MPCT: Minimum Protection Cost Tree for IP Fast Reroute using Tunnel

Mingwei Xu, Qing Li, Lingtao Pan and Qi Li
Dept. of Comp. Sci. & Tech., Tsinghua Univ. Tsinghua National Laboratory for Information Science and Technology
{xmw, liqing, plt, liqi}@csnet1.cs.tsinghua.edu.cn

***Abstract*—The demand for faster failure-recovery in the Internet has led to the development of several IP Fast Reroute (IPFRR) schemes, which are all too computationally expensive or unsatisfactory in protection coverage. In this paper, we propose Minimum Protection Cost Tree (MPCT) for IPFRR using Tunnel. By constructing an MPCT for each hypothetical failed neighbor, MPCT finds the protection paths for all the affected destinations. First, MPCT provides 100% single-node protection coverage by direct forwarding (DF) and re-protection. Second, the computational complexity of MPCT is less than one full shortest path first (SPF) calculation. Third, by simulation with the data from CERNET, Rocketfuel and Brite, we show that even without DF and re-protection, MPCT can provide more than 99.7% protection coverage for single-node failures. We believe that our scheme MPCT moves a big step towards practical deployment.**

I. Introduction

The current Internet routing protocols take on the order of a few hundred milliseconds or even more to re-converge after failure [1]. However, recent popularity of online realtime applications, *e.g.,* Voice over IP, has led to stringent demands on the transmission delay of the Internet [2]. Internet Service Providers (ISPs) hence have strong incentives to improve the network survivability.

IP fast reroute (IPFRR) [3–6] is one of the most significant directions for network survivability. In IPFRR, the failure-adjacent nodes (protection source nodes) pre-compute backup paths, which can be used to protect failure-affected packets immediately upon the detection of failures. For practical deployment, an IPFRR scheme should 1) provide high protection coverage and 2) introduce slight overhead in the current routing protocol. Thus far, there are four significant intradomain IPFRR schemes of routing protection: loop free alternates (LFA) [3], ESCAP [4], NotVia [5] and Tunnel [6]. Each of these schemes is either too computationally expensive or unsatisfactory in protection coverage.

In this paper, we propose minimum protection cost tree (MPCT) for Tunnel to compute the optimal protection paths. The development of the algorithm MPCT can be divided into two main steps. First, we introduce our naive solution of Tunnel-AT based on the incremental SPF (iSPF) [7]. We prove Tunnel-AT can provide 100% protection coverage for single-node failures by direct forwarding and re-protection [8]. However, the corresponding protection paths by Tunnel-AT might use unnecessary direct forwarding (not well supported by ISPs) or re-protection. Thus, in order to avoid the unnecessary direct forwarding and re-protection in Tunnel-AT as far as possible, we provide MPCT to compute the optimal tunnel end point for each destination. In MPCT, the protection path without direct forwarding is prior to that with direct forwarding; the protection path without re-protection is prior to that with re-protection. Except for the length of the protection path, the algorithm of MPCT reserves all the other advantages of Tunnel-AT, especially the computational efficiency: less than one SPF calculation for the full protection.

The research is supported by the National Natural Science Foundation of China (No. 61073166), the National Basic Research Program of China (973 Program) under Grant 2009CB320502, the National High-Tech Research and Development Program of China (863 Program) under Grants 2009AA01Z251, the National Science & Technology Pillar Program of China under Grant 2008BAH37B03.

978-1-4577-0103-0/11/$26.00

II. Minimum Protection Cost Tree

In each iteration of the algorithm of incremental shortest path first (iSPF): 1) the node with the smallest change of dist is selected; 2) a subtree, instead of only one node, is added to the new SPT. The process of iSPF can be viewed as attaching subtrees back one by one until the new SPT forms. In [8], we proposed Tunnel-AT according to iSPF. Tunnel-AT can achieve 100% protection coverage for single-node failures by direct direct forwarding (DF) and re-protection [9]. However, DF and re-protection are currently not well accepted by ISPs. Thus, we propose the scheme of Minimum Protection Cost Tree (MPCT) to avoid DF and re-protection as far as possible.

First we introduce the a few significant definitions mentioned in Tunnel-AT. An *attaching tree* (*ATTree* for short) is a subtree attached to the new SPT in an iteration of iSPF. Connected *ATTree*s in the new SPT form a *super ATTree*. The *incoming node* for any failure-affected destination is the root of the corresponding *super ATTree*. The *attaching node* for any failure-affected destination is the father of the corresponding *incoming node* in the new SPT.

A. Minimum Protection Cost Tree

To make Tunnel practical for commercial deployment, the top-level goal is providing 100% single-node protection coverage; the second-level goal is avoiding direct forwarding as far as possible; the third-level goal is avoiding re-protection; the fourth-level goal is optimizing the protection path length.s

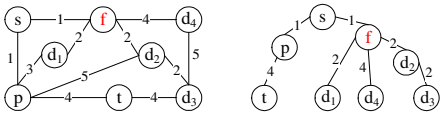
In our scheme, the Minimum Protection Cost Tree (MPCT) instead of SPT is used for protection path computation. The concepts of *ATTree*, *super ATTree*, *incoming node* and *attaching node* still apply. Different from iSPF, in each iteration,

the *ATTree* with the minimum Ω value will be attached. $\Omega = V_{DF} + V_{repro} + V_{path}$. V_{DF} is associated with direct forwarding, V_{repro} is associated with re-protection, and V_{path} is associated with protection path length. Now we provide the detailed definitions of V_{DF}, V_{repro} and V_{path}.

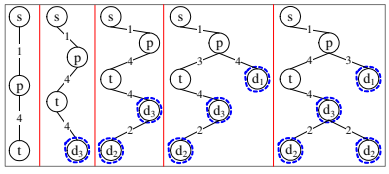
Let $\mathcal{D}$ ($diameter$) be an constant that is larger than the longest path in the network. Let q be the potential *incoming node*. Let p be the potential *attaching node*. Let r be the root of the *ATTree*.

- V_{DF}. If s is not on the route from p to r (direct forwarding is not needed), $V_{DF} = 0$; or else, $V_{DF} = 2 \times \mathcal{D}$.
- V_{repro}. 1) $V_{DF} = 0$. If f is on the route from p to d (re-protection is required), $V_{repro} = C_d$; or else, $V_{repro} = 0$. 2) $V_{DF} \neq 0$. If f is on the route from q to d (re-protection is required), $V_{repro} = C_d$; or else, $V_{repro} = 0$.
- V_{path}. $V_{path} = \widehat{dist}(p,d) - dist(s,p) - dist(s,d)$, where $\widehat{dist}(p,d)$ is the dist from p to d in the new tree to be formed. The smaller V_{path} is, the less possible that direct forwarding is required.

We show an example of MPCT construction with the topology in Fig. 1(a) and the SPT in Fig. 1(b) as the input. s and f are the protection source node and the failed node, respectively. As shown in Fig. 1(c), in the first iteration, there are three potential *ATTrees*: $\{d_1\}$, $\{d_2, d_3\}$ and $\{d_3\}$. The Ω values for them are respectively -1, $1+2\times\mathcal{D}$ and -6. Thus, the *ATTree* $\{d_3\}$ is attached below t in the first iteration. Then, $\{d_2, d_3\|$ is removed; $\{d_2\}$ and $\{d_4\}$ become new potential *ATTrees* (d_3 is the attaching node). The Ω values for $\{d_2\}$ and $\{d_4\}$ are respectively -2 and -1. Thus, the *ATTree* $\{d_2\}$ is attached below d_1 in the second iteration. Then $\{d_1\}$ and $\{d_4\}$ are attached in the following steps. The final MPCT shows the protection paths for all the failure-affected destinations. p is the selected tunnel end point for d_1; t is the selected tunnel end point for d_3, d_2 and d_4. In this example, direct forwarding and re-protection are completely avoided.



(a) A topology (b) The SPT



(c) The steps of MPCT

Fig. 1. (a) An example of topology. (b) The corresponding s-rooted SPT of (a). (c) The steps of MPCT after f fails.

MPCT Scheme. When f fails, s reroutes the packet to the corresponding *attaching node* by Tunnel, or to the corresponding *incoming node* by Tunnel with Direct Forwarding if f is on the route from the *attaching node* to the destination.

Theorem 1. *MPCT can provide 100% single-node protection by direct forwarding and re-protection.*

Proof: See [9] due to the limited space. ■

B. Method for Ω Calculation

The challenge of designing an efficient algorithm is the Ω calculation for each *ATTree* in MPCT. Now we introduce the novel method for s to calculate the Ω value for any *ATTree* of MPCT.

Before diving into the details of calculation, we first emphasize the fact that s can obtain $dist(s,x)$ and $\widehat{dist}(s,x)$ without any extra calculation. Let p be the *attaching node* and q be the *incoming node*.

First, $V_{path} = \widehat{dist}(p,d) - dist(s,p) - dist(s,d)$. $dist(s,p)$ and $dist(s,d)$ are known values. $\widehat{dist}(p,d)$ can be calculated according to $\widehat{dist}(p,d) = \widehat{dist}(s,d) - dist(s,p)$. Thus,

$$V_{path} = \widehat{dist}(s,d) - dist(s,p) - dist(s,p) - dist(s,d)$$

Second, if $V_{path} < 0$, $V_{DF} = 0$; or else $V_{DF} = 2 \times \mathcal{D}$. If $V_{path} < 0$, $dist(p,d) \leq \widehat{dist}(p,d) < dist(s,p) + dist(s,d)$, thus the packet encapsulated to p will not loop back to s after decapsulation and no direct forwarding is needed.

Now we discuss the calculation of V_{repro}. Let x be the end node of the protection action ($x = p$ if $V_{DF} = 0$; or else, $x = q$). If f is not on the route from x to d, no re-protection will be triggered, thus $V_{repro} = 0$. We use the following formula to calculate V_{repro}:

$$V_{repro} = \begin{cases} 0, & \widehat{dist}(x,d) < dist(x,f) + dist(f,d); \\ C_d, & \text{else.} \end{cases}$$

C. Construction of MPCT and Protection Paths

Table I summarizes the notations used in the algorithm. $\mathcal{T}$ is the given SPT rooted at s. All the operations in the algorithm are based on this SPT. ω is used to replace the above Ω.

TABLE I
SYMBOLS IN TUNNEL-AT ROUTING ALGORITHM

$\mathcal{T}$	The shortest path tree rooted at s
$I(x), I(N)$	In edges of node x or of a set of nodes N
$O(x), O(N)$	Out edges of node x or of a set of nodes N
$S(e)$	The source node of edge e
$E(e)$	The end node of edge e
Q	A priority queue
$\{d, (p, dist, \omega)\}$	An item in Q: d is a destination, p is its potential parent, $dist$ is a potential distance, ω is a potential Ω.
$ATTree(r)$	The ATTree rooted at r in the MPCT.
D	The set of all affected nodes.

Algorithm 1 shows the first stage of Tunnel-AT routing algorithm. Given the shortest path tree $\mathcal{T}$ rooted at s and a failed node f, this algorithm constructs the MPCT and finds incoming node for each f-affected node.

Algorithm 2 shows the second stage of our algorithm, which computes the protection route for each f-affected destination based on the MPCT from Algorithm 1.

Theorem 2. *The complexity of MPCT algorithm is less than one full SPF.*

Proof: See [9] due to the limited space. ■

Algorithm 1 First stage: find the incoming node. ($\mathcal{T}$ is the SPT rooted at s and f is the failed node.)

Set all nodes in D as *floating*
for all $e \in I(D)$ **do** ▷ Initialize the Priority Queue
  if $S(e).state \neq floating$ **then**
    $dist \leftarrow S(e).dist + lenght(e)$
    Set ω according to our method
    enqueue$(Q, \{E(e), (S(e), dist, \omega)\})$
  end if
end for
while $Q \neq \emptyset$ **do**
  $\{d, (p, dist, \omega)\} \leftarrow$ extractMin(Q)
  Move $ATTree(d)$ under p
  for all $x \in ATTree(d)$ **do**
    Set $x.innode$ and $x.nhop$ accordingly
    $x.dist \leftarrow p.dist + \widehat{dist}(p, x)$
    $x.state \leftarrow attached$
    if $x \in Q$, remove x from Q
  end for
  for all $e \in O(ATTree(d))$ **do**
    if $E(e).state = floating$ **then**
      $dist \leftarrow S(e).dist + lenght(e)$
      Set the ω according to our method
      enqueue$(Q, \{E(e), (S(e), dist, \omega)\})$
    end if
  end for
end while

Algorithm 2 Compute the backup routes

for all $d \in D$ **do**
  $DF \leftarrow d.innode$
  $TEP \leftarrow DF.parent$
  if $V_{path} \geq 0$ **then** ▷ DF is necessary
    Set the protection route as $< TEP, DF >$
  else ▷ Tunnel without DF is enough
    Set the protection route as $< TEP, null >$
  end if
end for

III. Simulation

We evaluate our scheme MPCT by comprehensive simulations. We obtain six backbone intradomain topologies with inferred weights from Rocketfuel Project [10]. We extract the biggest biconnected components from those topologies, since non-biconnected components can not be protected and will add noise data to the simulation results.

1) Direct Forwarding Ratio: As shown in Fig. 2, the DF ratios of MPCT are no more than 0.3% in all the six Rocketfuel topologies. This demonstrate that even without direct forwarding, MPCT can provide more than 99.7% protection coverage for single-node failures.

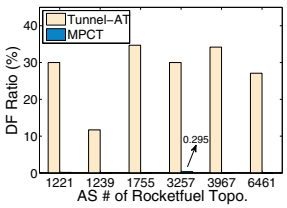


Fig. 2. DF ratios (Tunnel-AT and MPCT) in Rocketfuel topologies.

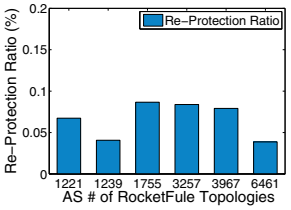


Fig. 3. Re-protection ratios of MPCT in the Rocketfuel topologies.

2) Re-protection Ratio: We now study the re-protection ratio of MPCT. Re-protection ratio is the ratio of the protection paths with re-protection to all the protection paths. As shown in Fig. 3, the re-protection ratios of the Rocketfuel topologies are all below 0.1%, which is almost ignorable.

3) Protection Path Stretch: As the length of protection path has the lowest priority in MPCT, protection path stretch is inevitable. We next study the protection path stretch in Rocketfuel topologies. As shown in Fig. 4, for single-node failures, the protection path stretches are from 10% to 20%, which is at the same level as NotVia (discussed in the section of related work). This is acceptable considering the achievement of full protection coverage, avoiding DF and avoiding re-protection. Besides, as previously mentioned, 70% of the failures are single-link failures. In this case, the protection path stretch would be alleviated.

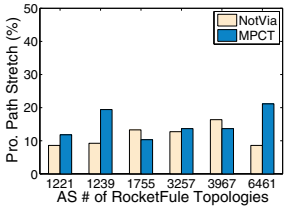


Fig. 4. Protection path stretch in the Rocketfuel topologies.

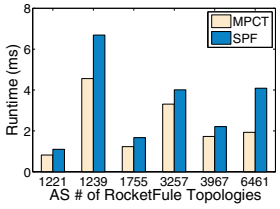


Fig. 5. Runtime of of MPCT and Shortest Path First (SPF) algorithm.

4) Runtime Evaluation: We next study the runtime of MPCT. We run our program on an Intel Core Duo CPU of 2.00GHz with RAM 2.0GB. As shown in Fig. 5, the runtime of MPCT is about 3 ms, which is below the runtime of shortest path first algorithm. This means that MPCT will not become a new computational bottleneck in OSPF protocol.

References

[1] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving Sub-Second IGP Convergence in Large IP Networks," *SIGCOMM CCR*, vol. 35, no. 3, pp. 35–44, 2005.

[2] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica, "Achieving Convergence-free Routing Using Failure-carrying Packets," in *Proc. SIGCOMM*, Kyoto, Japan, Aug. 2007.

[3] A. Alia, Z. Alex, T. Raveendra, C. Gagan, M. Christian, I. Brent, and F. Don, "Basic Specification for IP Fast-Reroute: Loop-free Alternates," IETF RFC 5286, Sep. 2008.

[4] K. Xi and H. J. Chao, "IP Fast Rerouting for Single-Link/Node Failure Recovery," in *Proc. IEEE Broadnets*, Raleigh, USA, Sep. 2007.

[5] M. Shand, S. Bryant, and S. Previdi, "IP Fast Reroute Using Not-via Addresses," IETF Draft, draft-ietf-rtgwg-ipfrr-notvia-addresses-05, Mar. 2010.

[6] S. Bryant, C. Filsfils, S. Previdi, and M. Shands, "IP Fast Reroute Using Tunnels," IETF Draft, draft-bryant-ipfrr-tunnels-03, Sep. 2007.

[7] P. Narváez, K.-Y. Siu, and H.-Y. Tzeng, "New dynamic SPT algorithm based on a ball-and-string model," *IEEE/ACM TON*, vol. 9, no. 6, pp. 706–718, 2001.

[8] L. Pan, M. Xu, Q. Li, and D. Jen, "Lightweight IP Fast Reroute with Tunnel-AT," in *Proc. IWQoS*, Beijing, P.R. China, Jun. 2011.

[9] M. Xu, Q. Li, L. Pan, and Q. Li, "MPCT: Minimum Protection Cost Tree for IP Fast Reroute using Tunnel," Dept. of Comp Sci. & Tech., Tsinghua University, Tech. Rep., 2011.

[10] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring Link Weights using End-to-End Measurements," in *Proc. IMW*, Marseille, France, Nov. 2002.