# Advanced Crankback Provisioning for Multi-Domain Networks

F. Xu[1], M. Peng[2], M. Esmaeili[1], N. Ghani[1]
[1]*University of New Mexico,* [2]*Wuhan University*

***Abstract:*** *Multi-domain traffic engineering is a major focus area for carriers and crankback signaling offers a very promising alternative. However, even though various crankback studies have been done, there remains significant latitude for improved multi-domain designs. To address these challenges, this work develops a novel solution for intra/inter-domain signaling crankback in IP/MPLS networks. Namely, dynamic intra-domain link-state routing information is coupled with inter-domain path/distance-vector routing state to improve the search process. Mechanisms are also added to limit setup signaling overheads and track crankback history from congested links. The performance of the proposed solution is analyzed using simulation and compared against other techniques including hierarchical inter-domain routing.*

***Keywords:*** *Crankback signaling, multi-domain networks*

## I. INTRODUCTION

*Traffic engineering* (TE) in IP-based *multi-protocol label switching* (MPLS) and optical *generalized MPLS* (GMPLS) networks is a very well-studied problem area. A wide range of constraint-based routing solutions have been proposed, but most have focused on single "domain" settings [1],[2]. However, as user application demands grow, there is a strong desire to achieve TE provisioning across multiple domains, i.e., inter *autonomous system* (AS) TE, particularly for higher-end applications such as *voice over IP* (VoIP), packet video, *virtual private network* (VPN), etc. Owing to obvious scalability and confidentiality concerns here, it is clear that this must be achieved in a distributed, decentralized manner.

To address these challenges, a diverse set of provisions have emerged to help improve multi-domain TE support, both at the IP/MPLS and underlying optical GMPLS layers [1]-[5]. On the standards side, many ubiquitous routing protocols already provide varying levels of inter-domain visibility, e.g., next-hop/path-vector dissemination in *exterior gateway protocol* (EGP) and hierarchical link-state dissemination in two-level *open-shortest-path-first* (OSPF-TE). Furthermore, the new IETF *path computation element* (PCE) [3] framework also defines multi-domain path computation strategies.

Meanwhile on the research side, a host of multi-domain TE schemes have been studied. A key focus here has been to address the tradeoff between inter-domain visibility and control plane complexity (i.e., dissemination, computation) [1]. The major contributions here are graph-theoretic topology abstractions for compressing domain-level state in IP/MPLS [6]-[9] and optical *dense wavelength division multiplexing* (DWDM) networks [10]-[13]. However, even though hierarchical routing delivers good performance, associated routing overheads are very high. Hence these schemes will be problematic in realistic settings where carriers tend to prefer distance/path-vector protocols, e.g., *border gateway protocol* (BGP) variants. These latter types only provide hop-count and reachability state and most operational versions do not support any *quality-of-service* (QoS) parameters, e.g., delay, bandwidth, etc. Hence to address these concerns, alternate "per-domain" computation schemes have been proposed for multi-domain TE [14]-[16], leveraging crankback signaling to overcome lower inter-domain visibility. Although good blocking gains are also seen here, related signaling overheads/setup delays are significant [15].

In light of the above, there is a clear need (and significant scope) to further develop new multi-domain crankback solutions and gauge their performance against counterpart hierarchical routing schemes. Along these lines, this paper proposes an enhanced crankback solution for multi-domain networks using the standard *resource reservation* (RSVP-TE) protocol. A key aim here is to address realistic settings where individual domains have full internal visibility via link-state routing, e.g., OSPF-TE, but otherwise generally limited "next-hop" inter-domain visibility, e.g., BGP routes. Namely, two levels of crankback are defined—intra and inter-domain—and active domain-level crankback history (i.e., congestion state) is also leveraged. Note that even though the work herein focuses on bandwidth provisioning IP/MPLS networks, it can readily be extended to optical wavelength-routing networks.

The paper is organized as follows. Section II first presents a survey of the latest work on multi-domain TE provisioning, including standards and research activities. Next, Section III details the proposed enhanced intra/inter-domain crankback solution. Performance analysis is then presented in Section IV and the results compared against advanced hierarchical inter-domain link-state routing schemes. Finally, conclusions and future research directions are highlighted in Section V.

## II. BACKGROUND

The IETF has defined a range of TE capabilities for multi-domain provisioning. Foremost, the PCE framework has been introduced to decouple path computation from signaling by defining domain-level computational entities. These entities can either reside in a standalone manner or be co-located with nodes and have access to the internal domain resource/policy databases. At the inter-domain level, these PCE entities can interact in a distributed manner to resolve end-to-end routes and two approaches have been defined for varying levels of "global" state, i.e., *per-domain* and *PCE-based* [3],[15]. The former schemes compute TE paths in a "domain-to-domain" manner and are most germane for limited inter-domain visibility. Meanwhile the latter schemes rely upon the head-end PCE to computes a *partial* or *loose* route to the destination (domain sequence) and are more suited for increased inter-domain visibility. However, even after path computation, blocking can occur during signaling along a chosen route. Hence new RSVP-TE crankback extensions have been defined to re-try alternate routes [4]. Specifically, varying levels of

multi-domain crankback have been outlined, i.e., *local domain*, *intermediate domain*, *source-based*.

Researchers have also studied various multi-domain TE schemes, broadly grouped as *hierarchical (link-state) routing* or *per-domain* strategies. In the latter, local domain topology/ resource state is condensed to generate an "abstracted" graph with fewer vertices and links, e.g., at a designated controller in each domain. This state is then flooded to other domains using hierarchical *link-state* routing between border gateways to build a "global" aggregated graph. Subsequent studies on multi-domain IP/MPLS networks have also proposed a variety of graph abstractions (e.g., star, mesh, tree, spanner graphs, etc) to compress bandwidth [6], bandwidth-delay [7],[8], and even diversity/survivability [9] information. When coupled with various computation heuristics (widest-shortest, shortest-widest, generalized cost, etc), these schemes yield very good blocking reduction and lower setup signaling overheads.

Now topology abstraction/hierarchical routing has also been applied in multi-domain DWDM networks, i.e., to summarize wavelength/converter/risk-group state. For example, [10] uses simple-node abstraction for all-optical domains. Meanwhile, [11] and [12] develop full-mesh and star schemes for more realistic multi-domain settings with partial (boundary) conversion. Distributed *routing and wavelength assignment* (RWA) algorithms are also defined to leverage this "global" state. Findings show good inter-domain blocking reduction with full-mesh abstraction (about 20-40% lower than single node abstraction), albeit routing overheads are much higher, almost 200-300% higher. Further abstractions for multi-domain optical survivability are also presented in [13].

Nevertheless, topology abstraction entails significant link-state routing overheads at the inter-domain level, e.g., second level of OSPF-TE [1],[12]. Hence the adoption of this approach may be limited in real-world settings where more scalable distance/path-vector protocols are already well-entrenched. Along these lines, a handful of studies have proposed *signaling-based* crankback strategies for "per-domain" path computation (akin to PCE classification [5]), in which individual domains compute their own traversing segments to build a concatenated end-to-end path. For example, [14] defines a basic *"per-domain"* (PD) crankback scheme which probes egress domain nodes for traversal routes and upon failure, notifies upstream border nodes. Overall results show higher request blocking rates and crankback delays, particularly when compared to PCE-based strategies utilizing pre-determined inter-domain routes. Meanwhile, [15] details a modified *compute while switching* (CWS) scheme for MPLS networks. Results here show good setup success rates as the scheme essentially mimics an exhaustive search. However, CWS entails very high signaling overheads/delays (not analyzed) and requires non-standard extensions to RSVP-TE attributes. Moreover, hitless post-setup flow switchovers may be difficult, especially in GMPLS settings. Finally, [16] addresses end-to-end path delays in multi-domain networks and presents two next-hop domain selection strategies. The first selects the next-hop as the "nearest" egress border node in the domain whereas the other uses tailored inter-domain *round-trip time* (RTT) measurements, i.e., pre-computed global state. The latter heuristic is shown to yield slightly higher carried load and less crankbacks. However, it requires adoption of a very specialized virtual coordinates system [16].

Overall, the above crankback solutions embody some good initial contributions. Still, added innovations are possible for multi-domain settings, e.g., novel schemes to limit crankback overheads/delays, improved history tracking and maintenance of crankback state, and use of available inter-domain distance/path-vector routing state. These are now considered.

## III. ENHANCED CRANKBACK SOLUTION

An enhanced multi-domain crankback solution is now presented based upon standard IETF protocols. The solution assumes realistic settings with full link-state routing at the intra-domain level (e.g., OSPF-TE) and more scalable path/distance-vector routing at the inter-domain level (e.g., BGP). Furthermore, each domain is assumed to have at least one PCE entity with full access to interior and exterior routing databases. This entity plays a key role in the crankback process as it helps resolve next-hop domains (i.e., find egress gateways). Meanwhile, all setup signaling is done using the recent crankback framework defined for RSVP-TE [4].

Overall, three key innovations are introduced to enhance multi-domain crankback, i.e., 1) dual intra/inter-domain crankback counters to limit signaling complexity/delay, 2) crankback history tracking to improve re-try domain traversal, and 3) intelligent per-domain search strategies leveraging existing "next-hop" EGP state. The details are now presented.

### A. Multi-Domain Crankback Operation

Before detailing the scheme, some initial notation is introduced. Consider a multi-domain network comprising of $D$ domains, with the $i$-th domain having $n^i$ nodes and $b^i$ border/gateway nodes, $1 \le i \le D$. This network is modeled as a set of domain sub-graphs, $\boldsymbol{G}^i(\boldsymbol{V}^i,\boldsymbol{L}^i)$, $1 \le i \le D$, 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 $i$ ($1 \le i \le D$, $1 \le j,k \le n^i$), i.e., $l^{ii}_{jk}$ is the link from $v^i_j$ to $v^i_k$ with available capacity $c^{ii}_{jk}$. A physical 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 $l^{ij}_{km}$ and has available capacity $c^{ij}_{km}$, $1 \le i,j \le D$, $1 \le k \le b^i$, $1 \le m \le b^j$. Also, $\boldsymbol{B}^i$ denotes the set of border nodes in domain $i$. Now consider the relevant RSVP-TE message fields. The path route is given by a node vector, $\underline{\boldsymbol{R}}$. Meanwhile, other fields are also defined for crankback [4], and include an exclude link vector, $\underline{\boldsymbol{X}}$, to track crankback failure history as well as dual intra/inter-domain crankback counters, $h_1$ and $h_2$ (whose usage will be detailed shortly). Note that [4] only defines a single counter field but bit masking can be used to generate two "sub-counters".

Per-domain computation is first overviewed for *non-crankback* operation. Consider a source node receiving a request for $x$ units of bandwidth to a destination node in another domain. This source queries its PCE to determine an egress link to the next-hop domain, e.g., via *PCE-to-PCE protocol* [3],[5]. The PCE then determines the next-hop domain to the destination domain (detailed in Section III.B) and returns a domain egress border node/link to this domain. Note that this response 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 (i.e., $c^{ii}_{jk} \geq x$) 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 basic hop count routing, see [6],[11]. Granted that an ER path is found, it is inserted in the path route vector, $\underline{\boldsymbol{R}}$, and RSVP-TE *PATH* messaging is then initiated (on expanded route) to the ingress border node in the next-hop domain. Here, each intermediate node checks for bandwidth on its outbound link and pending availability, propagates the *PATH* 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 an expanded *PATH*, the destination initiates an upstream *RESV* message.

Now clearly, insufficient bandwidth can cause downstream *PATH* setup failure at any route link. Hence various levels of crankback have been outlined [4], and two types are adapted here based upon *intra-domain* (local) and *inter-domain* (intermediate) crankback. Several key innovations are also introduced herein. First, dual counters ($h_1$, $h_2$) are defined to limit the number of re-tries. Namely, these counters are initialized to pre-specified limits ($H_1$ and $H_1$, respectively) in the initial *PATH* and are decremented during crankback to limit excessive setup delays/overheads for longer and less efficient paths. Next, crankback history tracking is introduced to help avoid congested links during the search process. Foremost, all congested links are noted in the setup signaling messages, i.e., both intra- and inter-domain links. In addition, *temporary* crankback history of congested *domain egress* links is also tracked by ingress border nodes over a fixed time window, $T_w$. Specifically, each ingress border node $v^i_j$ maintains a single history entry, $E^i_j$, which records the "latest" congested egress link found by signaling. This entry records the egress node's address, congested link id, and also the time when the entry was created. The aim here is to use this *inter-domain link* information to benefit future connections.

In terms of crankback operation, two key steps are defined here, i.e., *notification* and *re-computation*, as shown in Figure 1. The former refers to the (upstream) signaling procedures executed upon link congestion at an intermediate node, whereas the latter refers to the re-routing procedures to select a new route. In general, resource failures can occur at *three* different node types, i.e., domain ingress border nodes, domain egress border nodes, and interior nodes. However, in the proposed scheme only the former types perform *re-computation* whereas the latter two simply perform crankback *notification*. These steps are now detailed further.

**Crankback Notification:** Upstream crankback notification is done when there is insufficient bandwidth at an intra-domain link (at an intra-domain node) or an inter-domain link (at an egress border node) on an already-expanded ER. The overall procedure here is shown in Figure 2. Namely, the forward *PATH* message is terminated and its relevant fields updated/ 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" congested link is noted. Note that if blocking occurs in the source domain, the *PATH_ERR* is sent to source node.

An example of crankback notification is shown in Figure 1 for interior and egress border nodes ($H_1$, $H_2$=2). For example, consider bandwidth blocking on the link $l^{ii}_{42}$, i.e., step 1. Here, the interior node $v_4{}^i$ prunes the route vector $\underline{\boldsymbol{R}}$ to the domain ingress node, adds the blocked link to the exclude route vector $\underline{\boldsymbol{X}}$, decrements the intra-domain counter $h_1$, and sends all this information back to the ingress node $v_1{}^i$ via a *PATH_ERR* message. A similar procedure is also shown for blocking at the egress border node $v_3{}^i$ (i.e., step 2, Figure 1).

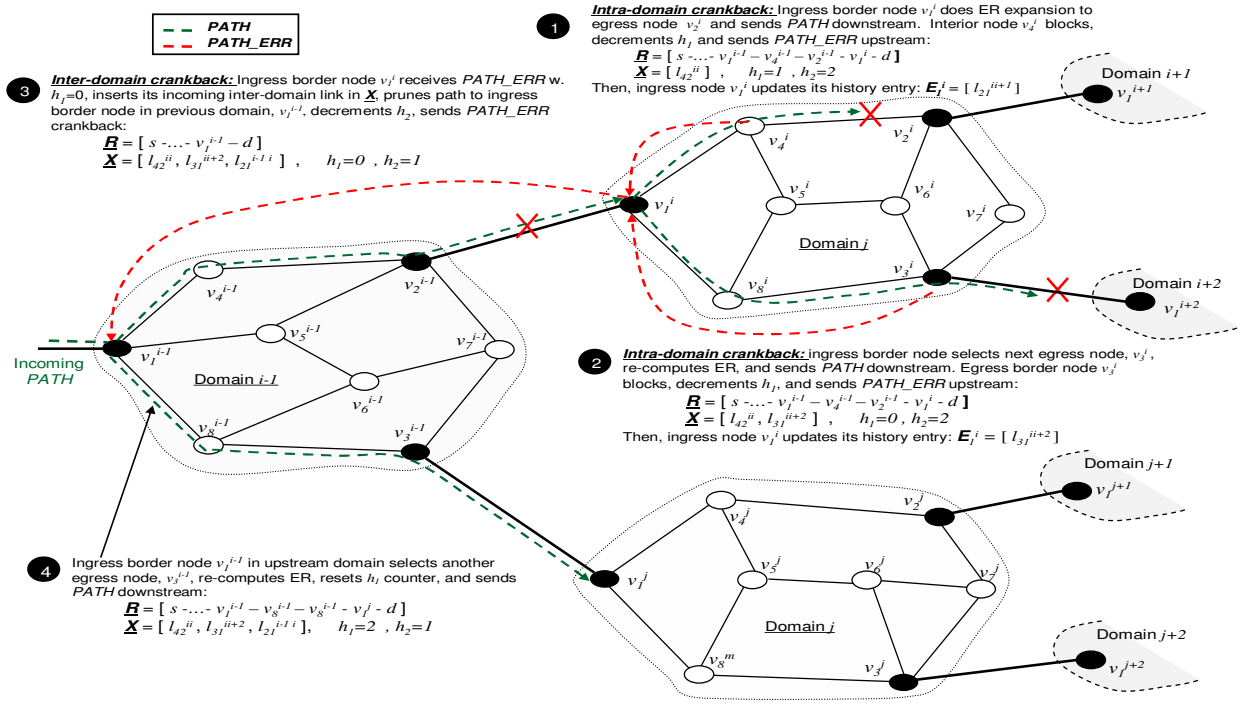


Figure 1: Enhanced intra/inter-domain crankback scheme ($H_1$=2, $H_2$=2)

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

Figure 2: Crankback notification procedure

**Crankback Re-Computation:** Path re-routing is done by ingress border nodes receiving a *PATH_ERR*, and the overall procedure is shown in Figure 3. (Note that for special case of a source domain, the receiving source node relays the *PATH_ERR* to its PCE for processing). First, upon receiving the *PATH_ERR* message, an immediate ingress border node, $v_j^i$, updates its crankback history entry, $E_j^i$, with any new information on congested egress link, i.e., track inter-domain links only. This is done by checking if the latest link added to $\underline{\boldsymbol{X}}$ is a domain egress link and is not the same as the currently-stored value. If so, a new entry is generated to overwrite the current one in $E_j^i$ (including node address/link id/current time).

Next, two types of re-computation procedures are attempted, i.e., intra- and inter-domain. Consider the *intra-domain* case first. If the arriving $h_1$ counter in the *PATH_ERR* message is not expired, a next-hop domain/egress border node is selected by the ingress border node (or PCE) for ER expansion. Here the exact sequence of next-hop domains tried is pre-computed to try *successively longer* inter-domain routes (i.e., via multi-entry distance vector table, detailed in Section III.B). The enhanced scheme also makes full use of crankback history to avoid any previously "failed" congested intra/inter-domain links. Namely, all inter-domain links in $\underline{\boldsymbol{X}}$ and/or in the (unexpired) crankback history entry $E_j^i$ that egress from the domain are removed from consideration, i.e., only consider "non-congested" next-hop domain egress links. Here, an unexpired crankback history entry $E_j^i$ is defined as one whose time-stamp falls within $T_w$ seconds of the current time. Additionally, all intra-domain links in the exclude link vector $\underline{\boldsymbol{X}}$ are also removed from *local* ER computation. Finally, 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.

However, it is clear that one may not be able to establish a domain-traversing route all the time, i.e., $h_1$ expired, LR expansion failure to selected egress node, all egress border links in exclude link vector $\underline{\boldsymbol{X}}$, etc. Hence in these cases added *inter-domain* crankback is initiated by the ingress border node to achieve a more global response. Namely, a *PATH_ERR* message is sent 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) with the $h_1$ counter reset to $H_1$. To improve history tracking, the notifying ingress border node here also inserts its own *ingress link* in the exclude route vector of the *PATH_ERR* message, i.e., in order to avoid future re-tries on this link. Note that inter-domain crankback is only initiated if the $h_2$ value is non-zero, otherwise the request is dropped (i.e., *PATH_ERR* to source, Figure 3). Finally, if the prior domain is the source domain, the *PATH_ERR* is sent to the source.

Crankback re-computation is also shown in Figure 1. For example, when blocking initially occurs on link $l^{ii}{}_{42}$, the ingress border node $v_1^i$ re-tries egress node $v_3^i$ if corresponding egress link $l^{ii+2}{}_{31}$ is not matching history entry $E_3^i$. When this second attempt fails at a *domain egress* link, i.e., $l^{ii+2}{}_{31}$, the ingress node $v_1^i$ receives a *PATH_ERR* with a zero $h_1$ counter. After storing the congested link $l^{ii+2}{}_{31}$ into its crankback history entry, $E_3^i$, this ingress node also marks its ingress link $l^{i-1\,i}{}_{21}$ as congested, prunes the route to the ingress border node in prior domain *i-1*, $v_1^{i-1}$, and sends a *PATH_ERR* message (step 3, Figure 1). This upstream node then initiates a re-try to a new egress border node $v_3^{i-1}$ (step 4, Figure 1).

```
/* Check and updated crankback history entry */
if (newly-added link in X is a domain egress link &&
    different from that stored in Ej^i)
    Extract new congested egress domain link from X, store
    new node address/link id values and current time in Ej^i
/* Attempt intra-domain re-routing at ingress border node vj^i */
while (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 and/or (unexpired) Ej^i

    if (next hop egress node found)
        Make copy of local network graph (via IGP database),
        prune all local "failed" congested links 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;
/* 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
   Copy h1, h2, R, X fields,generate PATH_ERR,send to source
```

Figure 3: Crankback re-computation (at ingress node $v_j^i$)

Overall, the above scheme limits the number of intra-domain attempts to $H_1$ per domain and the number of inter-domain attempts to $H_2$, i.e., maximum of $H_1H_2$. Also, note that link admission control (during *PATH* processing) only considers bandwidth constraints in this study, but future considerations can also incorporate delay and other metrics.

### *B.* Next-Hop Domain Computation

As mentioned earlier, a key provision in the enhanced crankback scheme is the use of existing inter-AS state (i.e., BGP databases) to improve the search process. This is achieved by pre-computing a *multi-entry* distance vector table at all domain border nodes (or PCE) to list up to $K$ next-hop domains/egress links to each destination domain, see Figure 4. Namely, at domain $i$, the $k$-th 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 $k$-th shortest "domain-level" hop-count path to domain $j$ ($1 \le i, j \le D$, $i \ne j$, $1 \le k \le K$). Clearly the number of entries to a destination will be upper-bounded by the minimum of $K$ and the maximum number of inter-

domain links egressing from the domain. Now consider the actual computation of this table at a border node (or PCE) in domain *i*. The algorithm first uses the inter-AS (EGP) path database to build a "simple node" graph model [6] of the global network, $\boldsymbol{H}(\boldsymbol{U},\boldsymbol{E})$, where $\boldsymbol{U}$ is the set of domains $\{\boldsymbol{G}^i\}$ reduced to vertices and $\boldsymbol{E}$ is the set of physical inter-domain links $\{l^{ij}_{km}\}$, $i \neq j$. An iterative shortest-path scheme is then used to compute multiple routes to all other destination domains over $\boldsymbol{H}(\boldsymbol{U},\boldsymbol{E})$ as follows:

```
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
```

Figure 4: Next-hop domain computation procedure

The above 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., $\boldsymbol{H'}(\boldsymbol{U},\boldsymbol{E})$. At the $k$-th 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 or the vertice for domain $i$ in $\boldsymbol{H'}(\boldsymbol{U},\boldsymbol{E})$ becomes disconnected. Hence the next-hop domain selection procedure during crankback re-computation (as detailed in Section III.A) simply searches these $k$ table entries, $T^i(j,k)$, to a destination domain $j$ in increasing order. This sequentially drives crankback along fixed "domain-level" sequences of increasing length, but with added provisions to avoid "failed" table entries (in $\underline{\boldsymbol{X}}$). Overall, these entry tables will be relatively static if EGP state changes are infrequent.

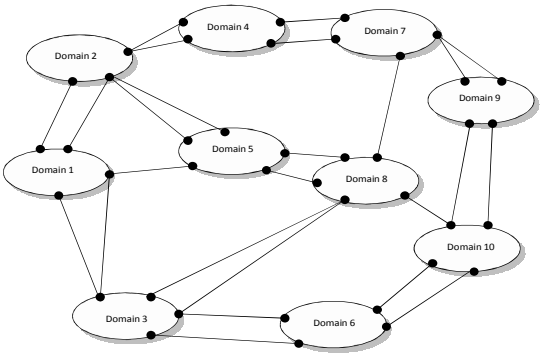


Figure 5: 10-domain topology (only border nodes shown)

## IV. PERFORMANCE EVALUATION

The performance of the enhanced multi-domain crankback solution is tested by developing specialized models in *OPNET Modeler*™. Here, a sample 10-domain test topology used, Figure 5, with 25 inter-domain links. Each domain has about 10 interior nodes and about 4-5 border nodes. Moreover, multi-homed interconnection is also done for some domains to reflect realistic settings, e.g., dual-homing between domains 6 and 10. Also, link capacities are set to 10 Gbps and connection requests sizes are varied from 200 Mbps–1 Gbps in increments of 200 Mbps, i.e., fractional Ethernet. All requests are generated between random nodes in randomly-selected domains and each run is averaged over 250,000 connections. In addition, mean connection holding times are set to 600 sec (exponential) and related inter-arrival times are also exponential and varied with load. Furthermore, a maximum of $K$=5 next-hop domain entries are computed in the distance vector table, although the number searched is limited by the $H_2$ value set in the simulation run. Finally, the time duration window, $T_w$, for tracking crankback history entries is set to 600 sec, i.e., equivalent to mean connection holding time.

Now a key objective here is to compare against hierarchical inter-domain link-state routing with topology abstraction, i.e., simple node, full-mesh [6],[7],[11]. Briefly consider these schemes in more detail here. In full-mesh abstraction, the PCE computes "abstract links" to condense trans-domain routes, $O(|\boldsymbol{B}^i|(|\boldsymbol{B}^i|-1))$ state. Here, the capacity of an abstract link is derived as the mean bottleneck capacity of the $k$-shortest paths between the respective border nodes [11]. These links (along with physical inter-domain links) are then advertised using a second level of OSPF-TE between border nodes [1]. Namely, link updates are generated using *significance change factors* (SCF) and hold-off timers [1], and the respective values are set to 10% (SCF) and 200 sec (hold-off timer). This inter-domain link state is then used to build a "global" topology for computing/ expanding end-to-end *loose-routes* (LR). Meanwhile in simple node abstraction, all domains are condensed to virtual nodes and only *physical* inter-domain links are advertised, i.e., no domain-internal state.

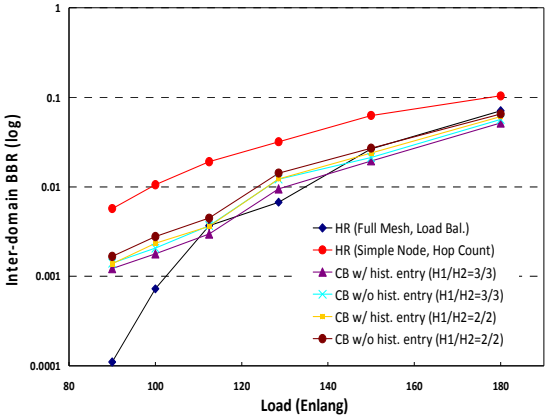


Figure 6: BBR (crankback, hierarchical routing)

First, the inter-domain *bandwidth blocking rates* (BBR) are plotted for the various schemes in Figure 6 ("HR" for hierarchical routing, "CB" for crankback). Moreover, several configurations are tested for the enhanced crankback scheme, including joint intra/inter-domain crankback ($H_1$=2/$H_2$=2 and $H_1$=3/$H_2$=3), both with/without crankback history tracking. The overall results here show some key findings. Foremost, when compared with hierarchical routing, the enhanced

crankback scheme notably outperforms simple-node abstraction and also closely tracks the performance of the more advanced full-mesh abstraction (at mid-high loads). Furthermore, the use of history tracking (i.e., $E_j^i$ values) also gives lower blocking, e.g., by about 5-10%. Finally, it is observed that moderate intra/inter-domain crankback counter limits yield very competitive results. For example, BBR reduction tends to level off after $H_1$=3/$H_2$=3 and added runs (not shown) with larger crankback values such as $H_1$=5/$H_2$=5 actually yield worsening BBR performances due to increased route lengths and bandwidth fragmentation.

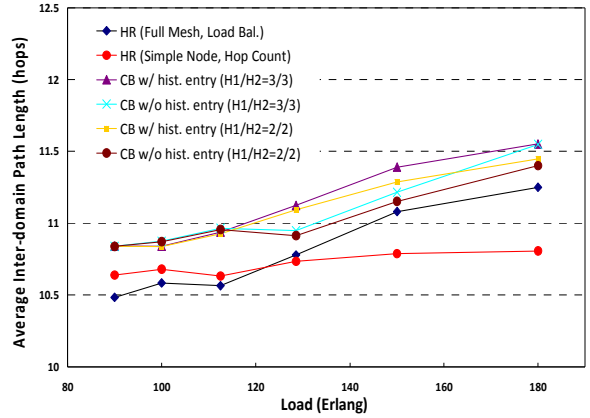


Figure 7: Average inter-domain path length

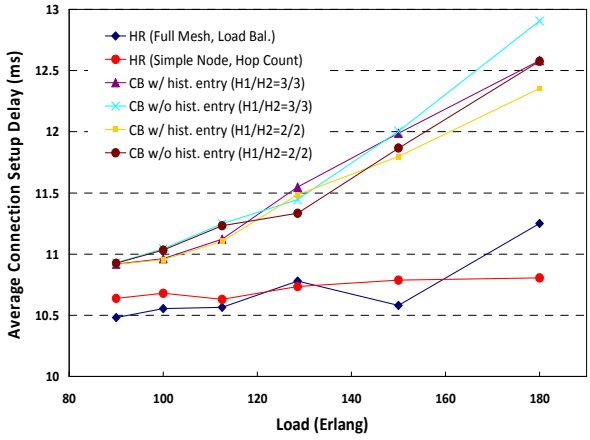


Figure 8: Average connection setup delay

Next, the resource usage/efficiencies of the respective schemes are gauged by plotting the average inter-domain path lengths, as shown in Figure 7. Here, it is seen that increased *inter-domain* crankback counters (i.e., $H_1$=3/$H_2$=3) result in the highest utilizations, particularly at increased loads. The use of crankback history entries also yields increased path lengths, as it tends to preclude shorter routes with congested links. However, lower levels of crankback (i.e., $H_1$=2/$H_2$=2) without history tracking can almost match the resource usage levels of hierarchical routing with full-mesh abstraction. As expected, simple node abstraction has the lowest overheads of all schemes (although its BBR performance is not optimal).

Finally, connection setup delays are plotted in Figure 8 (for 0.5 ms link delays and 0.05 ms node processing delays). Here, the results show that the proposed crankback strategies yield 15-20% higher setup delays versus hierarchical routing (at high loads), which is deemed as an acceptable result. Overall, the above findings show that moderate intra/inter-domain crankback driven by parsed distance vector state can yield very good overall performance.

## V. CONCLUSIONS

This paper studies crankback signaling for multi-domain networks and introduces several key innovations. Foremost, improved next-hop domain selection strategies are introduced to drive the search process with minimal inter-domain routing overheads. A dual counter scheme is also used to limit the number of intra/inter-domain crankback attempts. Finally, crankback history in the form of congested link state is tracked and leveraged at both the messaging and border node levels in order to improve the overall success of the setup process. Detailed performance results show much-improved blocking performance with the proposed scheme, i.e., as compared with more exhaustive "end-to-end" crankback schemes and even hierarchical inter-domain link-state routing with no topology abstraction. Future studies will look at extending this work for survivability considerations.

## VI. ACKNOWLEDGEMENTS

This research has been supported by the *Department of Energy* (DOE) under Award ER25828 and the *National Science Foundation* (NSF) under Award 0806637.

## REFERENCES

[1] N. Ghani, *et al*, "Control Plane Design in Multidomain/Multilayer Optical Networks," *IEEE Comm. Mag.,* Vol. 46, No. 6, June 2008, pp. 78-87.
[2] R. Zhang, J. Vasseur, "MPLS Inter-Autonomous Systems Traffic Engineering (TE) Requirements," *IEFT RFC 4226*, November 2005.
[3] J. Ash, J. Le Roux, "A Path Computation Element (PCE) Communication Protocol Generic Requirements," *IETF RFC 4657*, September 2006.
[4] A. Farrel, *et al*, "Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE," *IETF Request RFC 4920*, July 2007.
[5] P. Torab, *et al*, "On Cooperative Inter-Domain Path Computation," *IEEE ISCC 2006*, Sardinia, Italy, June 2006.
[6] F. Hao, E. Zegura, "On Scalable QoS Routing: Performance Evaluation of Topology Aggregation," *IEEE INFOCOM 2000*.
[7] T. Kormaz, M. Krunz, "Source-Oriented Topology Aggregation with Multiple QoS Parameters in Hierarchical Networks," *ACM TOMACS,* Vol. 10, No. 4, October 2000, pp. 295-325.
[8] K. Liu, *et al*, "Routing with Topology Abstraction in Delay-Bandwidth Sensitive Networks," *IEEE/ACM Transactions on Networking*, Vol. 12, No. 1, February 2004, pp. 17-29.
[9] A. Sprintson, *et al*, "Reliable Routing with QoS Guarantees for Multi-Domain IP/MPLS Networks," *IEEE INFOCOM 2007*, Alaska, May 2007.
[10] S. Sanchez-Lopez, *et al*, "A Hierarchical Routing Approach for GMPLS-Based Control Plane for ASON," *IEEE ICC 2005*, Korea, June 2005.
[11] Q. Liu, *et al*, "Hierarchical Routing in Multi-Domain Optical Networks", *Computer Comm.* Vol. 30, No. 1, December 2006.
[12] Q. Liu, C. Xie, T. Frangieh, N. Ghani, A. Gumaste, N. Rao, T. Lehman, "Inter-Domain Routing Scalability in Optical DWDM Networks," *IEEE ICCCN 2008*, US Virgin Islands, August 2008.
[13] D. Truong, B. Thiongane, "Dynamic Routing for Shared Path Protection in Multi-Domain Optical Mesh Networks," *OSA Journal of Optical Networks*, Vol. 5, No. 1, January 2006, pp. 58-74.
[14] S. Dasgupta, J. C. de Oliveira, J. P. Vasseur, "Path-Computation-Element-Based Architecture for Interdomain MPLS/GMPLS Traffic Engineering: Overview and Performance," *IEEE Network*, Vol. 21, No. 4, July/August 2007, pp. 38-45.
[15] F. Aslam, *et al*, "Inter-Domain Path Computation Using Improved Crankback Signaling in Label Switched Networks," *IEEE ICC 2007*, Glasgow, Scotland, June 2007.
[16] C. Pelssner, O. Bonaventure, "Path Selection Techniques to Establish Constrained Interdomain MPLS LSPs", *Proc. of IFIP Intn$^l$ Networking Conference*, Coimbra, Portugal, May 2006.