ARTICLE IN PRESS

Computer Networks xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

An efficient critical protection scheme for intra-domain routing using link characteristics

Mingwei Xu [a], Meijia Hou [a,*], Dan Wang [b], Jiahai Yang [c]

[a] Department of Computer Science and Technology, Tsinghua University, Tsinghua National Laboratory for Information Science and Technology, Beijing, China
[b] Department of Computing, The Hong Kong Polytechnic University, Hong Kong
[c] Network Research Center, Tsinghua University, Beijing, China

ARTICLE INFO

Article history:
Received 24 April 2012
Received in revised form 25 August 2012
Accepted 4 September 2012
Available online xxxx

Keywords:
Intra-domain routing
Link failure protection
Link criticality
Network availability

ABSTRACT

In recent years, there are substantial demands to reduce packet loss on the Internet. Among the proposed schemes, finding backup paths in advance is considered to be an effective method to reduce the reaction time. Very commonly, a backup path is chosen to be the most disjoint path from the primary path, or on the network level, a backup path is computed for each link (e.g., IPFRR). The validity of this straightforward choice is based on two things. The first thing is all the links may have the equal likelihood to fail; the second thing is, facing the high protection requirement today, it just looks weird to have links not protected or to share links between the primary and backup paths. Nevertheless, many studies have confirmed that the individual vulnerability of the links on the Internet is far from being equal. In addition, we have observed that full protection schemes (In this paper, full protection schemes means schemes (1) in which backup path is a most disjoint path from the primary path; or (2) which compute backup path for each link.) may introduce high cost (e.g., computation).

In this paper, we argue that such approaches may not be cost-efficient and therefore propose a novel critical protection scheme based on link failure characteristics. Firstly, we analyze the link failure characteristics based on real world traces of CERNET2 (China Education and Research NETwork 2). The analysis results clearly show that the failure probabilities of the links in CERNET2 backbone are heavy-tailed, i.e., a small set of links causing most of the failures. Based on this observation, we find out two key parameters which strongly impact link criticality and propose a critical protection scheme for both single link failure situation and multi-link failure situation. We carefully analyze the implementation details and overhead for backup path schemes of the Internet today; the problem is formulated as an optimization problem to guarantee the routing performance and minimize the backup cost. This cost is special as it involves computational overhead. Based on this, we propose a novel Critical Protection Algorithm which is fast itself for both the single link failure and the multi-link failure versions. A comprehensive set of evaluations with randomly generated topologies, real world topologies and the real traces from CERNET2, shows that our scheme gains significant achievement over full protection in both single link failure situation and multi-link failure situation. It costs only about 30–60% of the full protection cost when the network relative availability increment is 90% of the full protection scheme.

* Corresponding author.
E-mail address: meijia.hou@gmail.com (M. Hou).

1389-1286/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2012.09.006

1. Introduction

The proliferation of the real time and loss sensitive applications such as virtual lease line services, video services, and stock exchange data services today is far less tolerant to packet loss [2,3]. Internet failures, however, are routine rather than exceptional events [4]. To bridge this gap, many schemes are under active investigation in the network layer to improve the overall Internet performance. One direction focuses on reactive schemes which reduce the network convergence time when a failure occurs [5]. Another direction is to pre-establish backup paths [6]. There are also new protocols proposed that fundamentally switch away from the existing routing paradigm [7,8]. While we believe that providing a satisfiable failure recovery on the Internet is a joint force of different schemes, in this paper, we focus on pre-establishing backup paths, as it provides first reaction upon failures [9].

One very natural backup path design is to find a most disjoint path from the primary path [10], or on the network level, to find backup paths for all links [11]. Nevertheless, the validity needs an assumption. That is all links have the equal likelihood to fail, or at least, due to the high protection requirement of the Internet today, not protecting[1] a single link makes the entire backup scheme futile.

Nevertheless, it can be costly to build an overall protection for the network layer [9]. The computational cost is high for each router, and all the rerouting needs to be stored. Thus, in this paper, we question whether it is possible (i.e., cost-efficient) to have the links selectively protected. We conduct a trace analysis on the link failures in CERNET2 (China Education and Research NETwork 2 [12]). We have observed that the majority of the link failures in CERNET2 are caused by a small set of unstable links. The same effect also has been observed previously in the Sprint network [13]. Consequently, we propose a *critical protection* scheme as a cost-efficient solution for both single link failure situation and multi-link failure situation.

In this paper, we confine our study to intra-domain link state routing system such as OSPF, and we focus on link failures. We pre-establish backup paths to improve the network availability and do not consider bandwidth reservation in this paper.

An Example. Before exposition, we illustrate our critical protection by using the example in Fig. 1. The topology in this figure is a sub-set of the CERNET2 topology. The values of link cost and all other parameters are from CERNET2 topology and its history data. The number of shortest paths (whose source and destination can be any node in the topology) traversing link e_{ij} is denoted as s_{ij}, and the normalized failure of e_{ij} is denoted as f_{ij}. Both s_{ij} and f_{ij} are shown in Table 1. The failure data are taken from CERNET2 history record of July 2008. Due to commercial reasons, we normalized the failure of each link to the failure of link e_{23}, which has the smallest number of link failures in July 2008. The fourth column of Table 1, $s_{ij} \times f_{ij}$, is the total number of (normalized) reroute if the link e_{ij} is down. Intuitively, this shows the impact of having a link e_{ij} protected on the network availability, which is formally defined in Section 4 and 5.

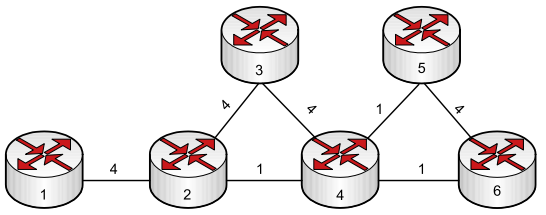


Fig. 1. A sub-set of the CERNET2 topology.

Table 1
Parameters for Fig. 1.

	s_{ij}	f_{ij}	$s_{ij} \times f_{ij}$	$rank_{ij}$
e_{12}	5	13.344	66.72	2
e_{23}	2	1.000	2.00	6
e_{24}	6	2.001	12.01	5
e_{34}	3	30.025	90.08	1
e_{45}	5	4.580	22.90	4
e_{46}	5	6.764	33.82	3
e_{56}	0	11.113	0	7
sum			227.53	

Table 2
Comparison between full protection and critical protection schemes.

Protected links	Protection cost	Protected times	Relative performance (%)
all$-\{e_{12}\}$	6Λ	160.81	100
$\{e_{34}\}$	1Λ	90.08	56.02
$\{e_{34}, e_{46}, e_{45}\}$	3Λ	146.80	91.29

We compare a full protection scheme with a critical protection scheme. From the last row of Table 1, we see that, during the period, there are in total 227.53 times of end-to-end reroute caused by link failures. For simplicity, we temporarily assume the cost for protecting a link to be Λ which is equal to each link. Thus, for full protection in which all links but e_{12} are protected, the protection cost is 6Λ, and a total of 160.81 normalized end-to-end reroute can be avoided.[2] For a critical protection, if e_{34} which ranks first in the $s_{ij} \times f_{ij}$ rank list (see the last column of Table 1) is protected, 90.08 reroute can be avoided, which is 56.02% of the performance of the full protection scheme, with a cost of only 1Λ. If we have a budget of 3Λ, a careful selection on the protected links (e.g., e_{34}, e_{46} and e_{45}) can achieve a relative performance of 91.29%. See Table 2 for the summary.

The above example is not special; from analysis and experiments, we find that different links have different impact on the network availability. This suggests that a brute

[1] In this paper, we use protection and backup interchangeably.

[2] We note that not all the links can be protected, e.g., e_{12} (this reflects a true situation in CERNET2). This indicates that even for a full protection scheme, we can only achieve a protection ratio of 70.68% (160.81/227.53 = 70.68%). As we have argued, in many situations, a joint force of multiple schemes is needed to handle Internet failures; for example, robust physical protection is required for failures on e_{12}. We restrict the scope of this paper, however, on backup path.

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

force protection of all links may not be the best choice, especially when the resource is limited. Nevertheless, to fully explore this opportunity, many issues need to be carefully addressed. Practically, the network availability should be formally defined, the cost should be carefully justified, and the practical issues in the implementation should be considered. Algorithmically, the selection of the links to be protected should optimize system performance.

In this paper, we provide a systematic study for critical protection in the link state routing. We first analyze the failure traces of CERNET2, and then the implementation details and overhead for pre-establishing backup paths are studied. After that, we formulate the critical protection problem. A special challenge that we face is that one major cost for pre-establishing backup paths is the computational overhead. Thus, the selection algorithm itself should be of low complexity so that it will not dominate over computing backup paths. We propose a novel Critical Protection Algorithm that successfully achieves this goal in both single link failure situation and multi-link failure situation. We evaluate our scheme with different topologies constructed by BRITE topology generator [14] and the real world topologies as well as real link failure traces from CERNET2. The results have shown the cost-efficiency of our scheme: in most conditions, with an relative availability increment of 90%, the cost of our scheme is less than 35% to that of the full protection scheme; and even with an relative availability increment of 99%, the cost of our scheme is around 55% to that of the full protection scheme.

The remaining part of the paper proceeds as follows. In Section 2, we present the background and related work. Trace studies, implementation details and the general problem formulation are specified in Section 3. Section 4 is devoted to the algorithm design for single link failure situation. In Section 5, we present the problem formulation and algorithm design for multi-link failure situation. We evaluate our scheme in Section 6 and compare it with a state-of-art method in Section 7. Section 8 concludes the paper and discusses future work.

2. Background and related work

2.1. Routing protection

Failures are common on the Internet today [4,13]. In a typical link state network (e.g., OSPF), when a router detects a failure, it will propagate this information to the network and each router will recalculate its routing table. During this interim period, the routers will have an inconsistent view of the network. Loops and blackholes may occur and cause packet drops. The packet loss can be huge; for example, three million packets (average packet size 1 KB) would be lost if an OC-48 link goes down for 10 s [10]. Such service disruption is unacceptable for current real time applications.

While there are studies that try to reduce the convergence period, in link state routing, pre-establishing backup paths is more effective to reduce the service disruption time to as short as the failure detection time. Finding disjoint paths has long been a research topic in theoretical computer science with various objectives [15–17] and many such problems are NP-hard. In practice, backup path can be pre-established by MPLS, where both the primary path and the backup path can be reserved [6]. This approach introduces large overhead for the virtual paths establishment and requests the infrastructure to be MPLS capable.

Another approach that is currently in heavy investigation is IP Fast Reroute (IPFRR). The IPFRR framework [18] is proposed where alternative paths are identified and entries are added for rerouting in the FIB of each router each time the topology changes. Several simple IPFRR techniques such as equal cost multiple paths (ECMP), and loop free alternates (LFA), only request modification on the forwarding table of the router that is adjacent to the failure. The ECMP or LFA paths, however, may not be found even though an alternative path that can avoid the failure exists. Other IPFRR schemes build tunnels, or specially, use not-via addresses, where each router computes the backup paths for each link. When failure occurs, the packets are encapsulated in a not-via address. All the routers will use this address to forward the packet, until the failure is bypassed. An evaluation of several IPFRR techniques is in [19], and a number of associated techniques for IPFRR, e.g., avoiding mini-loops, can be found in [20,11,21,10].

Pre-establishing backup paths is also widely used as a building block for many protection schemes, e.g., R-BGP [22], in which the most disjoint path of the primary one is used as the backup path.

In all these previous schemes, an intrinsic assumption is that all links are equally treated in protection. Based on our observation, we find that some links undertake more traffic and/or are more vulnerable than the others. We argue that facing resource limitation, the primary and the corresponding backup paths should be disjoint at such links in the path-based protection schemes, and in the link-based protection [1], it is more cost-efficient to protect these links.

2.2. Protection in optical networks

Though the critical protection solutions proposed in this paper is discussed within the field of link-based protection in the IP-layer[3] networks, path/link based protections in the optical networks are also traditionally hot topics in network research, especially the resource assignment patterns which we care most when facing resource limitation.

2.2.1. Link

The term *link* in this paper refers to the network link of IP-layer networks; we call it IP link. IP link is a kind of logical link undertaken by real fibers of the backbone optical networks. Because a real fiber link in the optical networks can undertake many logical IP links, the IP links undertaken by a same fiber will fail simultaneously if the fiber fails, resulting in multi-link failure in the IP-layer network. This is similar to the SRLG (Shared-Risk Link Groups)

[3] In this paper, we use IP layer and network layer interchangeably.

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

concept of optical networks, which means a group of network links that share common physical resources (e.g. cable, node, and conduit) whose failures will lead to failures of all those links in the group [23–25].

Besides, links in the IP-layer networks may also share risks due to some causes from the higher layers, such as the transport layer. For example, after traffic congestion occurs on a primary path, it is inclined to occur on the corresponding backup path, if both of the paths have low bandwidth and undertake heavy traffic. This is because when the primary path is congested under heavy traffic load with low bandwidth, it will switch the huge traffic to its backup path which also only owns low bandwidth, thus resulting in traffic congestion on the backup path soon. We have confirmed with a network operator of CERNET (China Education and Research NETwork) [26] that, this is common in real networks, and actually this phenomenon indeed occurs on a pair of primary and backup paths in CERENET, which urgently need to be upgraded.

Although SRLG auto-discovery methods can be used in a few cases [27,28], in most of the cases, the SRLGs of optical networks need to be obtained manually by the network operators with the knowledge of the physical network structure, such as fiber plant. However, on the IP layer, it is hard to identify the higher layer causes on which shared-risk link groups are based. In our protection solutions, link failures of both the two kinds of shared-risk link groups in IP-layer networks can be well protected (the multi-link failure protection in Section 5), with no need to identify the shared-risk links in advance.

2.2.2. Resource assignment

Previous studies have proposed many optical layer protection schemes, especially for wavelength-division-multiplexing (WDM) optical mesh networks. These protection schemes can be classified into three categories in the aspect of protection granularity: path-based protection [29–31], link-based protection [32], and segment-based protection [33]. In the resource assignment aspect, they can be roughly classified into: dedicated protection [30] and shared protection [29,34]. In both kinds of the schemes, the network resources (e.g. wavelength channels on fiber in WDM networks) are reserved for backup paths or backup segments.

The basic difference between the dedicated protection and the shared protection of the optical networks is whether a wavelength channel can be shared by different backup paths (segments) or primary paths (segments). Take the path-based protection for an example. In the dedicated path protection (DPP) [30], if a wavelength link is part of a backup (primary) path, it cannot work as part of any other backup (primary) path or any primary (backup) path simultaneously, indicating a backup path should be disjoint from any other paths and the spare capacities cannot be shared in any conditions. In the shared path protection (SPP), a wavelength link can be partially and conditionally shared. In the traditional SPP [34], any two backup paths can share the spare capacities in case that their corresponding primary paths are link disjoint due to the single failure constraint. Since SPP has obviously better resource utilization than DPP, there are many subsequent schemes proposed to improve SPP. In [29], the proposed mixed shared path protection (MSPP) scheme makes the spare capacities sharing between primary paths and backup paths become possible with certain constraints, besides the capacities sharing between two backup paths. We notice that although spare network capacities can be partially shared in SPP, the sharing conditions are still very strict.

In the IP-layer IPFRR framework [18], on which the protection solutions proposed in this paper are based, the resources are not reserved and the spare capacities can be freely shared by any kinds of paths. That means, in theory, any paths (primary and/or backup paths) can have common links. Since resources are not reserved for backup in IP-layer IPFRR [18] protection, a link may be overloaded, resulting in congestion. This will show as a link failure on the IP-layer, and can be easily protected by the single link failure protection (Section 4) or multi-link failure protection (Section 5) solutions proposed in this paper. In general, whether spare capacities are shared freely can be treated as one of the basic differences between the resource assignment way of the optical layer protection schemes and that of the IP-layer IPFRR [18] protection schemes.

Stub release [35] is an interesting concept proposed for path restoration in optical networks, which means if it is possible to release the surviving upstream and downstream portions of a failed working path, the freed capacity should be made available to the restoration process. The alternative is to leave the spare capacity as unused working capacity, reserved for the return of normal signal path after physical repair of the failure [35]. In other words, stub release encourages resource reuse in the dimension of time. Though stub release is used in path restoration instead of protection, such resource assignment concept is similar to the basic resource assignment concept of the IP-layer IPFRR protection: resources should be reused in both time and space dimension.

In recent years, a protection method named *Hamiltonian cycle protection* [36] is proposed in optical networks, which has better resource utilization than many other protection schemes in single link failure situation. A Hamiltonian path is a path that visits each vertex **exactly** once in the mathematical field of graph theory; and a Hamiltonian cycle (also called Hamiltonian circuit) is a Hamiltonian path that is a cycle [37]. That means a Hamiltonian cycle is a cycle that traverses all the nodes in a topology. However, not all the topologies contain Hamiltonian cycles and the problems that determining whether such paths and cycles exist in graphs is NP-complete [38]. Fortunately, we can find the Hamiltonian cycles in most of the current real optical backbone networks [36], such as US National, CHINA CERNET, NJLATA and ECNET networks. In the Hamiltonian cycle protection, because the resources are reserved and the backup paths are pre-computed only once instead of being recomputed repeatedly each time the topology changes (when facing link failure/up events), the Hamiltonian cycle which is used to protect all the primary paths/links/segments is usually pre-computed by off-line methods.

Hamiltonian cycle can protect path/link/segment failures in a very simple way that, since both the two ends of each primary path, link or segment are on the Hamilto-

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

nian cycle, the residual available route on the Hamiltonian cycle can be used as their backup paths under the single link failure assumption. Because all the backup paths can share common capacities (spare wavelength channels) along the Hamiltonian cycle, the resources consumed by backup paths can be greatly reduced. The study in [36] has proved that in some cases the resource utilization of Hamiltonian cycle protection is better than that of path-based shared protection, which is believed to have the best resource utilization among path/link/segment-based protections. In [39], the authors improve the resource utilization of Hamiltonian cycle protection with loading balancing by carefully redesigning the routing strategies of the primary paths. Then the performance of Hamiltonian cycle protection is further improved in [40], which proposes a differentiated two-level protection scheme and the lower-level connections are not provided with protection. The basic critical protection idea is similar to that of this paper. Further, Hamiltonian cycle protection is used to solve the protection of multicast problem in [41], and in [42], it is used to solve protection problem in multi-domain optical networking based on local and global Hamiltonian cycles.

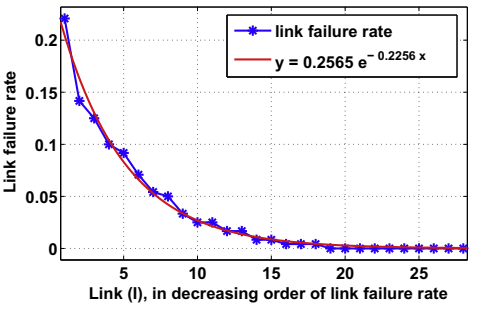
Though the resource assignment way of Hamiltonian cycle protection is interesting and effective in optical networks, it is currently confined to the single link failure situations, because a second link failure may break the original Hamiltonian cycle. Besides, the wavelength channels needed by backup paths and primary paths are still reserved in Hamiltonian cycle protection, just like the traditional protection methods of optical networks.

3. Critical protection: the motivation and problem

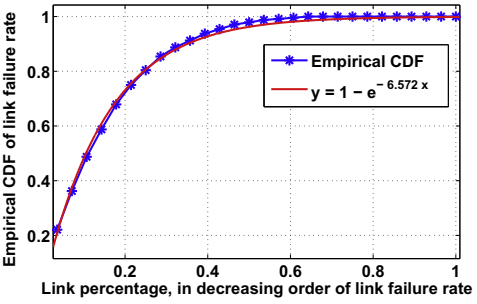
3.1. Data trace study

We conducted trace study on CERNET2, China Education and Research NETwork 2 [12], which consists of 25 nodes, 28 links and spans across most major cities in China, supporting high speed connectivity for more than 200 universities and other research institutions. We collected the network failures from October 10, 2008 to November 2, 2008. During this period, we have observed 240 failures. The failure rate of each link is shown in Fig. 2a. We can clearly see that the failure shows a heavy-tailed behavior and matches an exponential function. In Fig. 2b, we can see that almost 60% of failures are caused by a small set of links, e.g., only 14% (4 out of 28 links). Such a heavy-tailed behavior has also been observed in the Sprint network [4]. Thus a natural question is that as the majority failures are caused by a small set of links, whether a full protection scheme is cost-efficient. This observation motivates us to explore a critical protection scheme.

Another interesting observation lies in multi-link failure. We define single link failure as the event that one link failure occurs and no more link failure occurs at the "same time". That means the convergence periods caused by successively failed links do not overlap. Correspondingly, multi-link failure is the event that, multiple (no less than one) links fail simultaneously. It contains both single link failure



(a) Per-link failure rate, in descending order.



(b) The CDF of the link failure rate.

Fig. 2. Failure rate of links in CERNET2.

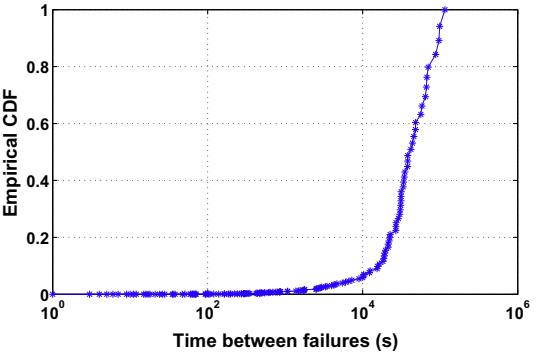


Fig. 3. CDF of the time between two adjacent failures in CERNET2.

and *pure multi-link failure*[4] in which there is at least one more link going down during the convergence period caused by a previously failed link. It is difficult to exactly estimate the time when the network is re-stabilized indicating a convergence period finishes. Even though we capture the time when each router in the network receives LSAs and recomputes the forwarding table, the time clock at each router may still be different. In Fig. 3, if we set the network re-stabilize time to be 10 s, which is considered as a conservative

[4] For example, a fiber may undertake more than one IP links, and thus a fiber breakdown may lead to more one IP link failures. This is a kind of pure multi-link failure we discuss in this paper.

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

number [43–45], the number of pure multi-link failures is almost zero.

Although pure multi-link failures rarely occur in the networks which contains relatively fewer links and nodes such as CERNET2, we believe the possibility does exist, especially in the bigger networks that contains hundreds of links and nodes. Therefore, as another important component, we discuss multi-link failure situation in Section 5 for completeness.

3.2. Implementation details and overhead for pre-establishing backup paths

3.2.1. An overview of the implementation framework

The framework of our critical protection scheme operates as follows:

Step 1: Given a network topology, each time the topology changes, each node computes the set of links to be protected, $\mathcal{L}$ (according to IPFRR methods especially not-via [11]). Algorithms are to be detailed in Section 4 for single link failure situation and Section 5 for multi-link failure situation.

Step 2: Under single link failure situation, for each link e_i ($e_i \in \mathcal{L}$), each node will compute the backup path by first removing e_i from the network and then computing the shortest path tree. The backup path of e_i is the new shortest path between e_i's two ends. Each node on the backup path inserts the backup-path information into its FIB.

The operations are similar in multi-link failure situation. Each node will compute backup path for each link e_i ($e_i \in \mathcal{L}$) in each multi-link failure situation (in which e_i fails) by removing all the failed links in the situation first and then computing the new shortest path tree. Each node on each backup path inserts the backup-path information into its FIB.

Step 3: During the convergence stage of a failure situation:

(a) A special header is used for all the packets that should be forwarded on each currently failed but protected link.
(b) If a node receives a packet with the special header, it will forward the packet by using the corresponding backup path.

The not-via address [11] can be used for this step. We emphasize, however, that our scheme is not restricted to a specific technique.

3.2.2. The overhead

Based on the current link state routing (e.g., OSPF), there are three types of overheads for pre-establishing backup paths.

Computational overhead: Every time the topology changes, each router computes backup paths for the links to be protected (Step 2). Note that for a full protection scheme, the computational cost can be $O(|E| \times SPT)$ in single link failure situation where $|E|$ is the number of links and SPT is the computational cost for a shortest path tree. This poses a high load on routers [9]. In multi-link failure situation, because links should be protected in failure situations of different multi-link combinations, the load for protecting a link is even far much higher.

Memory overhead: All the routers on a backup path of a selected link need to store additional items in their forwarding tables (Step 2). The more links are protected, the more memory overhead is required [9].

Control overhead: A router needs to configure itself to recognize specific headers of potential re-directed packets along the backup paths (Step 3 (b)), and add specific headers when it needs to send packets which should originally go through an adjacent, protected and currently failed link (Step 3 (a)). This control process can be processed by hardware [46], therefore, this cost is usually negligible.

In this paper, we develop a critical protection scheme which can effectively reduce the overall overhead as compared to a full protection scheme. Both single link failure version and multi-link failure version of the critical protection scheme are designed. We admit that in practice, a pure full protection scheme seldom exists alone, and there is often joint work of different schemes. Our work is also designed to be general enough so that it can be incorporated as a building block in various frameworks.

3.3. The critical protection problem

We can model the communication network as an undirected and connected graph $G(V, E)$, where $V = \{v_1, v_2, \ldots, v_{|V|}\}$ is the set of nodes and $E = \{e_1, e_2, \ldots, e_{|E|}\}$ is the set of links. We use P_{sd} to denote the primary path from s to d (this is the shortest path in OSPF).

Our objective is to carefully select a set of links to be protected in order to reduce the overhead, maintaining high network performance at the same time. Formally, we are looking for a link selection scheme

$$B = (b_1, b_2, \ldots, b_i, \ldots, b_{|E|}) \tag{1}$$

where

$$b_i = \begin{cases} 1 & \text{decide to protect } e_i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

We describe our problem more specifically in the following two sections.

4. The critical protection for single link failure

In the single link failure situation, we use $\widetilde{P}_i$ to denote the backup path for link e_i; $|\widetilde{P}_i| = 0$ if link e_i has no backup path, such as the situation of link e_{12} in Fig. 1.

4.1. Network availability

The basic function of the network layer is packet delivery. We use network availability to quantify the network performance.

Failure rate of link e_i is defined as

$$\Delta_i = \frac{f_i \times T_c}{t_{duration}} \tag{3}$$

where f_i is the total number of link failures observed on e_i during the concerned time $t_{duration}$, and T_c is the average

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

convergence time of the network. Note that Δ_i is assumed to be independent with Δ_j ($1 \leqslant i \neq j \leqslant |E|$).

In single link failure situation, convergence periods caused by successive link failures do not overlap. Thus the numerator of above equation's right part $f_i \times T_c$ is the total non-convergence time that link e_i contributes to the network during the concerned period. Therefore, Δ_i is the proportion of the total non-convergence time that e_i contributes to the network to the total time, representing the degree that link failure events on e_i impact the whole network during the concerned period.

Note that, because convergence periods caused by different single link failure do not overlap, we have $\sum_{e_i \in E}(f_i \times T_c) \leqslant t_{duration}$ and $\sum_{e_i \in E}\Delta_i \leqslant 1$.

We use g_i to infer whether link e_i *can* be successfully protected, which denotes whether there exists a backup path of e_i. As such, in single link failure situation, g_i is defined as

$$g_i = \begin{cases} 0 & |\widetilde{P}_i| = 0 \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

The end-to-end availability $A(s,d)$ of a node pair (s,d) is defined as available-to-total ratio in terms of time. Specifically, it is the proportion of the time that packets from s to d can be successfully delivered along either the primary path P_{sd} or the backup paths of the links on P_{sd}, to the total concerned time $t_{duration}$. Formally,

$$A(s,d) = 1 - \sum_{e_i \in P_{sd}} \Delta_i + \sum_{e_i \in P_{sd}} \Delta_i b_i g_i \tag{5}$$

Thus, $A(s,d) \cdot t_{duration}$ is the total available time of node pair (s,d) during our concerned period. It is the sum of, the total time when the network is not in the convergence period caused by any link failure on P_{sd}, and the total time when there exists a **successfully** protected link that goes down on P_{sd} and the network is still in convergence period caused by this failure, corresponding to the two parts of $A(s,d)$ in the above equation (in front of and behind "+").

Network availability should be the concentrated manifestation of all the node pairs' availability in the network. In general, the higher the availability of the node pairs in the network is, the higher the network availability should be. Thus, here we simply define network availability $Availability(G)$ as average availability of node pairs instead of complex proportion or probability.

$$Availability(G) = \frac{1}{C(|V|,2)} \sum_{s,d \in V \wedge s \neq d} A(s,d) \tag{6}$$

where $C(|V|,2)$ is a combination number which is the number of unique node pairs in the network.

We use $Incre\_Avail(G,B)$ to denote the network availability increment of network G under a specific protection scheme B.

$$Incre\_Avail(G,B) = Availability(G)_B - Availability(G)_O \tag{7}$$

where $Availability(G)_B$ and $Availability(G)_O$ are the network availability of network G under protection scheme B and no protection scheme situation respectively.

Finally, $R\_Availability(G,B)$ which we use to describe the relative availability increment of protection scheme B compared to full protection scheme F, is defined as

$$R\_Availability(G,B) = \frac{Incre\_Avail(G,B)}{Incre\_Avail(G,F)} \tag{8}$$

4.2. Cost of the network protection

We quantify the cost for a protection scheme, which consists of the computational cost and memory cost (we neglect the control cost). Because the aim of calculating protection cost is only to study the influence that a protection scheme introduces to the network performance, accurate values may not be necessary and we normalize both the computational and memory cost to certain standard values.

The computational cost for protecting a link e on one node is the cost to compute SPT in network $G - \{e\}$, which is approximately equal for all nodes and all links in the same topology. We denote such a cost as C_{SPT}. Thus, the computational cost for protecting a link e in the whole network is $C_{SPT} \cdot |V|$. We normalize such a value to C_{SPT}. Therefore, the normalized computational cost for protecting any link in the whole network is $\hat{c} = |V|$.

The memory cost is the total entries added in the nodes of G. If we treat the cost that a node pays for adding information of a backup path into its FIB as "standard value", the normalized memory cost for the backup path of a single link e_i in the whole network is $m_i = |\widetilde{P}_i|$, where $|\widetilde{P}_i|$ represents the numbers of nodes on $\widetilde{P}_i$.

To protect link e_i, our backup cost c_i is finally defined as:

$$c_i = \lambda_1 \hat{c} + \lambda_2 m_i \tag{9}$$

where λ_1 and λ_2 are the weights associated with the two types of cost.

4.3. The problem

Given the network G and a relative availability increment requirement Ω which is a non-negative real number, the problem is to search for a link protection scheme B which satisfies the relative availability increment requirement and minimizes the total cost for protecting the network. Formally,

$$\min. \quad C = \sum_{e_i \in E}(c_i \cdot b_i) \tag{10}$$

$$\text{s.t.} \quad R\_Availability(G,B) \geqslant \Omega \tag{11}$$

4.4. The Critical Protection Algorithm

Unlike the conventional optimization problem, a very unique challenge of our problem is that one major overhead of pre-establishing backup paths is the computational overhead. As such, intrinsically, this requests that the computation of the algorithm for link selection, i.e., the Step 1 in 3.2.1, must be very fast itself; and hopefully, negligible as compared with the computation of the backup paths, i.e., the Step 2. Thus, the selection algorithm to be developed should be at smaller order to $O(|E| \times SPT)$.

One observation is that the cost (both the computational cost and the memory cost) is highly correlated to the number of links that need to be protected; and generally, the less, the better. This leads us to focus more on the relative availability increment constraint. We devel-

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

op a novel Critical Protection Algorithm in this paper, the key observation of which is that the impact of every link on the network availability is not the same.

We introduce *criticality* ρ_i for link e_i, which reflects the impact of a link on the network availability.

$$\rho_i = s_i \cdot \Delta_i \tag{12}$$

where s_i is the number of shortest paths in G that traverse through e_i. Intuitively, a link is more critical if 1) its failure rate is high and/or 2) the number of routes that it undertakes is large. We would like to comment that one reason we finally choose this criticality definition for our algorithm development is also because of its simplicity.

Algorithm 1. Critical-protection (single link failure)

Input: G,Ω
Output: $(b_1,b_2,\cdots,b_i,\cdots,b_{|E|})$ AND $\{\widetilde{P}_i|b_i=1\}$.
1 begin
2 /*initialization*/
3 Construct a link list in descending order according to criticality, $(e_1,e_2,\cdots,e_{|E|})$;
4 **for** i = 1 to $|E|$ **do**
5 $b_i \leftarrow 0$;
6 $IncreA\_Bound \leftarrow Incre\_Avail(G,F) \times C(|V|,2) \times \Omega$;
7 $Incre\_Avail'(G,B) \leftarrow 0$;
8 $i \leftarrow 1$;
9 /*select and protect links*/
10 **while** $Incre\_Avail'(G,B) < IncreA\_Bound$ and $i \le |E|$ **do**
11 $b_i = 1$;
12 Compute $\widetilde{P}_i$;
13 **if** $|\widetilde{P}_i| > 0$ **then** $Incre\_Avail'(G,B) \leftarrow Incre\_Avail'(G,B) + s_i \times \Delta_i$;
14 $i \Leftarrow i + 1$;
15 end

We develop the Critical Protection Algorithm for single link failure situation (see Algorithm 1), which iteratively selects those links that are more critical to protect. In our algorithm, we combined Step 1 and Step 2; that is, the output of the CP algorithm[5] contains not only the selected links, but also the backup paths computed for these links. The reason for combining these two steps is that every time a link is selected, we must verify whether its backup path exists (see Algorithm 1 line 13). If not, such selection is not valid. Besides, in Algorithm 1, $Incre\_Avail'(G,B)$ is actually $C(|V|,2) * Incre\_Avail(G,B)$ rather than the exact $Incre\_Avail(G,B)$ for computational simplicity. Therefore, each round a new link e_i is successfully protected, $Incre\_Avail'(G,B)$ just needs add the incremental part e_i protection brings ($s_i \times \Delta_i$, Algorithm 1 line 13).

In this paper, we choose $R\_Availability(G,\cdot)$ as the main parameter to evaluate protection schemes' performance instead of $Availability(G)$ which is used in our preliminary version [1], because $R\_Availability(G,\cdot)$ is a more reasonable character to evaluate a protection scheme. $R\_Availability(G,\cdot)$ can indicate the relative performance improvement of a protection scheme compared to full protection. Therefore, to judge the performance of a protection scheme, it is better than the absolute value of network availability $Availability(G)$ which is easily affected by the topology itself.

Besides, to compute $R\_Availability(G,\cdot)$, due to the newly defined $A(s,d)$ in this paper, we can simply add an additional operation each time a backup path is successfully found (Algorithm 1 line 13). Therefore, the computational complexity of our algorithm is at the same order with that of backup path computation.

5. The critical protection for multi-link failure

In this section, we study the critical protection problem in multi-link failure situation. Specifically, it is in the situation $\mathbb{S}_{\mathcal{K}}$ that there are no more than $\mathcal{K}$ ($1 \le \mathcal{K} \le |E|$) simultaneous link failures in network G.

We consider multi-link failure event M_t that there are t ($1 \le t \le \mathcal{K}$) simultaneous link failures in network G, in which $e_{i_1}, e_{i_2}, \ldots, e_{i_t} \in E$ ($i_1 \ne i_2 \ne \cdots \ne i_t$) are the failed links and all the other links are not. We use $\widetilde{P}^{i_j}_{i_1\cdots i_t}$ to denote the backup path for link e_{i_j} ($1 \le j \le t$) when M_t occurs; $|\widetilde{P}^{i_j}_{i_1\cdots i_t}| = 0$ if link e_{i_j} has no backup path when M_t occurs.

Thus, the link failure event set of situation $\mathbb{S}_{\mathcal{K}}$ is

$$F(\mathbb{S}_{\mathcal{K}}) = \{M_t | 0 \le t \le \mathcal{K}\} \tag{13}$$

M_0 denotes the special "failure event" that there is no link failure in the network. Single link failure event which is M_1 is also a kind of multi-link failure here, and single link failure situation which we discuss in Section 4 is situation $\mathbb{S}_1$.

Let's reconsider our objective function in 3.3. In multi-link failure situation $\mathbb{S}_{\mathcal{K}}$, $b_i = 1$ represents link e_i is chosen to be protected for all the link failure events in $F(\mathbb{S}_{\mathcal{K}})$, and $b_i = 0$ represents link e_i is chosen to be protected for none of the link failure events in $F(\mathbb{S}_{\mathcal{K}})$.

5.1. Network availability

Failure rate for multi-link failure event M_t is defined as

$$\Delta_{i_1\cdots i_t} = \frac{f_{i_1\cdots i_t} \times T_c}{t_{duration}} \tag{14}$$

where $f_{i_1\cdots i_t}$ is the number of times that M_t occurs during the concerned period $t_{duration}$, and T_c is the average convergence time of the network. Thus, $\Delta_{i_1\cdots i_t}$ is the proportion of the total non-convergence time that M_t contributes to the network to the total time, representing the degree that multi-link failure M_t impacts the whole network during the concerned time period.

We use $g^{i_j}_{i_1\cdots i_t}$ to infer whether link e_{i_j} *can* be successfully protected under M_t.

$$g^{i_j}_{i_1\cdots i_t} = \begin{cases} 0 & |\widetilde{P}^{i_j}_{i_1\cdots i_t}| = 0 \\ 1 & \text{otherwise} \end{cases} \tag{15}$$

To describe the end-to-end availability of a node pair (s,d) in multi-link failure situation $\mathbb{S}_{\mathcal{K}}$, we define

[5] In this paper, CP algorithm is short for Critical Protection Algorithm.

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

$$\mathcal{U}(s,d)=\sum_{t\in[1,\mathcal{K}]}\sum_{p\in[1,min\{t,|P_{sd}|\}]}\sum_{e_{i_1},\ldots,e_{i_p}\in P_{sd}}\sum_{e_{i_{p+1}},\ldots,e_{i_t}\in E-P_{sd}}\left(\Delta_{i_1\cdots i_t}\cdot\prod_{1\leqslant q\leqslant p}b_{i_q}\cdot\prod_{1\leqslant q\leqslant p}g^{i_q}_{i_1\cdots i_t}\right)\tag{16}$$

where $|P_{sd}|$ is the number of nodes along P_{sd}. From the above equation, we can see that $\mathcal{U}(s,d)\cdot t_{duration}$is the total time when there are some successfully protected links that fail on P_{sd} and the network is still in convergence process. It is the total time that the protection scheme wins for (s,d)'s end-to-end availability from convergence processes caused by link failures within the concerned period, because the failed links on P_{sd} are successfully protected.

$$\mathcal{F}(s,d)=\sum_{t\in[1,\mathcal{K}]}\sum_{p\in[1,min\{t,|P_{sd}|\}]}\sum_{e_{i_1},\ldots,e_{i_p}\in P_{sd}}\sum_{e_{i_{p+1}},\ldots,e_{i_t}\in E-P_{sd}}\Delta_{i_1\cdots i_t}\tag{17}$$

Similarly, $\mathcal{F}(s,d)\cdot t_{duration}$ is the total time when there are multi-link failure events which contains failed link on P_{sd} and the network is still in the convergence process caused by such failures. It is the total non-convergence time that multi-link failure events which contain link failures on P_{sd} bring to the network, if no protection scheme exists.

Note that, we assume the link failures, the convergence periods caused by which overlap, belong to the same multi-link failure event. Thus convergence periods caused by different multi-link failure events do not overlap. Therefore, we have $\mathcal{F}(s,d)\leqslant 1$.

The end-to-end availability of node pair (s,d) is defined as

$$A(s,d)=1-\mathcal{F}(s,d)+\mathcal{U}(s,d)\tag{18}$$

$A(s,d)\cdot t_{duration}$ is the total availability time of node pair (s,d) during the concerned period. It is the sum of, the total time when there is no link failure that occurs on P_{sd} (corresponding to $1-\mathcal{F}(s,d)$), and the total time when there exist **successfully** protected links that fail on P_{sd} and the network is still in the convergence period caused by these failures (corresponding to $\mathcal{U}(s,d)$).

Similar to that of single link failure situation, the network availability is defined as

$$Availability(G)=\frac{1}{C(|V|,2)}\sum_{s,d\in V\wedge s\neq d}A(s,d)\tag{19}$$

The increment of network availability of network G under protection scheme B is

$$Incre\_Avail(G,B)=Availability(G)_B-Availability(G)_O\tag{20}$$

where $Availability(G)_B$ and $Availability(G)_O$ are the network availability under protection scheme B and no protection scheme situation respectively.

$R\_Availability(G,B)$, which we use to describe the relative availability increment of protection scheme B compared to full protection, is defined as

$$R\_Availability(G,B)=\frac{Incre\_Avail(G,B)}{Incre\_Avail(G,F)}\tag{21}$$

5.2. Cost of the network protection

For link failure event $M_t\in F(\mathbb{S}_{\mathcal{K}})$, the computational cost for protecting link e_{i_j} $(1\leqslant j\leqslant t)$ on one node is to compute SPT in network $G-\{e_{i_1},\ldots,e_{i_t}\}$. Though the topology size in such computation may have a little difference with different t, we assume the difference is too small to affect the result, and such a cost (denoted as C_{SPT} defined in 4.2) is also almost same for all nodes and all links in the same network. Thus, in a network G under situation $\mathbb{S}_{\mathcal{K}}$, the computational cost for protecting link e_k $(e_k\in E)$ on one node is $C_{SPT}\cdot\sum_{1\leqslant t\leqslant\mathcal{K}}C(|E|-1,t-1)$, where $\sum_{1\leqslant t\leqslant\mathcal{K}}C(|E|-1,t-1)$ is the number of failure events (M_t) that contain e_k as a failed link in situation $\mathbb{S}_{\mathcal{K}}$. Using C_{SPT} as standard value, the normalized computational cost for protecting link e_k in the whole network G under the situation $\mathbb{S}_{\mathcal{K}}$ is

$$\hat{c}^k_{\mathcal{K}}=|V|\cdot\sum_{1\leqslant t\leqslant\mathcal{K}}C(|E|-1,t-1)\tag{22}$$

Based on the same definition of memory cost and standard value in 4.2, the normalized memory cost for protecting link e_{i_j} under failure event M_t in the whole network is $m^{i_j}_{i_1\cdots i_t}=|\widetilde{P}^{i_j}_{i_1\cdots i_t}|$. Thus, the normalized memory cost for protecting link e_k under the situation $\mathbb{S}_{\mathcal{K}}$ is

$$m^k_{\mathcal{K}}=\sum_{1\leqslant t\leqslant\mathcal{K}}\sum_{e_k\in\{e_{i_1},\ldots,e_{i_t}\}}|\widetilde{P}^k_{i_1\cdots i_t}|\tag{23}$$

Therefore, backup cost for protecting link e_k in network G under situation $\mathbb{S}_{\mathcal{K}}$ is finally defined as

$$c^k_{\mathcal{K}}=\lambda_1\hat{c}^k_{\mathcal{K}}+\lambda_2 m^k_{\mathcal{K}}\tag{24}$$

where λ_1 and λ_2 are the weights associated with the two types of cost.

5.3. The problem

Given the network G and a non-negative relative availability increment requirement Ω, the problem is to search for a link protection scheme B which satisfies the relative availability increment requirement and minimizes the total cost for protecting the network. Formally,

$$\min\quad C=\sum_{e_k\in E}(c^k_{\mathcal{K}}\cdot b_k)\tag{25}$$

$$\text{s.t.}\quad R\_Availability(G,B)\geqslant\Omega\tag{26}$$

5.4. The Critical Protection Algorithm

The Critical Protection Algorithm for multi-link failure situation is similar to that for single link failure situation. The *criticality* ρ_k of link e_k is defined as

$$\rho_k=s_k\cdot\hat{\Delta}_k\tag{27}$$

where s_k is the number of shortest paths in G that traverse through e_k, and $\hat{\Delta}_k$ is the sum of the failure rate of the multi-link failure events in which e_k is one of the failed links. $\hat{\Delta}_k$ is the proportion of the total non-convergence time that all failure events that contain e_k as a failed link contribute to the network to the total time, representing the degree

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

that failures on e_k impact the whole network during the concerned time period.

$$\hat{\Delta}_k = \sum_{1 \leqslant t \leqslant \mathcal{K}} \Delta_{i_1 \cdots i_{t-1} k} \tag{28}$$

Algorithm 2. MultiDEAL (k, t)

```
1  begin
2    Incre = 0;
3    for all P_sd containing e_k do
4      for p = 0 to (min{t,|P_sd|} − 1) do
5        for all {e_{i_1}, ··· , e_{i_p}} ⊆ P_sd and
6        {e_{i_{p+1}}, ··· , e_{i_{t-1}}} ⊆ E−
         P'_sd(e_k ∉ {e_{i_1}, ··· , e_{i_{t-1}}}) do
7          Compute P̃^k_{i_1···i_{t-1}k};
8          if |P̃^k_{i_1···i_{t-1}k}| = 0 then g^k_{i_1···i_{t-1}k} ← 0;
9          else g^k_{i_1···i_{t-1}k} ← 1;
10         if b_{i_1} = ··· = b_{i_p} = 1 and
           g^{i_1}_{i_1···i_{t-1}k} = ··· = g^{i_p}_{i_1···i_{t-1}k} = 1 then
11           Incre ← Incre + Δ_{i_1···i_{t-1}k};
12   return Incre;
13 end
```

Algorithm 3. Critical-Protection (multi-link failure)

```
   Input: G, Ω, K
   Output: (b_1,b_2, ··· ,b_k, ··· ,b_|E|) AND
   {P̃^k_{i_1···i_t} | b_k = 1, ∀M_t ∈ F(S_K)}.
1  begin
2    /*initialization*/
3    Construct a link list in descending order
     according to criticality, (e_1,e_2, ··· ,e_|E|);
4    for k = 1 to |E| do
5      b_k ← 0;
6    IncreA_Bound ← Incre_Avail(G,F) × C(|V|,2) × Ω;
7    Incre_A'(G,B) ← 0; k ← 1;
8    /*select and protect links*/
9    while Incre_A'(G,B) < IncreA_Bound and k ⩽ |E|
     do
10     /*compute for e_k under all M_t containing e_k
       as a failed link*/
11     b_k = 1;
12     Compute P̃^k_k;
13     if |P̃^k_k| > 0 then
14       Incre_A'(G,B) ← Incre_A'(G,B) + s_k × Δ_k;
15       for t = 2 to K do
16         Incre_A'(G,B) ← Incre_A'(G,B) +
           MultiDEAL(k,t);
17     k ← k + 1;
18 end
```

We develop the Critical Protection Algorithm for multi-link failure situation. It is shown in Algorithm 2 and Algorithm 3, in which Algorithm 3 is the main algorithm and Algorithm 2 is the sub-function to deal with pure multi-link failure. This multi-link failure version algorithm chooses to protect those links which are more critical in order to minimize the protection cost while maintaining performance, and also combines the Step 1 and Step 2 of 3.2.1 due to the same reason in the single link failure version.

In this multi-link failure version, if a link is selected to be protected, it will be protected in all the multi-link failure events in which there are less than $\mathcal{K}$ simultaneous link failures containing e_k failure. Thus, backup paths for link e_k in both single link failure and pure multi-link failure events are computed (see Algorithm 3 line 12 and Algorithm 2 line 7). Also only one additional operation needs to be added each time a backup path is successfully found (see Algorithm 3 line 16 and Algorithm 2 line 11) as same as that in single link failure situation shown in Algorithm 1.

6. Simulation and experiments

6.1. Methodology

We evaluate our CP algorithm in both real/sub-real topologies, including CERNET2 [12], Abilene [47], DFN [48] and AT&T [49], and a variety of generated topologies. The network size ranges from a handful of nodes to hundreds of nodes. The reason that we choose such diverse topologies is to study the algorithms' performance under a wide range of scenarios.

BRITE topology generator [14] is chosen to generate random topologies with network size ranging from 40 to 200 nodes on mode *Router only*. The average number of links per new node, m is set to be 2, which directly affects the average node degree. The bandwidth on each link follows a heavy-tailed distribution as an intuitive mechanism for topology generation, and the link cost is a function of the inverse bandwidth. We adopt the same assumption with [19] that the topologies are static in terms of link costs and the link costs are symmetric.

DFN is the German research network and AT&T is a large US continental IP backbone. The sub-real topologies of DFN and AT&T are generated by BRITE under *Bottom up* mode based on the results of [50], in which the authors gives the parameters in detail to generate approximate DFN and AT&T real topologies.

Besides, we also evaluate the schemes with the real Abilene and CERNET2 topologies. Abilene is the nationwide backbone of next-generation research network in US, and CERNET2 is the network backbone of China Next Generation Internet project (CNGI) undertaken by CERNET. CERNET2 provides more than two hundreds universities and many other research institutions located in 25 major cities of China with high speed IPv6 connectivity.

The details of the parameters in BRITE are listed in Table 3, and the network size of the topologies we use are summarized in Table 4.

We evaluate the algorithms in single link failure, two link failure and three link failure situations (the latter

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

Table 3
Parameters of BRITE generator.

	DFN	AT&T	Other topologies
Topo. type	BOTTOM-UP	BOTTOM-UP	Router only
BOTTOM-UP topology parameters			
Grouping model	Random pick	Random pick	
NumAS	17	31	
AS assignment	Constant	Constant	
Inter BWdist	Heavy tail	Heavy tail	
BW range	100–1024	100–1024	
Router parameters			
N	30	154	40–200
Model	GLP	GLP	GLP
α	0.45	0.45	0.45
β	0.64	0.64	0.64
m	3	2	2
Pref. Conn	None	None	None
BWdist	Heavy tail	Heavy tail	Heavy tail
BW range	100–1024	100–1024	100–1024

Table 4
Topology size for evaluation.

	Num of nodes	Num of links
Abilene	10	13
CERNET2	25	28
DFN	30	153
AT&T	154	550
40-topo	40	140
80-topo	80	232
100-topo	100	358
120-topo	120	418
160-topo	160	596
200-topo	200	700

two belong to multi-link failure situation). The failure data are generated as follows: for each link $e_i \in E, f_i$ in both single link failure situation and multi-link failure situation is generated with negative exponential distribution based on the failure characteristic we have learnt from the real trace data of CERNET2. As an illustration, failure rate f_i for the AT&T topology in single link failure situation is shown in Fig. 4. In algorithm evaluation for multi-link failure situation, we assume $\Delta_{i_1 \cdots i_t} = \Delta_{i_1} \cdots \Delta_{i_t}$ for simplicity.

For CERNET2 topology, besides the generated failure data described above, we also use the link failure rate learnt from real trace data of CERNET2 during the time from October 10, 2008 to March 31, 2009 in our evaluation. This reflects the real performance of the algorithms under real network environment and verifies the accuracy of our failure rate generation.

One of the major metrics in evaluation is *backup cost rate*, which is the ratio of the backup cost of CP algorithm to that of full protection. In our evaluation, we use $C/MRate$ to denote different impact between the computational cost and the memory cost. That is $C/MRate = \lambda_1/\lambda_2$. We also use *R_Availability bound* to denote the relative availability increment requirement Ω as another major metrics.

The simulation and experiments are conducted on a PC with Intel Core Duo CPU P8400 2.26 GHz and 3 GB memory.

6.2. Simulation and experiment results

6.2.1. Single link failure

Fig. 5 shows the comparison between our CP algorithm and the full protection scheme on Abilene, DFN and AT&T topologies in single link failure situation. In Fig. 5a, we clearly see that the backup-cost of CP algorithm compared to that of the full protection scheme, goes almost straightly higher with the increase of *R_Availability bound*. It shows

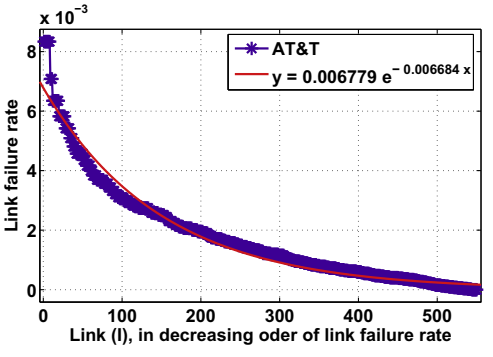
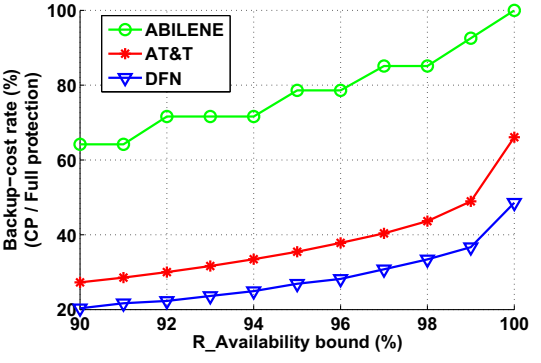
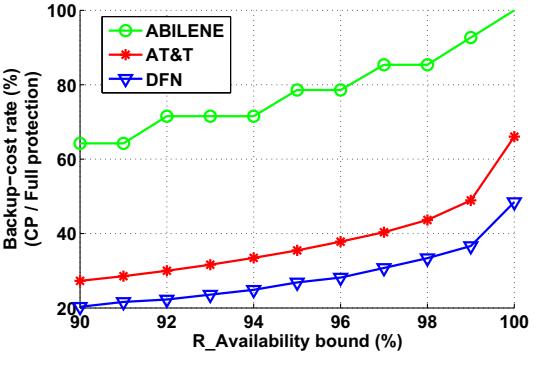


Fig. 4. Failure rate for AT&T topology (single link failure).



(a) $C/MRate = 1$



(b) $C/MRate = 2$

Fig. 5. Results on sub-real topologies and Abilene (single link failure).

that, when the *R_Availability bound* is 90%, our CP algorithm consumes less than 30% of full protection cost in AT&T and DFN topology. For Abilene topology, the cost gain of CP is not that significant. The CP algorithm consumes more than 60% of full protection cost when the *R_Availability bound* is 90%. Note that the DFN topology has 30 nodes and 153 links, the AT&T topology has 154 nodes and 550 links, and the Abilene topology has 10 nodes and 13 links. Thus, the DFN has the largest density while the Abilene has the smallest. Fig. 5 shows a trend that maybe our CP algorithm has better performance in the topologies with larger density if link failure follows the same distribution.

We can also see that even when the CP algorithm reaches the same network availability with full protection (*R_Availability bound* = 100%) in DFN and AT&T topologies, the cost of CP algorithm is less than 50% and 70% of the full protection cost. By looking into the details of our simulation log files, we find that in the generated topologies, there are some *absolutely not* critical links, i.e., the links that undertake no shortest paths, or of which the failure rate is zero. As such, these links do not need to be protected at all in CP.

Comparing Fig. 5a with b, we put different weights on the computational cost and the memory cost. We find that the difference is small. This indicates that the influence of different weights on different types of cost is not obvious. Thus, we use $C/MRate = 1$ in most of the following simulation and experiments.

In Fig. 6 we evaluate our scheme under different topology size with the same network density. We can see that, with different topology size, the performance of our CP algorithm is stable in single link failure situation. This indicates the major evaluation metrics we choose in this paper, *R_Availablity*(·), has effectively eliminated the interference of topology size in our preliminary version of this paper [1] which uses absolute network availability.

We then evaluate the total computing time of the CP algorithm and the full protection scheme, which reflects the real additional overhead of the protection schemes (Step 1 in 3.2.1). It shows that, both on a same topology with different *R_Availability bound* (Fig. 7a) and on a same *R_Availability bound* with different topology size (Fig. 7b),

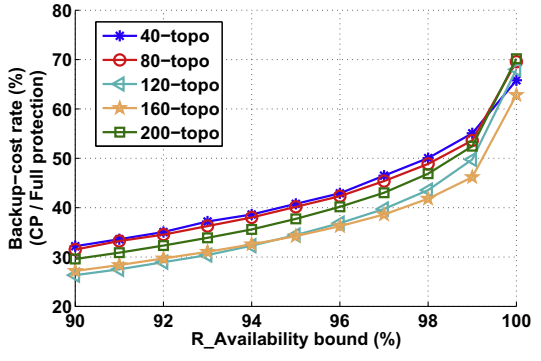
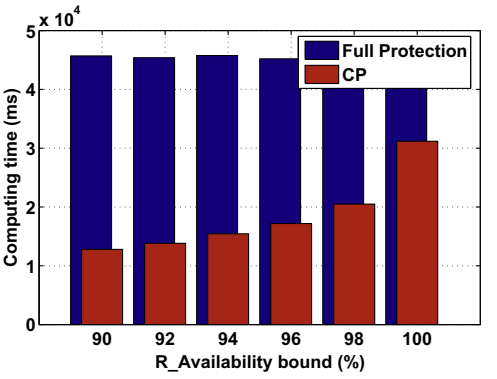
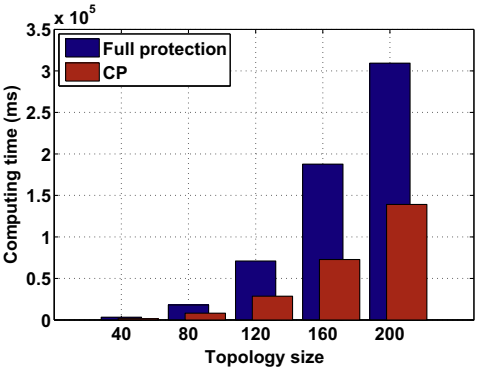


Fig. 6. Results on randomly generated topologies containing 40–200 nodes with different *R_Availability bound* ($C/MRate = 1$, single link failure).



(a) Topology size = 100.



(b) *R_Availability bound* = 97%.

Fig. 7. Time for CP algorithm and full protection scheme ($C/MRate = 1$, single link failure).

the computing time of the CP algorithm is much lower than that of full protection and is similar to the ratio of backup cost. That indicates the additional overhead the CP algorithm introduces is negligible compared with the backup path computing.

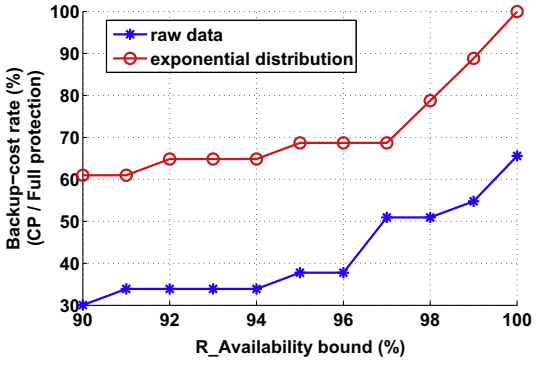
Fig. 8 shows the experiment results on CERNET2 topology with both raw failure data (real failure trace data) and the generated failure data described in 6.1. We can see that the CP algorithm still performs very well with CERNET2 topology and obtains significant cost gain with both raw and generated failure data. In Fig. 8, the CP algorithm's performance with raw failure data is even much better than that with generated failure data, indicating that the performance with generated failure data is a lower bound and the method we use to generate failure data is valid in this aspect.

6.2.2. Two link failure

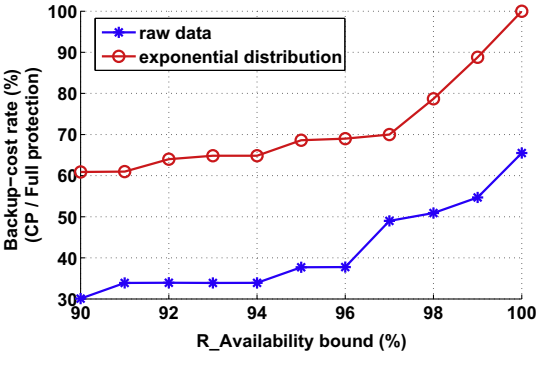
We also conduct similar simulation and experiments in two link failure situation, shown in Figs. 9–12. The CP algorithm shows similar characteristic in two link failure situation compared with that in single link failure:

- Cost gain of CP algorithm decreases with the increase of *R_Availability bound* (Fig. 9∼12).

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

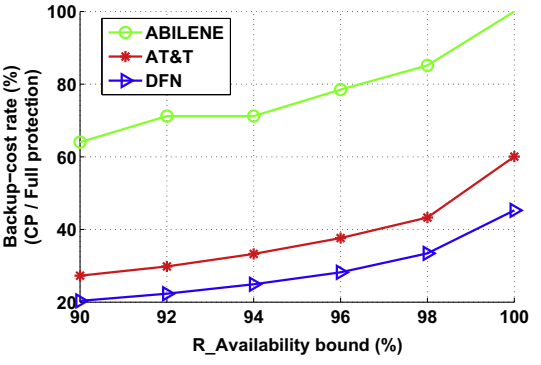


(a) $C/MRate = 1$.

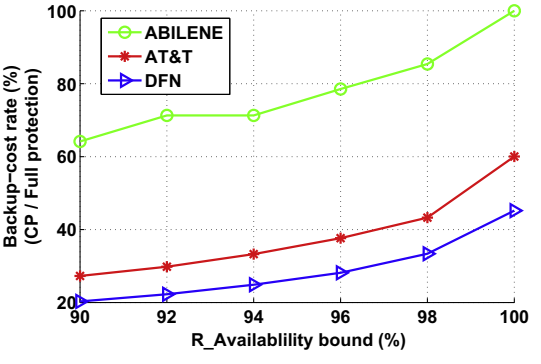


(b) $C/MRate = 2$.

Fig. 8. Results on CERNET2 (single link failure).



(a) $C/MRate = 1$



(b) $C/MRate = 2$

Fig. 9. Results on sub-real topologies and Abilene (two link failure).

- Different weights of computational cost and memory cost do not obviously affect the performance (Fig. 9).
- Additional computational cost that CP algorithm brings is small enough to be neglected, compared with that of computing backup paths (Fig. 11).
- In the experiments on CERNET2 with real and generated failure data, CP with real failure data performs even better than that with generated data in two link failures situation (Fig. 12).

In Fig. 10, we can also see that the performance is still stable with different topology size for two link failure situation, and the backup cost of CP algorithm is much lower than that of the full protection with different *R_Availability bound* in all the topologies, i.e. less than 35% when *R_Availability bound* = 90%.

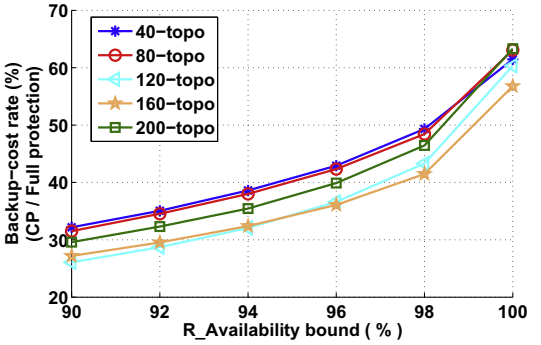
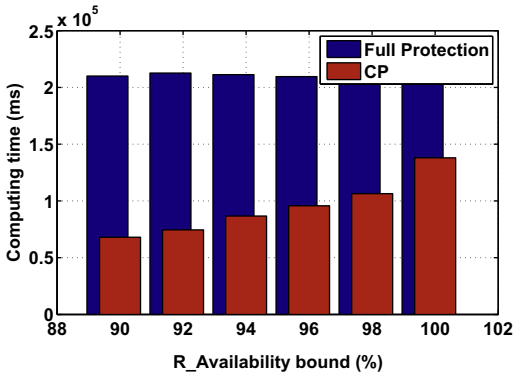


Fig. 10. Results on randomly generated topologies containing 40–200 nodes with different *R_Availability bound* ($C/MRate = 1$, two link failure).

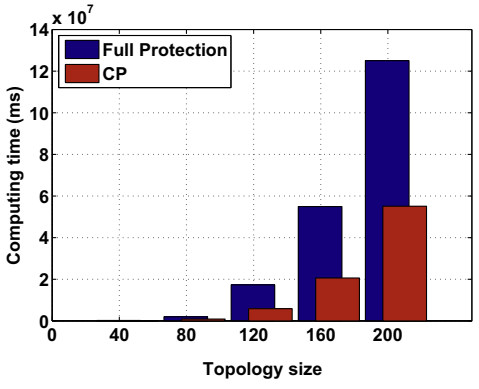
6.2.3. Multi-link failure comparison

To compare the performance of the CP algorithm under different multi-link failure situations, we conduct experiments on CERNET2 topology with real failure data in single link failure situation, two link failure situation and three link failure situation, respectively. Fig. 13 shows that, the CP algorithm has better performance in the failure situations with fewer simultaneous link failures on CERNET2 topology. For example, the cost of CP algorithm in single link failure situation, two link failure situation and three link failure situation with *R_Availability bound* = 90% and raw failure data, is about 30%, 40% and 60% of that of full protection, respectively.

In the CERNET2 topology there are some links which have no backup path even in the single link failure situation. That means the computational cost of the two and three simultaneous link failure situations can be saved

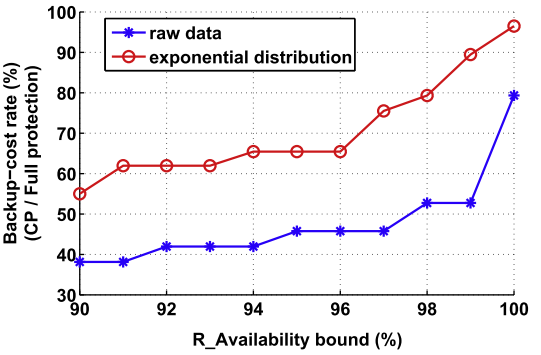


(a) Topology size = 40.

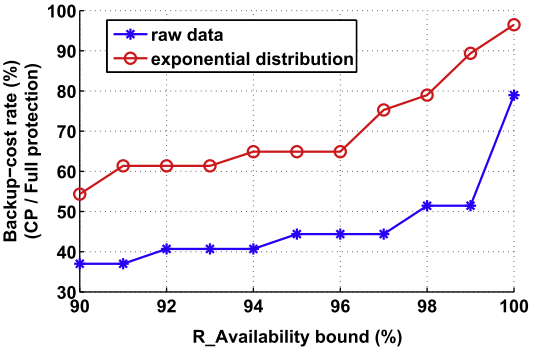


(b) *R_Availability bound* = 97%.

Fig. 11. Time for CP algorithm and full protection scheme ($C/MRate = 1$, two link failure).



(a) $C/MRate = 1$.



(b) $C/MRate = 2$.

Fig. 12. Results on CERNET2 (two link failure).

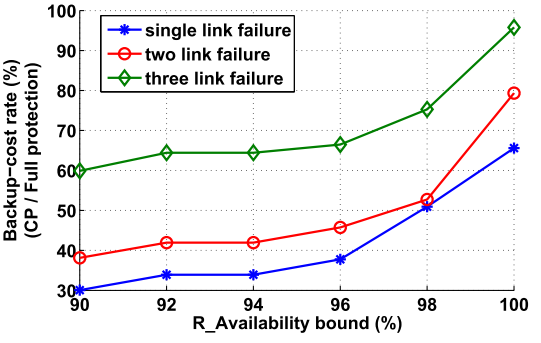


Fig. 13. Results in multi-link failure situations (CERNET2 topology, real failure data, $C/MRate = 1$).

for these links (see Algorithm 3 line 13), both in CP algorithm and the full protection scheme. Because the experiments in Fig. 13 are based on real failure data, the links which have the highest criticality are those which have the greatest impact on the network availability rather than those which have no backup path. Thus, the saved computational cost in CP algorithm is much less than that in full protection in the multi-link failure situations. It is more obvious in the situations with more simultaneous link failures. As a result, to achieve the same *R_Availability bound*, the ratio of the backup cost of CP algorithm to that of full protection goes higher (indicating the performance decreases) in the situations with more simultaneous link failures.

However, if comparing Fig. 8 with Fig. 12, we can see that the CP algorithm performs even sightly better with the generated failure data on CERNET2 topology in two link failure situation than in single link failure situation. That is because, though the generated failure distribution also confirms to heavy-tailed distribution, the corresponding relationship between link and failure rate is random rather than confirming to that with real failure data. Therefore, it is quite possible that the links with highest criticality are those which have no backup path, and CP algorithm may save cost even as much as the full protection does in multi-link failure situations. As a result, to achieve the same *R_Availability bound*, the ratio of the backup cost of CP algorithm to that of full protection in two link failure situation may be lower than that in single link failure situation with generated failure data. Besides, the performance with generated failure data in both of the situations may be lower than that with real failure data, which confirms to Fig. 8 and Fig. 12.

Besides, if comparing Fig. 6 with Fig. 10, we can see that the performance of CP algorithm in single link situation is almost equal to that in two link situation. That is because,

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

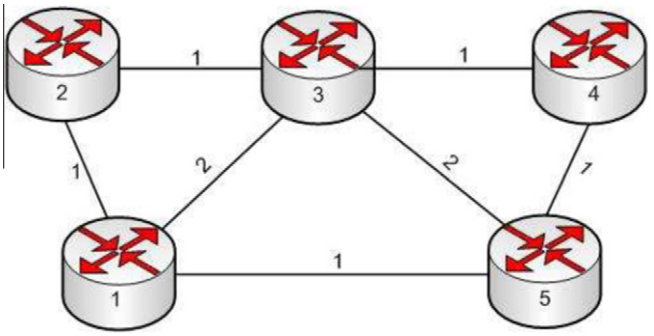


Fig. 14. Example topology with a Hamiltonian cycle.

in the topologies (with 40–200 nodes) generated by BRITE, there is no such link that has no backup path in single link failure situation. Thus, the backup cost ratio of CP algorithm to full protection in two and three link failure situations is not influenced.

6.3. Limitation of the study

We would like to comment on the limitation of our work. In our study, we apply the history information of the link failures to develop our algorithm as such that we know these data in advance. We admit that this is invalid. Our argument is that we have seen in many different situations (e.g., web caching [51]) the more unstable a link (or an object/file) was in the history, the more unstable this link will be in the future. As such using history data might be a reasonable approximation. We leave a more comprehensive autonomous self-learning on the network failures to our future work.

7. Discussion

Hamiltonian cycle protection [36] which has been fully discussed in 2.2, is a state-of-the-art protection method in optical networks. This method can provide better resource utilization than many other backup protection schemes. Since what we focus on in this paper is also how to reduce the backup cost and increase the resource utilization of protection, in this section, we compare the Hamiltonian cycle protection (HCP) solution and the critical protection (CP) solution we propose in this paper in a qualitative way.

In the mathematical field of graph theory, a Hamiltonian path is a path that visits each vertex exactly once, and a Hamiltonian cycle (also called Hamiltonian circuit) is a Hamiltonian path that is a cycle [37]. For example, there is a Hamiltonian path $(v_1 - v_2 - v_3 - v_4 - v_5 - v_6)$ in Fig. 1 and a Hamiltonian cycle $(v_1 - v_2 - v_3 - v_4 - v_5 - v_1)$ in Fig. 14. Here, in order to compare with CP that we propose in this paper, we adapt the HCP method to the IP-layer networks by giving up resource reservation.

Then, we can compare HCP and CP in the following aspects:

- *Working scope.* HCP can only working in single link failure situation, while CP can protect both single link failures and multi-link failures. This is because, a second link failure may break the Hamiltonian cycle which is the critical part of protection.
- *Computational cost.* HCP computes the Hamiltonian cycle only once. That is enough to protect all the single link failure in the network. However, not all the topologies contain Hamiltonian cycles and the problems that determining whether such paths and cycles exist in graphs is NP-complete [38]. Thus, the computation cost depends on the adopted off-line heuristic algorithm. In CP, each node needs to do computation each time the topology changes, but only needs to compute SPT for the protected links. Thus, the more links are protected, the more computational cost is required on each node. Therefore, if CP protects more links, HCP will take more advantages at the computation cost, while if CP protects less links, CP will take more advantages.
- *Memory cost.* Each link which is on the backup path of a protected link needs to add an additional item. That is the memory cost we discuss here. In HCP, every node is on the backup path of each on-cycle link, and averagely half of the nodes are on the backup path of a straddling links.[6] Thus, in HCP, each node need to add an item for each on-cycle link, and averagely half of the nodes need to add an item for each straddling link. In Fig. 14, if on-cycle link e_{23} fails, its backup path is $(v_2 - v_1 - v_5 - v_4 - v_3)$ including all the nodes, thus each node needs to store an backup item for the protection of e_{23}. If straddling link e_{13} fails, its backup path is $(v_1 - v_2 - v_3)$ or $(v_1 - v_5 - v_4 - v_3)$, thus no matter which path is chosen as backup path, nodes that are not on it do not need to store addition items.
  In CP, only the nodes on the backup paths of the selectively protected links need to add additional items. In Fig. 14, we assume only e_{23}, e_{45} and e_{35} are selected to be protected, whose the backup path are $(v_2 - v_1 - v_3)$, $(v_4 - v_3 - v_5)$ and $(v_3 - v_4 - v_5)$ respectively. Then, the total number of items increased in the whole network in CP is $3 + 3 + 3 = \mathbf{9}$. In HCP, the number of nodes on a on-cycle links' backup path is 5 (e.g. backup path $(v_2 - v_1 - v_5 - v_4 - v_3)$ of e_{23}), and the number of on-cycle links is 5, while the number of nodes on a strad-

[6] On-cycle links refer to the links that are on the Hamiltonian cycle, while straddling link are the ones that are not on the Hamiltonian cycle.

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

dling links' backup path is 3 (e.g. backup path $(v_1 - v_2 - v_3)$ of e_{13}), and the number of straddling links is 2. Thus, the total number of items increased in the whole network in HCP will be $5 \times 5 + 3 \times 2 = \mathbf{31}$, which is much bigger than that in CP.

Therefore, memory cost for link protection in HCP is much bigger than that in CP, because HCP greatly increases the average length of backup paths, especially in large networks with huge number of nodes.

– *Length of backup paths.* As we've discussed in the Memory cost part, HCP can significantly extend the backup path length. As a result, the delay of packet delivery will increase in HCP, besides the increase of memory cost.

We find HCP can gain advantages in computational cost in some cases, but pays much more memory cost than CP, and greatly extends the average backup path length, especially when the network contains a huge number of nodes. Besides, HCP is confined in the single link failure situation, while CP can cover both of single link failure and multi-link failure situations.

8. Conclusion and future work

In this paper, we proposed a critical protection scheme for handling failures in link state routing. By carefully studying on the failure characteristics of CERNET2, we observed that a substantial number of failures on the Internet are caused by a small set of links; this conforms to previous measurement studies on Sprint. Consequently, we proposed a critical protection scheme in which only a subset of links is protected for both single link failure situation and multi-link failure situation. We formulated an optimization problem in which the cost should be reduced and the network performance should be guaranteed. The challenge for the algorithm design was that the system cost highly depends on the computational overhead for the backup paths. Therefore, the link selection algorithm should be fast itself. We thus proposed a novel Critical Protection Algorithm where we identified critical links to be selected early. We evaluated our scheme comprehensively with topologies generated from BRITE and other real world topologies in both single link failure situation and multi-link failure situation. We further evaluated our scheme by using the traces collected from CERNET2.

We have shown that, our critical protection is cost-efficient. In the future, we would like to conduct a larger scale failure analysis. We believe a more precise prediction on failure behavior and identification of vulnerable links can further improve the performance. We also believe that our critical protection scheme can be used as a building block or a starting phase for more sophisticated protection schemes.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61170211 and 61073166, the National Basic Research Program of China (973 Program) under Grant No. 2009CB320505 and 2012CB315803, the National High-Tech Research and Development Program of China (863 Program) under Grant No. 2011AA01A101, the National Science & Technology Pillar Program of China under Grant No. 2011BAH19B01, and Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) under Grant No. 20110002110056. Dan Wang's work is also in part supported by HK PolyU Grant A-PB0R.

References

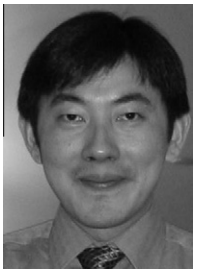
[1] M. Hou, D. Wang, M. Xu, J. Yang, Selective protection: a cost-efficient backup scheme for link state routing, in: IEEE ICDCS, Montreal, Canada, 2009.
[2] P. Pérez, J. Macías, J.J. Ruiz, N. García, Effect of packet loss in video quality of experience, Bell Labs Tech. J. 16 (1) (2011) 91–104.
[3] W. Jiang, H. Schulzrinne, Comparison and optimization of packet loss repair methods on VoIP perceived quality under bursty loss, in: the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video, ACM NOSSDAV, 2002, pp. 73–81.
[4] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. nee Chuah, Y. Ganjali, C. Diot, Characterization of failures in an operational ip backbone network, IEEE/ACM Trans. Netw. 16 (4) (2008) 749–762.
[5] D. Pei, M. Azuma, D. Massey, L. Zhang, BGP-RCN: improving BGP convergence through root cause notification, Comput. Networks 48 (2) (2005) 175–194.
[6] M. Kodialam, T.V. Lakshman, Dynamic routing of bandwidth guaranteed tunnels with restoration, in: IEEE INFOCOM, Tel-Aviv, Israel, 2000.
[7] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, J. Wilcox, Information-centric networking: seeing the forest for the trees, in: ACM HotNets, Cambridge, MA, 2011.
[8] K. Levchenko, G.M. Voelker, R. Paturi, S. Savage, XL: An efficient network routing algorithm, in: SIGCOMM, Seattle, WA, 2008.
[9] A. Li, P. Francois, X. Yang, On improving the efficiency and manageability of NotVia, in: ACM CoNEXT, New York, NY, 2007.
[10] S. Nelakuditi, S. Lee, Y. Yu, Z. li Zhang, C. nee Chuah, Fast local rerouting for handling transient link failures, IEEE/ACM Trans. Netw. 15 (2) (2007) 359–372.
[11] S. Bryant, S. Previdi, M. Shand, IP Fast Reroute Using Not-Via Addresses, Internet Draft (June 2012).
[12] China Education and Research NETwork 2 (CERNET2). <http://www.cernet2.edu.cn/>.
[13] G. Iannaccone, C. nee Chuah, R. Mortier, S. Bhattacharyya, C. Diot, Analysis of link failures in an IP backbone, in: ACM IMC, Marseille, France, 2002.
[14] BRITE: Boston university Representative Internet Topology gEnerator. <http://www.cs.bu.edu/brite/>.
[15] C.-L. Li, S.T. McCormick, D. Simchi-Levi, The complexity of finding two disjoint paths with min–max objective function, Discrete Appl. Math. 26 (1) (1990) 105–115.
[16] J.W. Suurballe, Disjoint paths in a network, Networks 4 (2) (1974) 125–145.
[17] D. Xu, Y. Chen, Y. Xiong, C. Qiao, X. He, On the complexity of and algorithms for finding the shortest path with a disjoint counterpart, IEEE/ACM Trans. Netw. 14 (2006) 147–158.
[18] M. Shand, S. Bryant, IP Fast Reroute Framework, IETF RFC 5714 (January 2010).
[19] M. Gjoka, V. Ram, X. Yang, Evaluation of IP fast reroute proposals, in: IEEE COMSWARE, Bangalore, India, 2007.
[20] A. Atlas, A. Zinin, Basic Specification for IP Fast Reroute: Loop-Free Alternates, IETF RFC 5286 (September 2008).
[21] A. Kvalbein, A.F. Hansen, T. Cicic, S. Gjessing, O. Lysne, Fast IP network recovery using multiple routing configurations, in: IEEE INFOCOM, Bacelona, Spain, 2006.
[22] N. Kushman, S. Kandula, D. Katabi, B.M. Maggs, R-BGP: staying connected in a connected world, in: USENIX NSDI, Cambridge, MA, 2007.
[23] B. Rajagopalan, J. Luciani, D. Awduche, IP over optical networks a framework, in: IETF RFC 3717 (March 2004).
[24] L. Guo, L. Li, A novel survivable routing algorithm with partial shared-risk link groups (SRLGs)-disjoint protection based on differentiated reliability constraints in WDM optical mesh networks, J. Lightwave Technol. 25 (6) (2007) 1410–1415.

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006

[25] D. Xu, Y. Xiong, C. Qiao, G. Li, Trap avoidance and protection schemes in networks with shared risk link groups, J. Lightwave Technol. 21 (11) (2003) 2683–2693.
[26] China Education and Research NETwork (CERNET). <http://www.edu.cn/>.
[27] P. Sebos, J. Yates, G. Hjalmtysson, A. Greenberg, Auto-discovery of Shared Risk Link Groups, in: OFC, Anaheim, CA, USA, 2001.
[28] P. Sebos, J. Yates, A. Greenberg, D. Rubenstein, Effectiveness of shared risk link group auto-discovery in optical networks, in: OFC, Anaheim, CA, USA, 2002.
[29] L. Guo, J. Cao, H. Yu, L. Li, Path-based routing provisioning with mixed shared protection in WDM mesh networks, J. Lightwave Technol. 24 (3) (2006) 1129–1141.
[30] S. Ramamurthy, L. Sahasrabuddhe, B. Mukherjee, Survivable WDM mesh networks, J. Lightwave Technol. 21 (4) (2003) 870–883.
[31] H. Zang, C. Ou, B. Mukherjee, Path-protection routing and wavelength assignment (RWA) in WDM mesh networks under duct-layer constraints, IEEE/ACM Trans. Netw. 11 (2) (2003) 248–258.
[32] H. Choi, S. Subramaniam, H. Choi, Loopback methods for doublelink failure recovery in optical networks, IEEE/ACM Trans. Netw. 12 (6) (2004) 1119–1130.
[33] P. Ho, J. Tapolcai, T. Cinkler, Segment shared protection in mesh communications networks with bandwidth guaranteed tunnels, IEEE/ACM Trans. Netw. 12 (6) (2004) 1105–1118.
[34] P.H. Ho, H.T. Mouftah, Shared protection in WDM mesh networks, IEEE Commun. Mag. 42 (1) (2004) 70–76.
[35] W.D. Gover, Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking, Prentice HALL PTR, 2003.
[36] L. Guo, X. Wang, X. Wei, T. Yang, W. Hou, T. Wu, A new link-based Hamiltonian cycle protection in survivable WDM optical networks, in: AICT, Athens, Greece, 2008.
[37] L. Pósa, Hamiltonian circuits in random graphs, Discrete Math. 14 (1976) 359–364.
[38] C. Iwamoto, G. Toussaint, Finding Hamiltonian circuits in arrangements of jordan curves is NP-complete, Inform. Process. Lett. 52 (1994) 183–189.
[39] L. Guo, X. Wang, C. Yu, Enhanced Hamiltonian cycle protection algorithm in survivable networks, in: ICSCN, Chennai, India, 2008.
[40] L. Guo, X. Wang, W. Hou, Y. Li, C. Wang, A new differentiated Hamiltonian cycle protection algorithm in survivable WDM mesh networks, in: International Conference on Signal Processing Systems, Singapore, 2009.
[41] L. Guo, X. Wang, W. Hou, Enhanced multicast Hamiltonian cycle protection in WDM optical networks, in: ICCSNA, Hong Kong, 2010.
[42] L. Guo, X. Wang, J. Cao, W. Hou, J. Wu, Y. Li, Local and global Hamiltonian cycle protection algorithm based on abstracted virtual topology in fault-tolerant multi-domain optical networks, IEEE Trans. Commun. 58 (3) (2010) 851–859.
[43] P. Francois, Achieving sub-second IGP convergence in large IP networks, SIGCOMM Comput. Commun. Rev 35 (3) (2005) 2005.
[44] G. Iannaccone, C. nee Chuah, S. Bhattacharyya, C. Diot, Feasibility of IP restoration in a Tier-1 backbone, IEEE Network 18 (2004) 13–19.
[45] J. Vasseur, M. Pickavet, P. Demeester, Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS, Morgan Kaufmann Series in Networking, Morgan Kaufmann, 2004.
[46] Personal communication with Dr. Wenlong Chen, Bitway Networks. <http://www.bit-way.com/default.aspx>.
[47] Abilene. <http://itservices.stanford.edu/service/network/internet2/abilene>.
[48] DFN (German research network).
[49] AT&T. http://www.corp.att.com/globalnetworking/.
[50] O. Heckmann, M. Piringer, J. Schmitt, R. Steinmetz, Generating realistic ISP-level network topologies, IEEE Commun. Lett. 7 (7) (2003) 335–337.
[51] K. Krishnamurthy, J. Rexford, Web Protocols and Practices, Addison-Wesley Professional, 2001.

Mingwei Xu received the B.Sc degree in 1994 and Ph.D. degree in 1998 both from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. Now he is a professor in Tsinghua University. His research interests include computer network architecture, Internet routing and high-speed router architecture.

Meijia Hou received the B.Sc and M.Sc degrees in computer science from Northeastern University, Shenyang, China, in 2004 and 2007, respectively. She is current a Ph.D. candidate in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. Her current research interests include intra-domain and inter-domain routing.



Dan Wang received the B.Sc degree from Peking University, Beijing, China, in 2000, the M.Sc degree from Case Western Reserve University, Cleveland, Ohio, USA, in 2004, and the Ph.D. degree from Simon Fraser University, Burnaby, BC, Canada, in 2007; all in computer science. He is currently an assistant professor at the Department of Computing, The Hong Kong Polytechnic University. His research interests include wireless sensor networks, Internet routing, and peer-to-peer networks.

Jiahai Yang received his M.Sc. and Ph.D. degrees both in computer science from Tsinghua University, Beijing, China, in 1992 and 2003 respectively. He had been with the Department of Computer Science and Technology, Tsinghua University since 1992. In August of 1999, he joined the Network Research Center of Tsinghua University as an associate professor. He was a visiting scholar of George Mason University from October 1999 to May 2000, and a Member of Technical Staff (MTS) of Bell Labs, Lucent Technologies from June 2000 to June 2001. He has published more than 90 articles in refereed international conferences and journals, and two books on network management and Internet measurement. Now he is a professor of the Network Research Center of Tsinghua University. Jiahai's research interest includes computer network architecture and its protocols, computer network applications, Internet routing technology, network management, network measurement, etc.

Please cite this article in press as: M. Xu et al., An efficient critical protection scheme for intra-domain routing using link characteristics, Comput. Netw. (2012), http://dx.doi.org/10.1016/j.comnet.2012.09.006