is less than that of SPF.

Theorem 4: ERPA can compute all the backup next-hop set that satisfies the DC Rule.

Proof: We will prove the theorem by contradiction. Supposing that node $v(v \neq c, v \neq x)$, $v \notin D(spt(c),x)$ and $cost(x,v) < cost(c,v)$ $x \notin bn(c,d)$ exist when ERPA is terminated. For any node $x \in N(c)$, if $cost(x,v) < cost(c,v)$, then $v \in D(sptnew(c),x)$. Therefore, according to Theorem 2, we can obtain. Thus, $x \in bn(c,v)$ contradicts the assumptions.

5 Performance Evaluations

In this section, we will evaluate ERPA in terms of computation overhead and network availability. To indicate the performance of ERPA, we compare the results with TBFH, DMPA, and DC. All the algorithms are implemented on a PC (Intel i7, 3.7 GHz CPU, and 8G memory). All of the experimental results correspond to the average values of 15 random experiments. To truly reflect the link failure distribution, the failure probability of links in this paper adopts Weibull distribution

We conduct the simulations on a wide space of relevant topologies, including real, inferred, and synthetic ones. The real topology includes Abilene, USLD, ITALY, NJLATA, and TORONTO [16], as well as six ISP topologies that are inferred from the measurement results from Rocketfuel [17]. The parameters for real and Rocketfuel topology are summarized in Tab. 2. We also use BRITE [18] to generate some topologies, the numbers of nodes range from 100 to 1000, and the average node degree ranges from 5 to 40. The detailed parameters for BRITE are shown in Tab. 3.

5.1 Computation Complexity

Theoretical analysis has indicated that the time complexity of ERPA is less than that of constructing a SPT, which has a great advantage over DC, TBFH, and DMPA. To further verify computational performance, we make simulations on different types of topologies. In this section, we evaluate the computational overhead of different algorithms on three types of topologies to avoid the uncertain impact of factors on the algorithm's performance. The computational overhead of an algorithm is defined as the ratio of computation time of the algorithm to that of constructing a SPT.

Fig. 4 indicates the computational overhead obtained by different algorithms on real and Rocketfuel topologies. Fig. 4 shows that ERPA has the lowest computation overhead among all the algorithms. The computation overhead of ERPA is less than building a SPT, whereas DMPA need to construct a SPT, and

TBFH need to compute two SPTs. The computation overhead of DC is proportional to the degree of the network average node degree.

Table 2: Parameters for Rocketfuel Topology

Topology Name	#Node	#Link
Telstra	108	153
Sprint	315	972
Ebone	87	162
Tiscali	161	328
Exodus	79	147
Abovenet	128	372
Abilene	11	14
USLD	28	45
ITALY	21	36
NJLATA	11	23
TORONTO	25	55

Table 3: Parameters for BRITE Topology

Model	N	HS	LS
Waxman	20–200	1000	100
m	NodePlacement	GrowthType	alpha
2–25	Random	Incremental	0.15
beta	BWDist	BwMin-BwMax	model
0.2	Constant	10.0–1024.0	Router

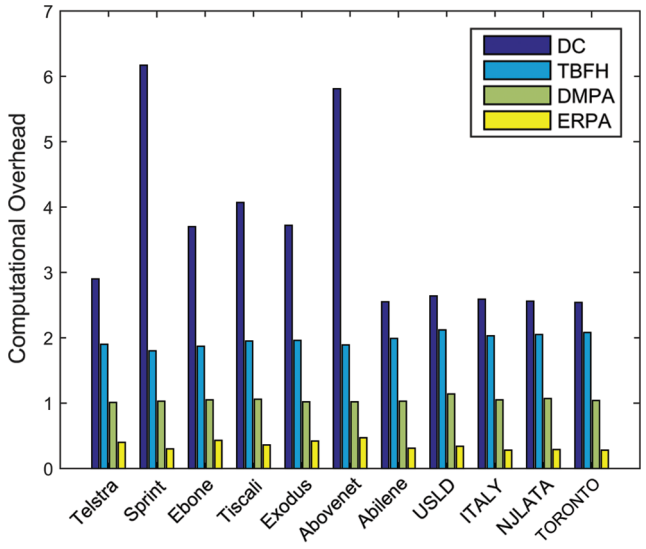

Figure 4: Computational overhead on Real and Rocketfuel topologies

Fig. 5 illustrates the relationship between the computation overhead and topology size on generated topologies when the average node degree is 8. In Fig. 5, the computation overhead does not depend on the topology size for all the four simulated algorithms. The computation overhead of ERPA is lowest in the four algorithms among all the topologies.

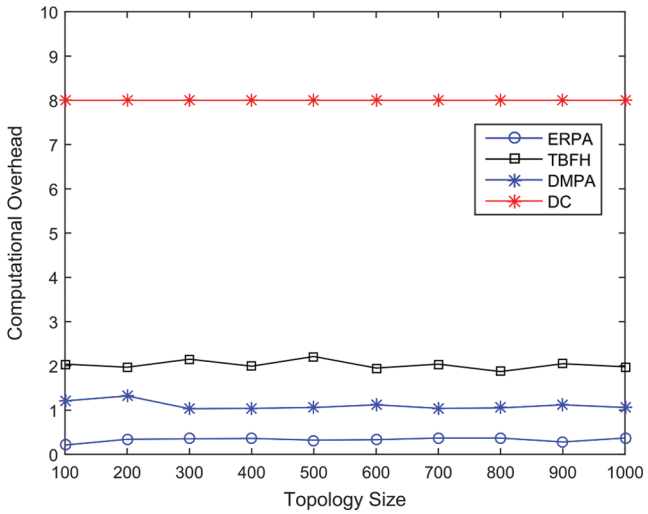

Figure 5: Computational overhead on Brite topologies

Fig. 6 shows the relationship between the computation overhead and average node degree on Brite topologies when the topology size is 1000. As the average node degree increases, the computation overhead of DC rises accordingly. Also, ERPA has exhibited the best performance among all of the tested algorithms. Therefore, the above experiment results are consistent with the theoretical analysis on computational complexity.

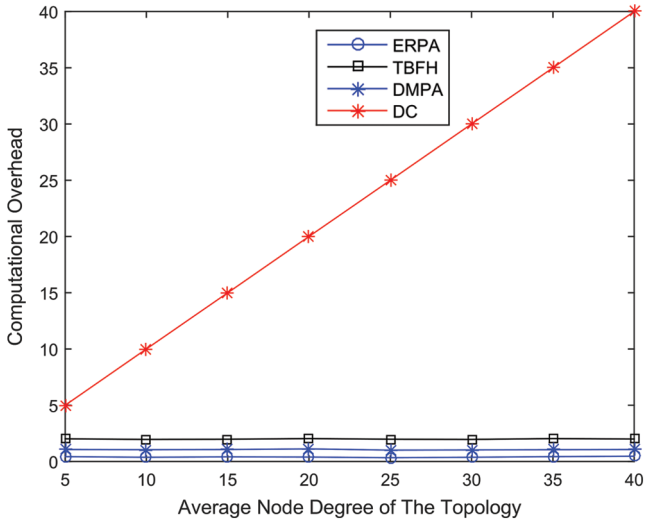

Figure 6: Computational overhead on Brite topologies

5.2 Network Availability

In this section, we will employ network availability as our main evaluation criterion to assess the reliability of the network. Network availability $A(G)$ can be formally defined as:

$$A(G) = \sum_{s,d \in V, s \neq d} A(s,d) \tag{2}$$

where $A(s,d)$ is the availability of source–destination $s-d$ pairs. We will describe the computation of $A(s,d)$ in the following. Supposing that n different paths exist from s to d, we use $p_i(s,d)$ to denote the i-th path in them. We use $A_i(s,d)$ to indicate $p_i(s,d)$ works. The probability of $A_i(s,d)$ can be written as:

$$P(A_i(s,d)) = \prod_{(m,n) \in P_l(s,d)} (1 - r(m,n)) \tag{3}$$

According to the inclusion–exclusion principle, $A(s,d)$ be expressed as:

$$A(s,d) = \sum_{k=1}^{n} (-1)^{(k-1)} S_k \tag{4}$$

where S_k is the total sum of the probabilities that a unique set of k paths from s to d are working simultaneously and can be expressed as:

$$S_k = \sum_{i<j<\ldots k} P\left(A_i \cap A_j \cap \ldots A_k\right) \tag{5}$$

$$\sum_{i<j<\ldots k} \prod_{(m,n) \in P_i(s,d) \cup P_j(s,d) \ldots \cup P_k(s,d)} r(m,n) \tag{6}$$

Tab. 4 provides the network availability provided by each protection scheme on real and inferred network topologies. The results show that ERPA has a clear advantage over TBFH and DMPA and has the same performance as DC. Therefore, the experimental results of network availability are consistent with those of the theoretical analysis.

Table 4: Network availability on Abilene and Rocketfuel topologies

Network	Network Availability (%)			
	ERPA	DC	TBFH	DMPA
Telstra	98.35	98.35	95.34	96.21
Sprint	97.23	97.23	93.46	95.36
Ebone	96.45	96.45	94.32	95.57
Tiscali	98.23	98.23	95.43	96.46
Exodus	92.34	92.34	86.45	89.45
Abovenet	95.34	95.34	92.34	93.76
Abilene	97.85	97.85	94.34	95.71
USLD	91.35	91.35	85.49	87.37
ITALY	92.84	92.84	86.39	88.28
NJLATA	91.27	91.27	86.45	89.64
TORONTO	90.65	90.65	85.86	88.29

Fig. 7 illustrates the relationship between the network availability and the average node degree. The network availability increases with the average node degree. Notably, when the average node degree increases, all schemes provide better network availability results, whereas ERPA and DC are always better than those of DMPA and TBFH. Fig. 8 shows the relationship between network availability and topology size on generated topologies when the average node degree is 6. Fig. 8 shows that the network availability performance of ERPA and DC is obviously better than the two other algorithms. From the experiment, we can conclude that ERPA not only reduces the complexity of DC implementation greatly but also has the same routing availability as DC.

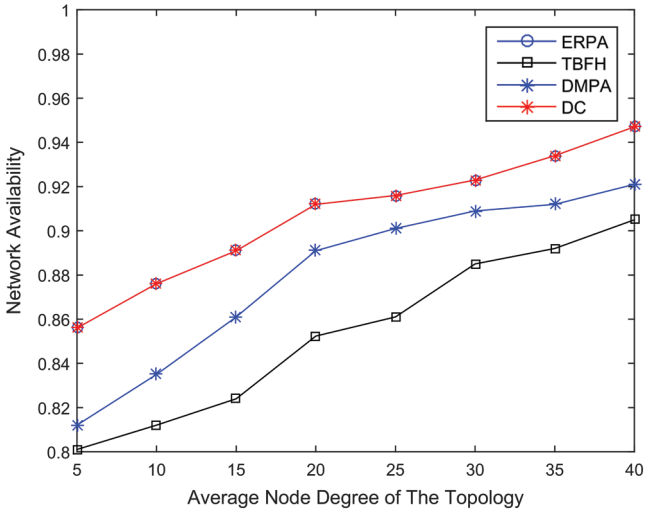

Figure 7: Network availability on Brite topologies

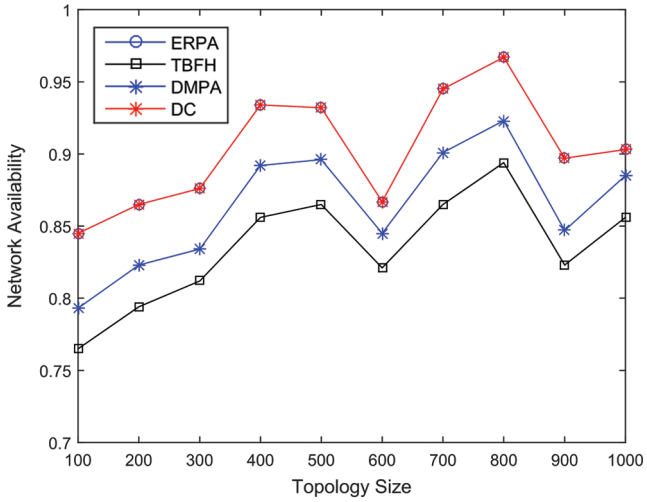

Figure 8: Network availability on Brite topologies

6 Conclusions

This study proposed an efficient scheme called ERPA to implement DC-based hop-by-hop routing protection. The computation complexity of ERPA is irrespective of the degree of the calculating router

and is less than a full SPT calculation. We simulate ERPA on numerous topologies in comparison with DC, DMPA, and TBFH. The theoretical and experimental results show that ERPA can reduce the computational overhead and can provide the same network availability as DC dramatically. We are convinced that our proposed scheme ERPA takes a big step toward actual deployment.

Funding Statement: This work is supported by the National Natural Science Foundation of China (No. 61702315), the Key R&D program (international science and technology cooperation project) of Shanxi Province China (No. 201903D421003), the National Key Research and Development Program of China (No. 2018YFB1800401).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. Klaus-Tycho, P. Yvonne-Anne, S. Stefan and T. Gilles, "Local fast failover routing with low stretch," *ACM Sigcomm Computer Communication Review*, vol. 48, no. 1, pp. 35–41, 2018.

[2] Z. Yang and K. L. Yeung, "SDN candidate selection in hybrid IP/SDN networks for single link failure protection," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 312–321, 2020.

[3] S. Petale and J. Thangaraj, "Link failure recovery mechanism in software defined networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1285–1292, 2020.

[4] S. G. Kulkarni, G. Liu, K. K. Ramakrishnan, M. Arumaithurai, T. Wood *et al.,* "REINFORCE: Achieving efficient failure resiliency for network function virtualization-based services," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 695–708, 2020.

[5] T. Liu and J. C. S. Lui, "FAVE: A fast and efficient network flow availability estimation method with bounded relative error," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 505–518, 2020.

[6] Y. Wang, S. X. Feng, H. T. Guo, X. S. Qiu and H. B. An, "A single-link failure recovery approach based on resource sharing and performance prediction in SDN," *IEEE Access*, vol. 7, pp. 174750–174763, 2019.

[7] J. Q. Zheng, H. Xu, X. J. Zhu, G. H. Chen and Y. H. Geng, "Sentinel: Failure recovery in centralized traffic engineering," *IEEE/ACM Transactions on Networking*, vol. 27, no. 5, pp. 1859–1872, 2019.

[8] X. W. Yang and D. Wetherall, "Source selectable path diversity via routing deflections," in *Proc. of the SIGCOMM*, Pisa, Italy, pp. 159–170, 2006.

[9] H. J. Geng, X. G. Shi, Z. L. Wang, X. Yin and S. P. Yin, "Algebra and algorithms for multipath QoS routing in link state networks," *Journal of Communications and Networks*, vol. 19, no. 2, pp. 189–200, 2017.

[10] M. Motiwala, M. Elmore, N. Feamster and S. Vempala, "Path splicing," in *Proc. of the SIGCOMM*, Seattle, WA, USA, pp. 27–38, 2008.

[11] S. Antonakopoulos, Y. Bejerano and P. Koppol, "Full protection made easy: The DisPath IP fast reroute scheme," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1229–1242, 2015.

[12] S. Cho, T. Elhourani and S. Ramasubramanian, "Independent directed acyclic graphs for resilient multipath routing," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 153–162, 2012.

[13] Y. Yang, M. W. Xu and Q. Li, "Fast re-routing against multi-link failures without topology constraint," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 384–397, 2018.

[14] P. Mérindol, P. Francois, O. Bonaventure, S. Cateloin and J. J. Pansiot, "An efficient algorithm to enable path diversity in link state routing networks," *Computer Networks*, vol. 55, no. 5, pp. 1132–1149, 2011.

[15] H. J. Geng, X. G. Shi, Z. L. Wang and X. Yin, "A hop-by-hop dynamic distributed multipath routing mechanism for link state network," *Computer Communications*, vol. 116, no. 4, pp. 225–239, 2018.

[16] W. Braun and M. Menth, "Loop-free alternates with loop detection for fast reroute in software-defined carrier and data center networks," *Journal of Network and Systems Management*, vol. 24, no. 3, pp. 470–490, 2015.

[17] N. Spring, R. Mahajan, D. Wetherall and T. Anderson, "Measuring ISP topologies with Rocketfuel," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 2–16, 2004.

[18] H. J. Geng, J. Y. Yao and Y. Y. Zhang, "Single failure routing protection algorithm in the hybrid SDN network," *Computers, Materials & Continua*, vol. 64, no. 1, pp. 665–679, 2020.