Optical Switching and Networking 8 (2011) 68–78

Contents lists available at ScienceDirect

Optical Switching and Networking

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

Multi-domain restoration with crankback in IP/MPLS networks

F. Xu [a], M. Peng [b], M. Esmaeili [a], N. Ghani [a,*], A. Rayes [c]

[a] University of New Mexico, United States
[b] Wuhan University, China
[c] Cisco Systems, United States

ARTICLE INFO

Article history:
Received 15 March 2010
Received in revised form 11 June 2010
Accepted 1 July 2010
Available online 14 July 2010

Keywords:
Network survivability
Multi-domain survivability
Restoration
Crankback signaling

ABSTRACT

Multi-domain network survivability is a key problem area and crankback signaling offers a very viable alternative for post-fault restoration. However, although some initial multi-domain crankback studies have been done, most have not considered post-fault recovery. Along these lines, this paper proposes a novel solution framework for joint intra/inter-domain crankback restoration in realistic MPLS/GMPLS network settings. Namely, dynamic link failure and intra-domain link-state routing information is coupled with the available inter-domain path/distance vector routing state to improve the recovery process. Mechanisms are also introduced to limit crankback overheads and delays. The performance of the proposed solution is then analyzed using simulation.

1. Introduction

Network survivability in IP-based *multi-protocol label switching* (MPLS) and optical *generalized MPLS* (GMPLS) networks is a very well-studied problem area. However, even though a wide range of pre-fault protection and post-fault restoration schemes have been proposed, most have assumed complete "network-wide" topology and resource visibility in the provisioning process. In general, this assumption is only valid in single "domain" settings, e.g., such as those controlled by a centralized provisioning entity and/or running distributed link-state routing protocols [1,2]. However, as user application demands continue to grow, there is a pressing need to extend service recovery across larger geographic ranges spanning multiple domains or *autonomous systems* (AS). In such settings, it is generally very difficult to have full "global" visibility across domains and inter-domain links, i.e., owing to obvious scalability and confidentiality limitations. Hence commensurate multi-domain recovery schemes must be designed to operate in a distributed, decentralized manner.

Now various survivability schemes have been proposed for handling inter-domain failures in IP and optical networks, nearly all of which focus on "pre-provisioned" protection [3–7]. (Note that most *intra-domain* failures are usually handled using existing domain-level recovery schemes.) For example, some designers have leveraged earlier SONET/SDH designs to implement "localized" dual/multi-homing interconnection [4] between border nodes in optical *dense wavelength division multiplexing* (DWDM) networks. Alternatively, others have developed more "globalized" protection schemes for multi-domain MPLS/GMPLS networks using sequential or parallel working/protection path computation [5–7]. In particular, some of these solutions have proposed hierarchical routing algorithms to compress and disseminate the inter-domain survivability states, i.e., path, link diversity [6,7]. This aggregated information is then used to compute and expand diverse paths across domains. However, even though these latter schemes deliver good blocking reduction, they pose notable concerns. Foremost, associated routing overheads are quite high owing to the larger dimensionality of the survivability-related state [6]. Moreover, many of these solutions will be difficult to realize in operational networks as carriers will prefer existing "inter-domain" distance/path

* Corresponding author. Tel.: +1 505 277 1475.
E-mail address: ghanin@yahoo.com (N. Ghani).

1573-4277/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.osn.2010.07.001

vector protocols, e.g., *border gateway protocol* (BGP) variants. Now these protocols only provide next-hop domain and endpoint reachability state and do not support any link-state updates for *quality-of-service* (QoS) or survivability support [1]. Finally, most protection schemes have been primarily designed to handle single-fault (link) recovery, and hence may not be very effective against multiple failures.

In light of the above, there is a pressing need to develop alternate *restoration* schemes for multi-domain survivability. These solutions basically rely upon active post-fault crankback signaling [8] to search for new routes in reduced visibility settings. However, few studies have been done in this overall area. For example, most recent "per-domain" crankback schemes have only addressed *traffic engineering* (TE) setup for working routes [9–12]. Moreover, many of these crankback solutions pursue rather cumbersome "exhaustive" search strategies, yielding high signaling overheads. In response, the authors here have studied various improved multi-domain crankback strategies for optical DWDM [13,14], and IP/MPLS [15] networks. However, all of these contributions only treat working-mode operation. Hence there is significant scope to apply crankback strategies in more complex post-fault restoration settings. This is the focus herein.

Along these lines, this paper proposes a novel multi-domain crankback scheme for restoration in MPLS/GMPLS networks using the standard *resource reservation* (RSVP-TE) protocol [8]. Specifically, two levels of crankback are defined – intra- and inter-domain – and these are applied at the intermediate and end-to-end path levels. Furthermore, the solution addresses realistic settings where nodes have full internal domain visibility via link-state routing, e.g., *open shortest path first* (OSPF-TE), but limited "next-hop" inter-domain visibility, e.g., as per inter-area or inter-AS routing protocols such as hierarchical OSPF or BGP. Overall, this paper is organized as follows. Section 2 presents a survey of the latest work on multi-domain provisioning and survivability, including standards and research. Next, Section 3 details the enhanced intra/inter-domain crankback provisioning/restoration solution. Detailed performance analysis is then conducted in Section 4 and conclusions and future directions presented in Section 5.

2. Background

Multi-domain networking is a relatively well-studied topic area and a range of protocol standards have been evolved over the years, i.e., at the IP/MPLS and optical DWDM layers. In general, many of these standards also provide requisite capabilities for network survivability. For example, many IP routing protocols support varying degrees of inter-domain state exchange, e.g., next-hop/path vector exchange in *exterior gateway protocols* (EGP) and hierarchical link-state exchange in two-level OSPF-TE. Inter-domain routing is also supported by the *Optical Internetworking Forum* (OIF) as part of its *network-to-network interface* (NNI) standard [1]. Furthermore, the recent IETF *path computation element* (PCE) [3] framework has also formalized a new framework for multi-domain TE and survivability route computation. Specifically, this solution introduces domain computational entities to decouple path computation from setup signaling. At the inter-domain level, these PCE entities can interact in a distributed manner to resolve end-to-end routes using a specialized PCE-to-PCE protocol. Overall, the PCE framework supports two computation strategies to handle varying levels of "global" state, i.e., *per domain* and *PCE-based* [3,10]. The former computes paths in a "domain-to-domain" manner and is most germane for limited inter-domain visibility. Meanwhile the latter relies upon the head-end PCE to compute a *partial* or *loose* route to the destination ("skeleton path") and is better suited for increased inter-domain visibility. However, since blocking can occur at signaling setup, crankback extensions have also been defined for RSVP-TE to support re-tries on alternate routes [8]. Namely, several multi-domain crankback strategies have been outlined (local, intermediate, and source), but detailed algorithms are left to vendor implementation.

Meanwhile on the research side, a wide range of multi-domain networking studies have been done, see [1] and related references. Now with regards to distributed *multi-domain survivability* in particular, various protection strategies have been studied. For example, [4] proposes optical-layer domain interconnection strategies, e.g., dual, multi-homed, to protect working and protection paths traversing the same domain sequence. These solutions leverage legacy SONET/SDH and are quite robust to localized link failures. However, many of these "domain-to-domain" schemes are quite inefficient (costly) and highly susceptible to multiple failures at domain boundaries. Hence more advanced *distributed* multi-domain protection strategies have also been proposed for improved "domain diversity" between working/protection routes. For example [5] tables *sequential* and *parallel* strategies for working/protection route computation in MPLS networks. In particular, the sequential scheme first computes end-to-end working routes and then uses the returned paths to compute diverse protection routes. However, these schemes are shown to be less optimal (higher blocking) and also more susceptible to "trap" topology problems between domains.

Meanwhile, alternate parallel strategies implement more complex joint, i.e., *concurrent*, path pair computation. However, these schemes require added state to ensure non-overlapping routes across common domains. A means of achieving this is to use hierarchical routing to extract and propagate critical state information between domains. Namely, various graph topology abstraction schemes have been developed to condense domain resource and survivability (diversity) state. This information is then flooded to build "abstracted" global views for use in working/protection path pair computation. For example, [6] applies Surballe's path pair algorithm to provision dedicated protection recovery. Meanwhile [7] proposes novel shared inter-domain path and overlapped segment protection schemes using full mesh and virtual edge abstractions. Results here show good blocking reduction for several topologies. Nevertheless, many of these "survivability-aware" abstraction schemes generate

significant levels of compressed state, e.g., whole sets of arrays [6]. As such they have high inter-domain overheads and will complicate deployment in real-world settings where carriers will prefer existing distance/path vector routing protocols.

In light of the above concerns, some researchers have started to study alternate *signaling crankback* strategies for multi-domain provisioning. The objective here is to limit the dependence on global state and instead have individual domains compute concatenated end-to-end paths in a "per-domain" manner. However, for the most part, these schemes have only been applied for regular working connection setup. For example, [8] defines a basic *"per-domain"* (PD) crankback scheme which probes egress domain nodes for traversal routes and upon failure, notifies upstream border nodes. Results show higher blocking rates and crankback delays, particularly when compared to PCE-based strategies utilizing pre-determined inter-domain routes. Meanwhile, [9,10] detail a MPLS-based *compute while switching* (CWS) scheme. Here, a similar crankback procedure to [8] is used to first compute an initial inter-domain route. If this search is successful, transmission is started and *simultaneous* crankback is initiated to search for a shorter route. If a shorter route is found, data switchover is performed. Results here show good setup success as the scheme essentially mimics an exhaustive search. However, the CWS scheme entails very high signaling overheads/delays (not analyzed) and requires non-standard extensions to RSVP-TE attributes. In addition, [11] addresses end-to-end path delays in multi-domain settings and presents two next-hop domain selection strategies. The first selects the next-hop as the "nearest" egress border node whereas the other uses pre-computed inter-domain *round-trip time* (RTT) measurements. Here, the latter heuristic is shown to yield slightly higher carried loads and less crankback attempts, although it requires adoption of a specialized coordinates system [11].

To address the above concerns with existing multi-domain crankback strategies, the authors have recently proposed some further innovations of their own. For example, [13,14] outline joint intra/inter-domain crankback algorithms for achieving lightpath *routing and wavelength assignment* (RWA) in multi-domain DWDM networks. Namely, these solutions define added counter mechanisms to limit crankback overheads and also outline novel intelligent next-hop domain selection schemes based upon hop counts. Meanwhile, [15] applies the above framework to multi-domain IP/MPLS networks and also introduces new features to track crankback history state between multiple independent crankback attempts. Although these studies represent some good contributions in the field, they are mostly focused on regular *working-mode* operation. As a result, the effort herein expands upon this base and develops new provisions for post-fault restoration. In particular, the solution re-uses some of the key aspects of the above strategies (intra/inter-domain counters, next-hop domain selection) but introduces further innovations based upon *intermediate* and *end-to-end* crankback recovery.

3. Enhanced crankback restoration

A novel multi-domain crankback provisioning/recovery solution framework is now presented. Although the focus is on IP/MPLS networks, this solution can readily be tailored for optical DWDM RWA settings as well. The approach assumes realistic settings with full intra-domain link-state routing and more scalable path/distance vector routing at the inter-domain level. Each domain is also assumed to have a PCE entity with full access to interior and exterior routing databases. This entity plays a key role in crankback recovery as it resolves next-hop domains (egress border gateways). Meanwhile, all setup signaling is done using new crankback extensions for RSVP-TE [8]. Leveraging these standards and the work in [13–15], three key saliencies are introduced for post-fault restoration, i.e., (1) dual intra/inter-domain crankback counters to limit signaling complexity/delay, (2) intermediate and end-to-end crankback recovery, and (3) intelligent per-domain selection. Carefully note that the overall presentation here is done is the context of MPLS "bandwidth provisioning" networks. However, since the proposed framework here is quite generic, it can easily be modified to support lightpath restoration in multi-domain DWDM networks (with and without conversion). Indeed, this is a focus of some ongoing efforts. Further details are now presented.

3.1. Multi-domain crankback: working mode

Before presenting the crankback scheme, the requisite notation is introduced. A multi-domain network is comprised of D domains, with the ith domain having n^i nodes and b^i border/gateway nodes, $1 \leq i \leq D$. This network is modeled as a set of domain sub-graphs, $\boldsymbol{G}^i(\boldsymbol{V}^i, \boldsymbol{L}^i)$, $1 \leq i \leq D$, where $\boldsymbol{V}^i = \{v^i{}_1, v^i{}_2, \ldots\}$ is the set of domain nodes and $\boldsymbol{L}^i = \{l^{ii}{}_{jk}\}$ is the set of *intra-domain* links in domain i ($1 \leq i \leq D$, $1 \leq j, k \leq n^i$), i.e., $l^{ii}{}_{jk}$ is the link from $v^i{}_j$ to $v^i{}_k$ with available capacity $c^{ii}{}_{jk}$. The inter-domain link connecting border node $v^i{}_k$ in domain i with border node $v^j{}_m$ in domain j is further denoted as $l^{ij}{}_{km}$ with available capacity $c^{ij}{}_{km}$, $1 \leq i, j \leq D$, $1 \leq k \leq b^i$, $1 \leq m \leq b^j$. All connectivity is assumed to be bi-directional, i.e., interconnected nodes $v^i{}_k$ and v^j_m have two links between them, $l^{ij}{}_{km}$ and $l^{ji}{}_{mk}$. All nodes also maintain a list of traversing connections, $\underline{\boldsymbol{A}}^i_j$ for node $v^i{}_j$, where each entry in $\underline{\boldsymbol{A}}^i_j$ is a route vector. Finally, relevant RSVP-TE message fields support a path route vector, $\underline{\boldsymbol{R}}$, and other fields for crankback support [8]. The latter include an exclude link vector, $\underline{\boldsymbol{X}}$, to track failed links and crankback history and dual intra/inter-domain crankback counters, h_1 and h_2 (usages detailed shortly). Note that RSVP-TE extensions in [8] only define a single counter field but bit masking can be used to generate the above two "sub-counters".

First consider regular per-domain provisioning for *non-crankback* operation, i.e., no resource failures, using "per-domain" route computation. Here a source fielding a request for x units of capacity to a destination in another domain first queries its PCE to determine an egress link to the next-hop domain, e.g., via PCE-to-PCE protocol. The PCE then determines the next-hop domain to the

destination domain (detailed in Section 3.3) and returns a domain egress border node/link to this domain. Note that this information also contains the *ingress* border node in the downstream domain. Upon receiving the PCE response, the source uses its local OSPF-TE database to compute an *explicit route* (ER) to the specified egress border node. This step searches the *k*-shortest path sequences over the *intra-domain* feasible links ($c^{ii}{}_{jk} \geq x$) and chooses the one with the lowest "load-balancing" *cost*, i.e., individual link costs inversely proportional to free link capacity, i.e., $1/c^{ij}{}_{km}$. This method is used as it outperforms minimum hop-count routing [6,11].

Granted that an ER path is found above, it is inserted in the path route vector, $\underline{\boldsymbol{R}}$, and RSVP-TE *PATH* messaging is then initiated (along the expanded route) to the ingress border node in the next-hop domain. Here, each intermediate node checks for available bandwidth resources on its outbound link and pending availability, propagates the message downstream. The above procedure is repeated at all next-hop domain border nodes until the destination domain. When the *PATH* message finally arrives at the destination domain, the border node (or PCE) expands the ER to the destination. Upon receiving a fully expanded *PATH* message, the destination initiates upstream reservation by sending a *RESV* message. Here, all intermediate nodes also store the final path routes, i.e., list of traversing connection routes, for use in post-fault recovery.

Now consider the case of resource unavailability during *PATH* message processing, i.e., due to insufficient bandwidth resources on a route link. Here, crankback signaling is invoked to help re-compute an alternate route. Now current extensions to RSVP-TE [8] have outlined various alternatives for crankback operation, and two types are chosen herein, i.e., *intra-domain* (local) and *inter-domain* (intermediate). Namely, the proposed scheme uses two crankback counters that are carried in the RSVP-TE *PATH* messages, i.e., h_1 and h_2. These counters are initialized to pre-specified values at the start of the *working* path signaling phase, i.e., H_1 and H_2, respectively, and then decremented during crankback to avoid searching excessively long resource consuming paths. Overall, these values effectively bound the number of intra- and inter-domain crankback attempts to $H_1 \cdot H_2$.

Using the above counters, two key operations are defined for working-mode crankback operation, i.e., *notification* and *re-computation*, as originally outlined in [15]. The former refers to the (upstream) signaling procedures executed upon link resource failure at an intermediate node, whereas the latter refers to the re-routing procedure to select a new route. Now in general, resource signaling (*PATH* processing) failures can occur at *three* different nodes, i.e., domain ingress border nodes, domain egress border nodes, and interior nodes. However, in the proposed scheme, only the former performs *re-computation* whereas the latter two simply perform crankback *notification*. These steps are presented in Figs. 1 and 2 and now detailed.

Crankback notification: Leveraging from [15], upstream notification is done when there is insufficient bandwidth on an intra-domain link (i.e., at an intra-domain node) or an inter-domain link (i.e., at an egress border node) on an already-expanded ER, i.e., *PATH* signaling failure. The overall algorithm here is shown in Fig. 1. Namely, the *PATH* message is terminated and its appropriate fields updated and copied to an upstream *PATH_ERR* message to the domain's ingress border node. Specifically, the intra-domain counter h_1 is decremented and the failed (resource-deficient) link is noted. If blocking occurs in the source domain, the *PATH_ERR* message is sent back to the source node.

Crankback re-computation: Meanwhile as per [15], path re-routing is done by ingress border nodes receiving a *PATH_ERR*, Fig. 2. Note that, for special case of a source domain (i.e., non-ingress border node), the receiving source node relays the *PATH_ERR* to its PCE. Here, two types of crankback re-computations can be done. At the "intra-domain" level, if the intra-domain h_1 counter has not expired in the received *PATH_ERR* message, another next-hop domain/egress border node is selected by the ingress border node for ER expansion. Now the exact sequence of next-hop domains tried is pre-computed to try successively longer inter-domain routes (via multi-entry distance vector table, Section 3.2). This scheme makes use of crankback history to avoid any failed intra/inter-domain links. Foremost, all failed inter-domain links in $\underline{\boldsymbol{X}}$ that egress from the domain are removed from consideration, i.e., only consider "non-failed" next-hop domain egress links. Additionally, all intra-domain links listed in the exclude link vector $\underline{\boldsymbol{X}}$ are also precluded from *local* ER computation. The route vector $\underline{\boldsymbol{R}}$ is also searched to make sure that an upstream domain is not traversed twice, i.e., no "domain-level" loops. Regardless, it still may not be possible to initiate/establish a domain-traversing route for various reasons, i.e., h_1 counter expired, LR expansion failure to selected egress node, or all egress border links in exclude link vector $\underline{\boldsymbol{X}}$, etc. In these cases, the ingress border node must initiate a more globalized "inter-domain crankback" response via a *PATH_ERR* message to the ingress node in the *upstream* domain in the *PATH* route vector $\underline{\boldsymbol{R}}$ (or source node if upstream domain is source domain). In addition, the intra-domain counter is also reset to $h_1 = H_1$. Also, to improve history tracking, the ingress border node also inserts its own *ingress link* in the exclude route vector of the *PATH_ERR*. Note that "inter-domain crankback" is only initiated if the h_2 counter is non-zero, otherwise the request is failed (*PATH_ERR* to source, Fig. 2). Carefully note re-computation only tracks and uses crankback history for the individual connection being signaled, i.e., stored in *PATH* or *PATH_ERR* message. The broader tracking of history information between *multiple* independent requests is not done here in order to simplify complexity at ingress border nodes (although this is done in [15]).

Overall, sample working-mode crankback is shown further in Fig. 3 for counter values $H_1 = 3/H_2 = 3$. The case of intra-domain crankback is first seen in domain 3, where ingress border node $v_1{}^3$ makes $H_1 = 3$ unsuccessful ER expansion attempts to its border nodes. Crankback is then initiated to the upstream node $v_2{}^3$ which resolves a new route via domain 5, Fig. 3.

```
if (insufficient resources on outbound link at PATH message)
    Decrement intra-domain counter h1, extract route vector R and exclude link vector X
    from PATH

    Add failed outbound link to exclude route vector X

    Remove all nodes in route vector R up to ingress border node, i.e., prune failed intra-
    domain segment

    Generate PATH_ERR, copy h1, R, X fields and send to upstream ingress border node
```

Fig. 1. Crankback notification (local or egress border node).

```
/* First attempt intra-domain re-routing */
if (h1 not expired)
    Select next-hop domain/egress link using multi-entry distance vector table s.t.
        next-hop domain is not in R and egress link is not in X

    if (next hop egress node found)
        Make copy of local network graph (via IGP database), prune all local failed
        links listed in X, compute new ER to egress border node

        if (LR expansion successful)
            Initiate PATH signaling to new egress node
            intra_domain_crankback_done=1;

/* Next attempt inter-domain re-routing */
if (!intra_domain_crankback_done & h2 not expired)
    Decrement inter-domain counter h2, extract route vector R and exclude route
    vector X from PATH

    Add ingress inter-domain link to exclude link vector X

    Remove all nodes in route vector R up to previous domain's ingress border node

    Copy h2, R, X fields, reset h1=H1, generate PATH_ERR and send to previous
    domain's ingress border node

else if (h2 has expired)
    Copy h1, h2, R, X fields,generate PATH_ERR, send to source
```

Fig. 2. Crankback re-computation (at domain ingress border node).

3.2. Multi-domain crankback: restoration mode

Now consider crankback restoration after a link failure, intra- or inter-domain. Here, it is assumed that both endpoints of a link can quickly discover the failure using rapid lower layer detection/localization mechanisms (out of scope herein). Using these notifications, the overall restoration procedures are initiated at the two endpoint nodes detecting the failure, as shown in Fig. 4. Namely, each node loops through its active connection list, $\underline{\boldsymbol{A}}_j^i$, and searches for any connections traversing the failed link. All such connections are removed from $\underline{\boldsymbol{A}}_j^i$ and appropriate restoration procedures initiated for each. Specifically, the endpoint nodes first send appropriate resource takedown messages along the failed upstream and/or downstream path segments. This is done by either sending downstream *PATH_TEAR* or upstream *RESV_TEAR* messages (depending upon which side of link the destination node is). Next, if the endpoint node detecting the failure is on the upstream side of the failed link, it also notifies the source node of the connection via a *PATH_ERR* message with the failed link noted in exclude route vector, $\underline{\boldsymbol{X}}$. This notification message also sets an appropriate restoration flag to indicate that this is a failed connection event and not a regular connection crankback.

Upon receiving the *PATH_ERR* failure notifications from downstream endpoint failure nodes, source nodes initiate (post-fault) crankback setup of new routes. Carefully note that since multiple crankback attempts can be initiated in close succession after a link failure, signaling race conditions can arise, particularly on inter-domain links. Therefore, each notified source must wait for a random back-off interval before initiating re-tries, typically averaging a few seconds. Again intra/inter-domain crankback counter limits are set to H_1 and H_2, respectively, and two different restoration schemes are proposed:

End-to-end (E2E): In this case the source node simply generates and sends a new *PATH* message request to the destination, i.e., cleared route vector $\underline{\boldsymbol{R}}$, crankback counters reset to $h_1 = H_1$ and $h_2 = H_2$, and failed link (from received *PATH_ERR*) copied to outgoing exclude route list $\underline{\boldsymbol{X}}$. This request is then processed as per regular setup procedures in Section 3.1.

Intermediate (IM): Here the source nodes try to preserve as much of the original path as possible, i.e., up to the domain in which the failure occurred. Namely, the failed connection route (in returning *PATH_ERR*) is copied up to domain ingress border node of the failed domain. This partial route is then inserted into the route vector $\underline{\boldsymbol{R}}$ of a new *PATH* message with reset counter values ($h_1 = H_1$, $h_2 = H_2$) and failed link noted in $\underline{\boldsymbol{X}}$. Hence this setup message will basically track along the original connection route and then initiate crankback processing from the failed node/domain. Note that race

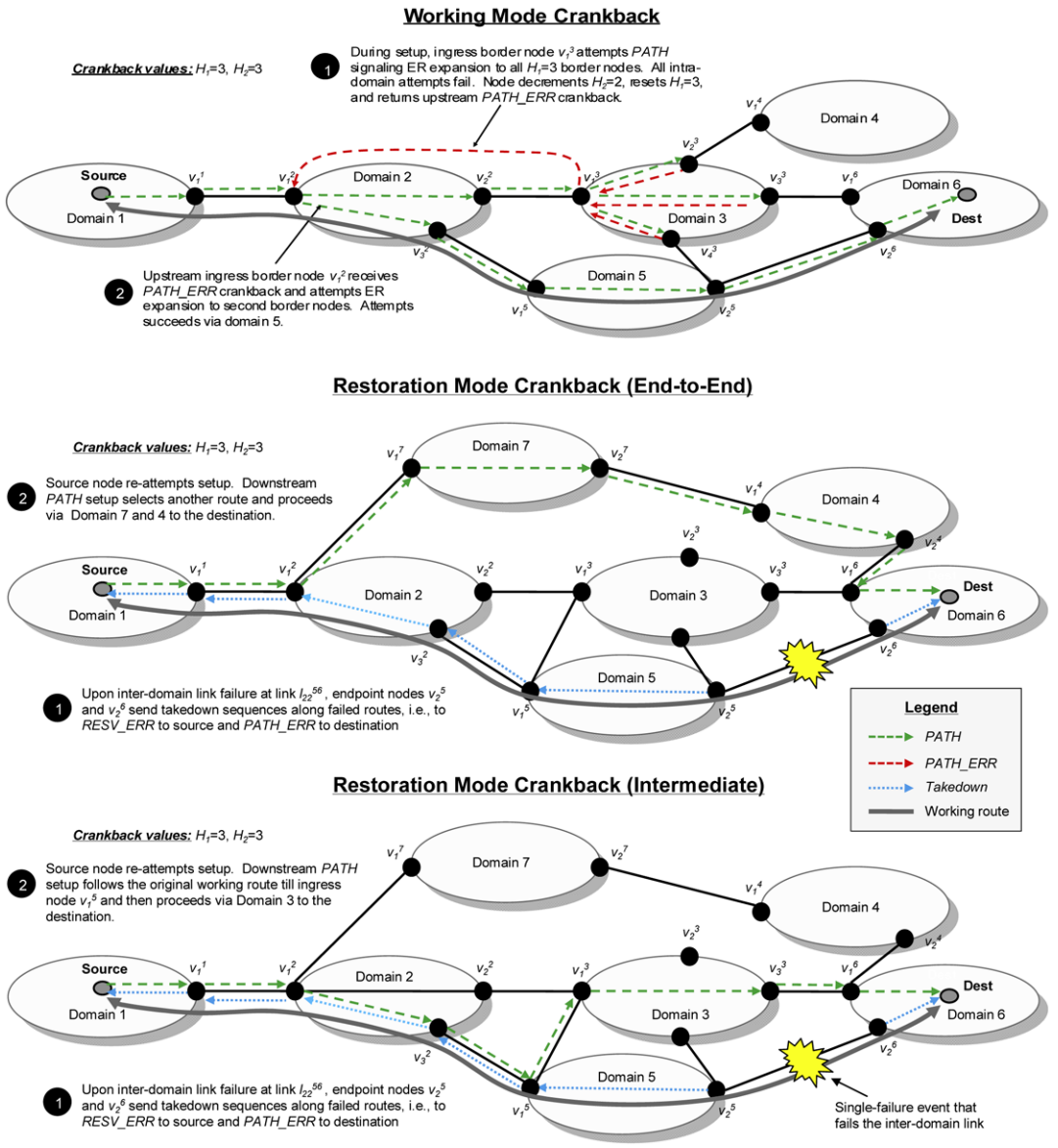

Fig. 3. Proposed intra/inter-domain crankback scheme for working and two restoration modes.

```
/* Process all traversing connections at node vij, which is connected with the failed link */
Loop through all traversing connections at node, Aij
    if (connection route uses failed link)
        Remove connection from list Aij
        if (source is not on the upstream side of failed link)
            Send PATH_TEAR along downstream segment
        else
            Send RESV_TEAR along upstream segment
            Generate PATH_ERR, set restoration flag, save route
            in R and failed link in X, send to source node
```

Fig. 4. Link failure notification (at a node with failed link).

conditions can occur if resources along the partial route are allocated between post-failure takedown (of original failed connection) and subsequent source-based crankback re-initiation. However, these conditions are rare at mid-light loads, and regardless, can be handled by the crankback process.

The case of post-fault restoration crankback is shown in Fig. 3 for $H_1 = 3/H_2 = 3$. Here, given a failure on link $l_{22}{}^{56}$ between domains 5 and 6, the endpoint nodes first issue takedown sequences to free resources along the failed connection paths. Next, the source node re-initiates crankback to setup new routes. Now for the case of E2E restoration, border node $v_1{}^2$ picks another "next-hop domain", i.e., domain 7, which ends up routing the restored path through domain 4. Conversely for the case of IM restoration, the recovery procedure follows part of the original working path up to ingress node $v_1{}^5$ in domain 5, i.e., prior domain of the failed link $l_{22}{}^{56}$. In turn, this routes the recovered path through domain 3, see Fig. 3.

3.3. Next-hop domain computation

A key saliency of the scheme is its use of existing (available) inter-domain routing state to improve the crankback search. Specifically, the scheme re-uses the approach in [15] to pre-compute *multi-entry* distance vector tables at all domain border nodes (or PCE entities), i.e., to list K next-hop domains/egress links to each destination domain. Namely, at domain i, the kth table entry to a destination domain j, $T^i(j, k)$, is computed as the egress inter-domain link (to the next-hop domain) on the kth shortest "domain-level" hop-count path to domain j ($1 \leq i, j \leq D$, $i \neq j$, $1 \leq k \leq K$). Clearly the number of entries to a destination will be upper bounded by the minimum of K and the maximum number of links egressing from the domain.

Now consider the actual computation of this table at a border node (or PCE) in domain i, the algorithm for which is summarized in Fig. 5. Here a "simple node" [2] view of the global topology is first derived, i.e., $\boldsymbol{H}(\boldsymbol{U}, \boldsymbol{E})$, where $\boldsymbol{U}$ is the set of domains $\{\boldsymbol{G}^i\}$ reduced to vertices and $\boldsymbol{E}$ is the set of inter-domain links $\{\boldsymbol{l}^{ij}{}_{km}\}$, $i \neq j$. Now at the inter-area level, this graph can be obtained from hierarchical OSPF-TE link-state databases whereas at the inter-AS level it can approximately be deduced from BGP path vector state (albeit not all inter-domain connectivity may be visible due to policy restrictions). An iterative shortest path scheme is then used to compute multiple routes to all destination domains over $\boldsymbol{H}(\boldsymbol{U}, \boldsymbol{E})$. Namely, the scheme basically loops over all destination domains $j \neq i$ (index j) and computes up to K next-hop egress links (index k) over a temporary copy of $\boldsymbol{H}(\boldsymbol{U}, \boldsymbol{E})$, i.e., $\boldsymbol{H}'(\boldsymbol{U}, \boldsymbol{E})$, Fig. 5. At the kth iteration, the scheme computes the shortest "domain-level" hop-count path to the destination domain using $\boldsymbol{H}'(\boldsymbol{U}, \boldsymbol{E})$, and if found, stores the egress link from the source domain in $T^i(j, k)$. This link is then pruned from $\boldsymbol{H}'(\boldsymbol{U}, \boldsymbol{E})$ and the procedure repeated to compute the next shortest "domain-level" hop-count path. The procedure is terminated if all K entries are filled and/or the vertex for domain i in $\boldsymbol{H}'(\boldsymbol{U}, \boldsymbol{E})$ becomes disconnected.

From the above, it is seen that next-hop domain selection during crankback re-computation (as detailed in Section 3.1) simply searches these K table entries, $T^i(j, k)$, to the destination domain. As such, this sequentially drives the search along fixed "domain-level" sequences of increasing length, but with provisions to prune "failed" entries (in $\underline{\boldsymbol{X}}$). By and large, these table entries will be relatively static during non-failure conditions, but must be re-computed in post-fault settings if there are any inter-domain topology changes. Finally, from a storage overhead perspective, this setup has $O(KD)$ requirements at each border node. As this amount increases linearly in the number of domains, it is generally acceptable for multi-domain settings.

```
Generate simple-node abstraction of global topology via
EGP database information, i.e., H(U, E)
/* At domain i, loop across all possible destination domains */
for j = 1 to D
  if (j ≠ i)
    Make temporary copy of graph H(U,E), i.e., H'(U,E)
    /* Compute up to K table entries */
    for k=1 to K
      Compute shortest-path from domain i to j in H'(U,E)
      if (shortest path route found)
        Save route line from domain i in k-th table entry T^i(j,k),
        i.e., link from domain i vertice in H'(U,E)
        Prune above-selected link from H'(U,E)
      if (domain i becomes disconnected)
        break k-loop
```

Fig. 5. Multi-entry distance vector table computation (at PCE).

4. Performance evaluation

The performance of the proposed multi-domain restoration schemes are tested using specially developed models in *OPNETModeler*^{TM}. Specifically, simulations are done using two multi-domain backbone topologies, including a modified NSFNET topology (with nodes replaced by domains) with 16 domains/25 bi-directional inter-domain links (i.e., 3.12 links per domain), Fig. 6, and a 10-domain topology with 24 bi-directional inter-domain links (i.e., 4.8 links per domain), Fig. 7. In these two networks, all intra- and inter-domain link rates are set to 10 Gbps and the domain size is varied from 7–10 nodes. Furthermore, all connection requests are generated between random nodes in random domains, and each run is averaged over 500,000 connections with mean holding times of 600 s (exponential). Request inter-arrival times are defined via random exponential distributions and varied as per load (no intra-domain requests). The actual connection request sizes are further varied uniformly between 200 Mbps–1 Gbps in increments of 200 Mbps, in order to model fractional Ethernet demands. Meanwhile, link failures are limited to inter-domain links with exponential mean inter-arrival times of 1200 s. Finally, $K = 5$ next-hop domain entries are computed in the distance vector table, although the number searched is limited by the H_1 or H_2 values chosen. Simulation results of both topologies are now presented and evaluated.

Carefully note that *pre-provisioned* protection schemes have also been studied for multi-domain recovery, see [1,6,7] (as noted in Section 2). However, these strategies are not compared against the proposed restoration scheme for several key reasons. Foremost, protection and (crankback) restoration strategies are generally applied to different user scenarios. Namely, the former are more suited for stringent higher-priced services and mandate

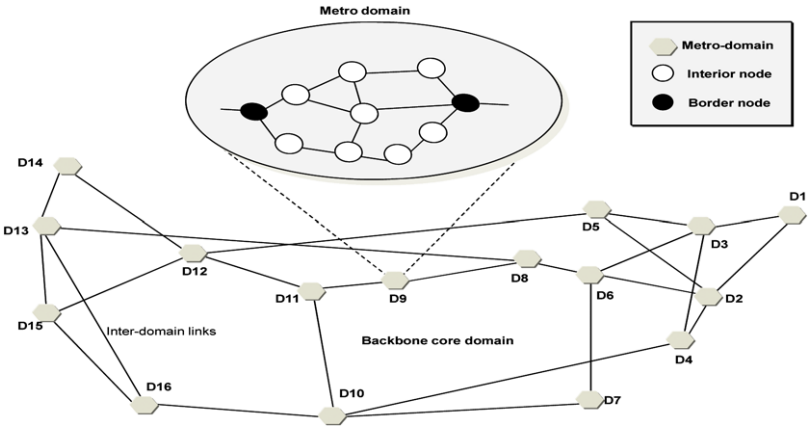

Fig. 6. 16-domain modified NSFNET test topology.

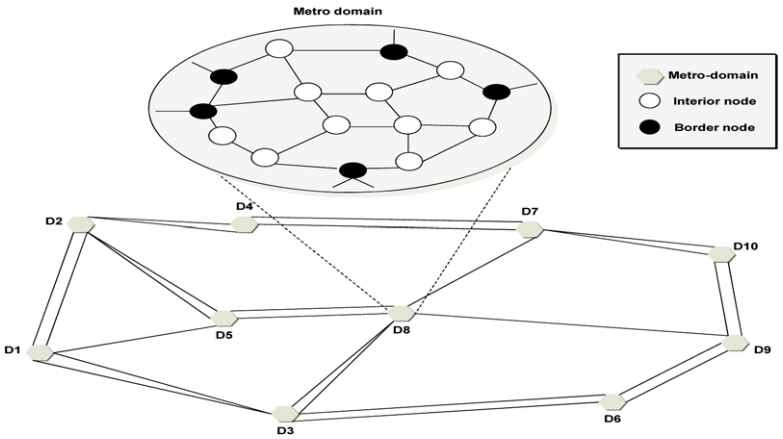

Fig. 7. 10-domain test topology.

separate protection routes for high availability during failure conditions. Conversely, the latter are more geared towards mid-lower priority services that can suffice without "100%" recovery guarantees. Moreover, given the fact that protection schemes consume more resources (i.e., by routing extra protection connections), they generally operate at different network loading/blocking regimes. Hence for a given user request arrival rate, the output blocking probabilities for multi-domain protection schemes will be notably higher than those for restoration schemes. In turn, this makes it difficult to normalize loads and properly compare these two methodologies for a given network topology. As such, many carrier will prefer to treat these schemes as complimentary – and not competitive – and will choose to apply restoration as a "last-gasp" measure to recover from complex failures affect both working and protection routes.

Post-fault restoration performance is first evaluated by measuring the restoration success rates of failed connections for various intra/inter-domain restoration counters, i.e., intra-domain only ($H_1 = 6/H_2 = 0$), inter-domain only ($H_1 = 0/H_2 = 6$), and joint ($H_1 = 3/H_2 = 2$). Carefully note that the H_1 and H_2 counter limits here are selected to limit the maximum number

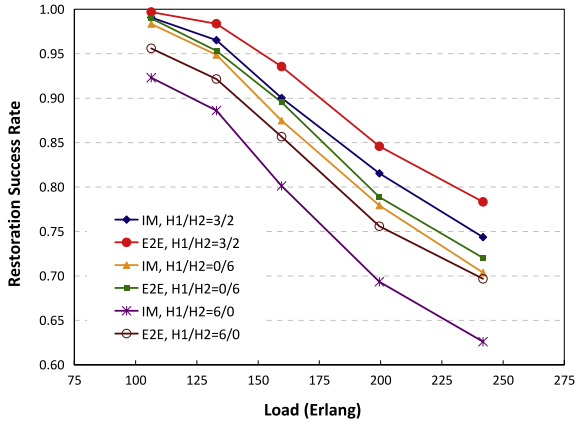

Fig. 8. Post-fault restoration success rate (NSFNET).

of crankback attempts to the same value in all cases, i.e., 6, in order to form a basis for comparison. However, the actual number of intra-domain crankback attempts will be upper bounded by the number of border nodes in a domain and the number of inter-domain crankback attempts may not reach H_2 if the source domain is reached.

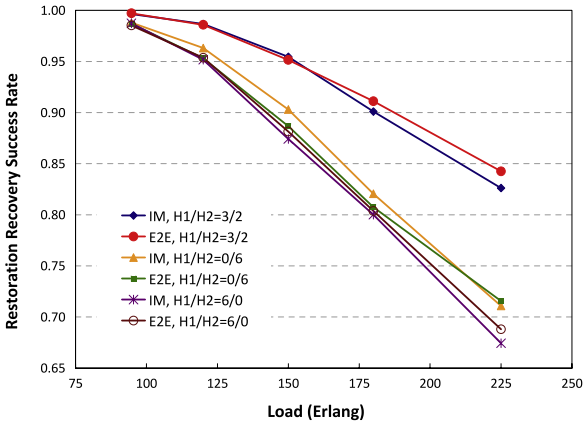

Fig. 9. Post-fault restoration success rate (10-domain).

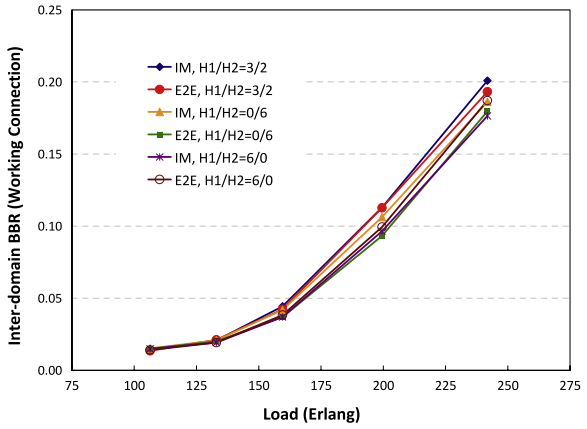

Fig. 10. Working connection BBR results (NSFNET).

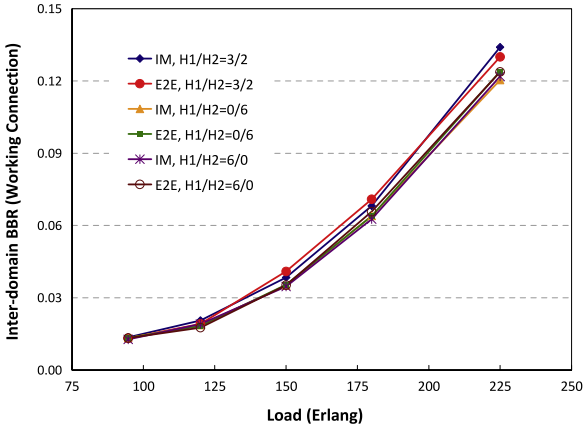

Fig. 11. Working connection BBR results (10-domain).

The overall restoration results for the two topologies are shown in Figs. 8 (NSFNET) and 9 (10-domain). In addition, the resulting *bandwidth blocking rates* (BBR) for *working* (non-restoration) connections are also plotted in Figs. 10 and 11 for these two topologies. Foremost, results for the modified NSFNET topology in Fig. 8 show that E2E crankback outperforms intermediate crankback, particularly at higher loads. This is due to the fact that the latter tries to re-route multiple failed connections from the domain immediately prior to the failed link. In turn, such "localized" recovery exacerbates resource contention on inter-domain links in the two domains adjacent to the failed link. This problem is particularly evident in the modified NSFNET topology, which has a relatively lower number of outbound links in each domain. By contrast, the E2E scheme achieves better load distribution as it attempts path restoration over the wider network. Meanwhile, the results for the 10-domain topology in Fig. 9 also show better performance with the E2E scheme. However, since this topology has higher inter-domain connectivity and fewer domains (i.e., lower average domain-hop counts on routed paths), the resultant separation between the E2E and intermediate schemes is much lower, particularly at lighter loads. The results in Figs. 8 and 9 also show that *joint* intra/inter-domain crankback generally yields higher restoration rates versus just intra- or inter-domain only crankback. Finally, related BBR results in Figs. 10 and 11 for working-only connections show slightly lower blocking rates with E2E restoration for both network topologies. These findings are very important as they confirm that improved post-fault recovery performance (from E2E restoration in Figs. 8 and 9) does not come at the expense of increased blocking of regular working connection requests.

The resource efficiency and recovery delays are then gauged for the two restoration schemes. Specifically, the average inter-domain path length is plotted in Figs. 12 (NSFNET) and 13 (10-domain) and the average restoration delay for successfully restored connections is plotted in Figs. 14 (NSFNET) and 15 (10-domain). Here, link propagation delays are set to 1 ms in the 10-domain network and to realistic distance-based propagation values in NSFNET. Furthermore, OXC nodal processing delays are also set to 0.05 ms. From these results, it is clear that E2E restoration consistently yields better performance for both of these restoration metrics. Namely, for the modified NSFNET topology the average path utilization with E2E crankback is about 6%–14% lower at most loads and the average restoration delay is also about 6%–19% lower. Meanwhile, for the 10-domain topology, the average path utilization is about 1%–8% lower with E2E crankback and the average restoration delay is about 1%–7% lower. By contrast, intermediate restoration gives higher crankback re-tries, thereby leading to longer average path length and delay. Nevertheless, the separation between the two schemes is moderated in 10-domain topology because of smaller domain counts and increased inter-domain connectivity.

Furthermore, it is noted that when using intra- or inter-domain only crankback in the modified NSFNET topology (i.e., $H_1 = 6/H_2 = 0$ and $H_1 = 0/H_2 = 6$), the E2E scheme actually shows decreasing average path length and delay for increasing traffic loads. This indicates that under these "boundary" parameter settings, the E2E scheme tends to prefer shorter paths. To show this more clearly, the number of successful restoration attempts with recovery times greater than 14 ms are measured for inter-domain only crankback ($H_1 = 0/H_2 = 6$) for the modified NETNET

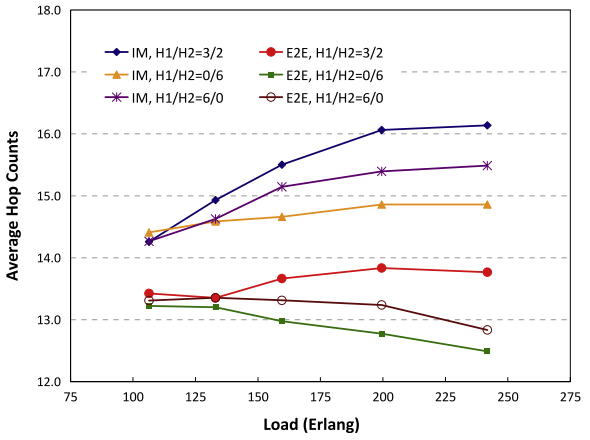

Fig. 12. Average inter-domain path length (NSFNET).

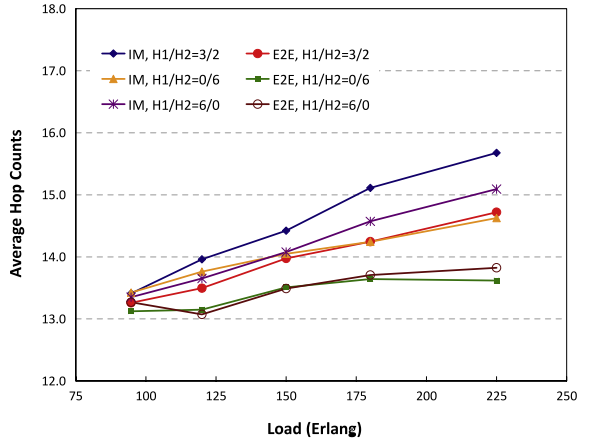

Fig. 13. Average inter-domain path length (10-domain).

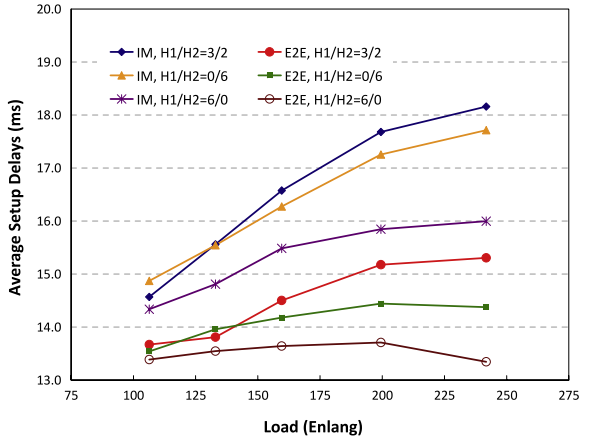

Fig. 14. Average restoration delay (NSFNET).

topology. Here, the results for intermediate restoration show that this percentage increases from 44.4% (at 108 Erlang) to 53.9% (at 240 Erlang). Conversely, in the E2E scheme, this percentage actually decreases from 36.4% (at 108 Erlang) to 34.4% (at 240 Erlang).

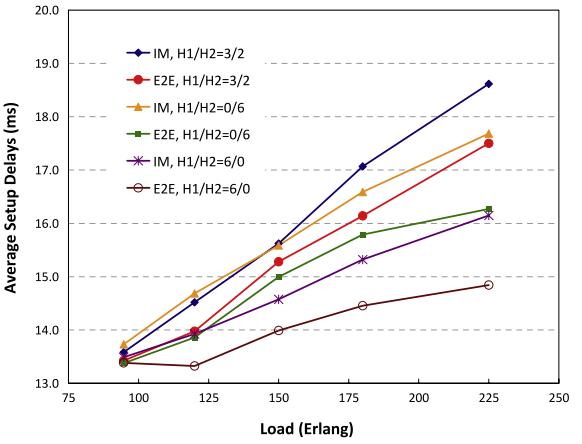

Fig. 15. Average restoration delay (10-domain).

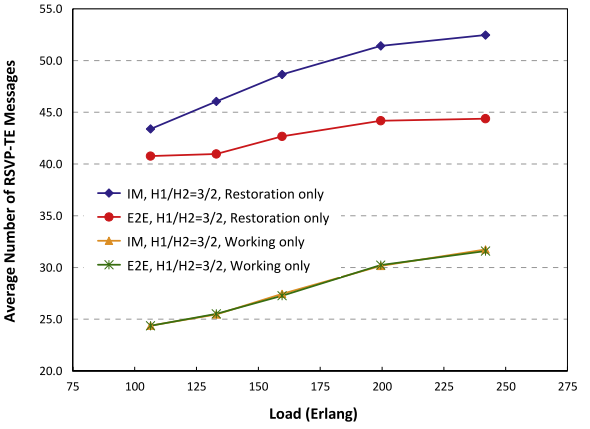

Fig. 16. Average number of messages (NSFNET).

Finally, in order to gauge the signaling control overheads of the crankback scheme, the average RSVP-TE message loads for the restored connections are measured and compared against those for working connections, i.e., Figs. 16 (NSFNET) and 17 (10-domain). These results clearly indicate that restoration crankback poses notably higher signaling overheads than regular working crankback, i.e., over 50% higher for both networks. To an extent, this is expected as the restored routes are not necessary the most efficient ones. In particular, intermediate restoration yields the highest overheads, as its restored route lengths tend to be longer (see Figs. 12 and 13). This is particularly evident in the NSFNET topology which has increased node counts and lower inter-domain connectivity.

5. Conclusions

This paper proposes novel crankback solutions for post-fault restoration of single failure scenarios in multi-domain IP/MPLS networks. Namely, a dual crankback counter approach is used to limit the number of intra/inter-domain crankback re-tries for end-to-end and intermediate restoration strategies. In addition, link failures and crankback history is also leveraged to improve the

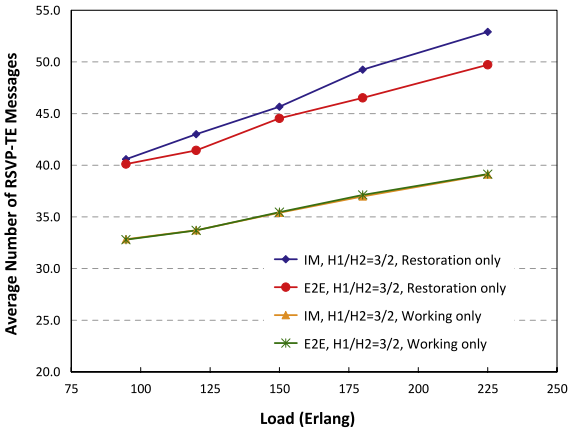

Fig. 17. Average number of messages (10-domain).

overall success and speed of the recovery process. Finally, improved next-hop domain selection strategies are developed to drive the overall search process by using existing (limited) inter-domain routing state. Detailed performance results for two test topologies with varying network characteristics (i.e., domain counts and inter-domain connectivity) show very high post-fault recovery, particularly for end-to-end crankback restoration. Future studies will look at extending this work for multi-domain DWDM settings and multiple failure scenarios.

Acknowledgement

This work has been funded in part by the US Department of Energy and Defense Threat Reduction Agency (DTRA, Basic Research Program). The authors are very grateful for this support.

References

[1] N. Ghani, et al., Control plane design in multidomain/multilayer optical networks, IEEE Communications Magazine 46 (6) (2008) 78–87.
[2] R. Zhang, J. Vasseur, MPLS inter-autonomous systems traffic engineering (TE) requirements, in: IEFT RFC 4226, November 2005.
[3] J. Ash, J. Le Roux, A path computation element (PCE) communication protocol generic requirements, in: IETF RFC 4657, September 2006.
[4] D. Staessens, et al., Enabling high availability over multiple optical networks, IEEE Communications Magazine 46 (6) (2008) 120–111.
[5] T. Takeda, et al. Analysis of inter-domain label switched path (LSP) recovery, in: IETF Internet draft-ietf-ccamp-inter-domain-recovery-analysis-02.txt, September 2007.
[6] A. Sprintson, et al. Reliable routing with QoS guarantees for multi-domain IP/MPLS networks, in: IEEE INFOCOM 2007, Alaska, May 2007.
[7] D. Truong, B. Thiongane, Dynamic routing for shared path protection in multi-domain optical mesh networks, OSA Journal of Optical Networks 5 (1) (2006) 58–74.
[8] A. Farrel, et al. Crankback signaling extensions for MPLS and GMPLS RSVP-TE, in: IETF Request RFC 4920, July 2007.
[9] S. Dasgupta, et al., Path-computation-element-based architecture for interdomain MPLS/GMPLS traffic engineering: overview and performance, IEEE Network 21 (4) (2007) 38–45.
[10] F. Aslam, et al., Interdomain path computation: challenges and solutions for label switched networks, IEEE Communications Magazine 45 (10) (2007) 94–101.
[11] F. Aslam, et al. Inter-domain path computation using improved crankback signaling in label switched networks, in: IEEE ICC 2007, Glasgow, Scotland, June 2007.
[12] C. Pelssner, O. Bonaventure, Path selection techniques to establish constrained interdomain MPLS LSPs, in: IFIP International Networking Conference, Coimbra, Portugal, May 2006.
[13] M. Esmaeili, F. Xu, N. Ghani, C. Xie, M. Peng, Q. Liu, Enhanced crankback for lightpath setup in multi-domain optical networks, IEEE Communications Letters (May) (2010).
[14] M. Esmaeili, F. Xu, N. Ghani, M. Peng, Q. Liu, Enhanced crankback signaling in multi-domain optical networks, in: IEEE/OSA Optic Fiber Communications Conference, OFC, 2010, San Diego, CA, March 2010.
[15] F. Xu, M. Peng, M. Esmaeili, N. Ghani, Advanced crankback provisioning for multi-domain networks, in: IEEE HPSR 2010, Dallas, TX, June 2010.