December 2020

EMBEDDED BACKUP PATH INFORMATION FOR STATELESS SRV6 MICRO SEGMENT IDENTIFIER NODE PROTECTION

Nagendra Kumar Nainar

Kamran Raza

Carlos M. Pignataro

Rajiv Asati

EMBEDDED BACKUP PATH INFORMATION FOR STATELESS SRV6 MICRO SEGMENT IDENTIFIER NODE PROTECTION

AUTHORS:
Nagendra Kumar Nainar
Kamran Raza
Carlos M. Pignataro
Rajiv Asati

ABSTRACT

Techniques presented herein provide for the ability to improve Topology Independent Loop-Free Alternate Fast Re-route (TI-LFA) with node protection by embedding backup path information (on a per segment basis) in a stack itself. In particular, techniques herein introduce a Node protected Prefix/Adjacency Segment Identifier (NP-SID) with a new (yet simple) forwarding semantic that can be used to embed backup path information directly in a segment stack. The NP-SID is always followed by the backup node information in the segment stack and the forwarding semantic to the NP-SID will involve a lookup on the backup information only if a protected node fails. The techniques may provide a benefit of being purely stateless by not involving context tables on nodes.

DETAILED DESCRIPTION

Topology Independent Fast Re-Route (TI-FRR) seeks to provide guaranteed FRR coverage in any Interior Gateway Protocol (IGP) network by establishing the protection over post-convergence paths. Internet Engineering Task Force (IETF) draft "draft-ietf-rtgwg-segment-routing-ti-lfa" defines the procedure required to perform the same that is applicable for both Segment Routing (SR) with Multiprotocol Label Switching (SR-MPLS) and SR over Internet Protocol (IP) version 6 (SRv6). While TI-LFA seeks to provide protection for any failures, it is still a challenge in scenarios where a transit node fails and if the transit node is part of the segment stack. Consider an example SRv6 topology as shown below in Figure 1 with reference to various nodes, R1–R7.

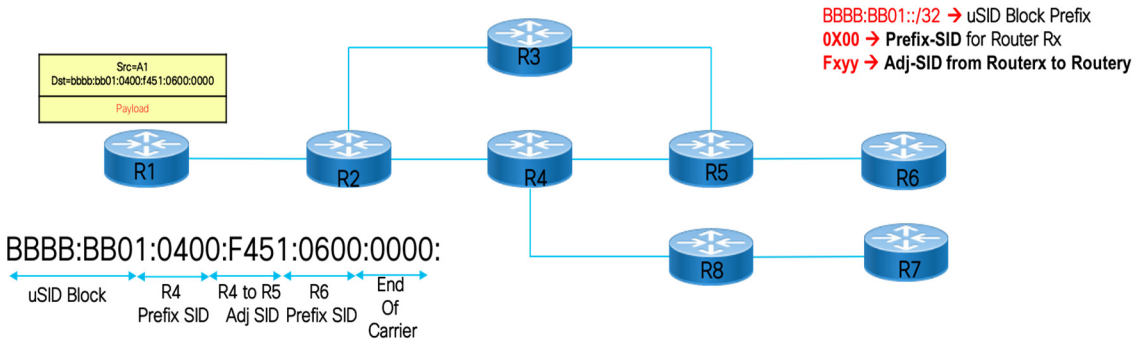

Figure 1: Example SRv6 Topology

As illustrated in Figure 1, consider that node R1 steers traffic over nodes R1->R4->R5->R6 using <BBBB:BB01><0400><F451><0600> as the segment stack. In this stack, SIDs specific to node R4 and node R6 are included to achieve the required traffic steering. For this example topology, consider that a failure occurs for node R4, as shown in Figure 2, below.

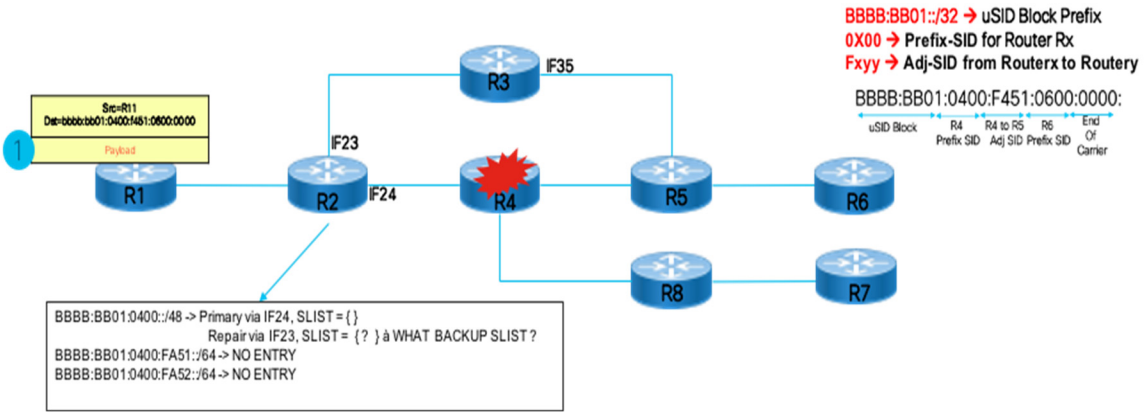

Figure 2: Example Failure involving the SRv6 Topology

As illustrated above, if the R4 node fails, when node R2 receives a packet, the node cannot protect the traffic and ends up dropping the traffic until node R1 detects the failure and avoids inserting the R4 node SID in the stack. Protection in such scenarios is not straightforward, as a node (e.g., node R2) does not know the forwarding instruction associated with a local SID of a failed node (e.g., node R4). With more critical applications relying on software-defined transports, it would be advantageous to avoid such failures.

One option to address such failures may involve creating context tables and maintaining forwarding instructions of node R4 on node R2 so that the R2 node can perform a lookup in the relevant context table upon a failure of the R4 node. This approach is also explained in "draft-hegde-spring-node-protection-for-sr-te-paths." However, such an approach is tedious, as it requires the entire table of all the neighbors to be maintained locally for protection purposes. This can drastically increase the Routing Information Base (RIB)/Forwarding Information Base (FIB) load on node R2 and also involves additional computation to re-compute the same each time there is a topology change.

This proposal provides a stateless data plane (segment stack) based approach in which backup details are directly embedded in the stack header that could be used for protection purposes for instances in which node failures may occur. In particular, techniques of this proposal introduce a Node protected Prefix/Adjacency Segment Identifier (NP-SID) with a new (yet simple) forwarding semantic that can be used to embed backup path information directly in the segment stack. The NP-SID is always followed by the backup node information in the segment stack and the forwarding semantic to the NP-SID will involve a lookup on the backup information only if a protected node fails.

Various example details that illustrate various features of the techniques of this proposal are provided herein below with reference to the example topology discussed above for Figure 1. For segment ID assignment, each node is assigned 2 additional SIDs including:

- A Node Protected Prefix SID (NPP-SID); and
- A Node Protected Adjacency (Adj) SID (NPA-SID)

Consider for the example SRv6 topology, as discussed above and as illustrated below in Figure 3, that each SID is assumed to be 16-bits in size.

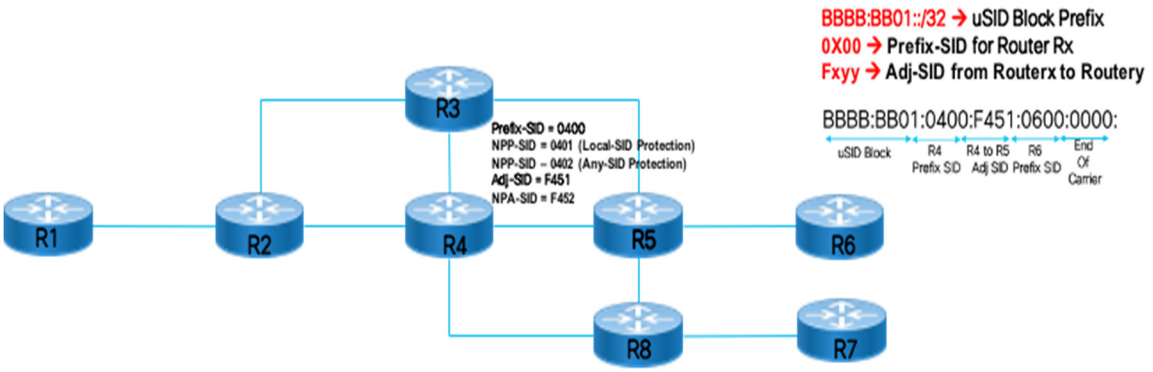

Figure 3: Example SID Assignment Details

For the above topology, node R4 can be assigned with the following set of segments:

- **0400** --> Traditional Prefix SID
- **0401** --> Node Protected Prefix SID (For Local-SID Protection)
- **0402** --> Node Protected Prefix SID (For Any-SID Protection)
- **F451** --> Traditional Adj-SID from node R4 to node R5
- **F452** --> Node Protected Adj-SID from node R4 to node R5

In various implementations, the SID assignment can be provided for all nodes in the domain or only for critical nodes/adjacencies. For various examples involving techniques of this proposal discussed herein below, it is assumed that each protected node will have the above additional SIDs assigned.

Consider additional example details regarding configuring forwarding semantics for various nodes. Continuing with the example topology above, consider that node R4 is a protected node. Thus, any node connected to node R4 is considered to be an FRR node.

For configuring the forwarding sematic in FRR nodes, consider an example involving node R2. As with any Adj-SID, an NPA-SID is not installed by nodes other than an assigned node. Accordingly, the NPA-SID assigned by R4 will not be installed in the R2 node forwarding table. Rather, the forwarding semantic associated with the NPP-SID assigned by node R4 on node R2 can be configured as follows:

```
Prefix::/mask
--> Primary via egress_Intf //Primary Path
--> Repair // TI-LFA Path
-->COPY[63..127] to [32..112] && lookup //Analogous to SHIFT 2 SID
```

As illustrated in Figure 4, below, node R2 is the FRR node and assigns 'BBBB:BB01:0401::/48' (where BBBB:BB01 is the micro SID (uSID) block and '0401' is the NPP-SID of R4) with interface 24 (IF24) as the primary egress interface. Node R2 further installs a backup/repair path with a semantic involving shifting 2 SIDs and performing a lookup.

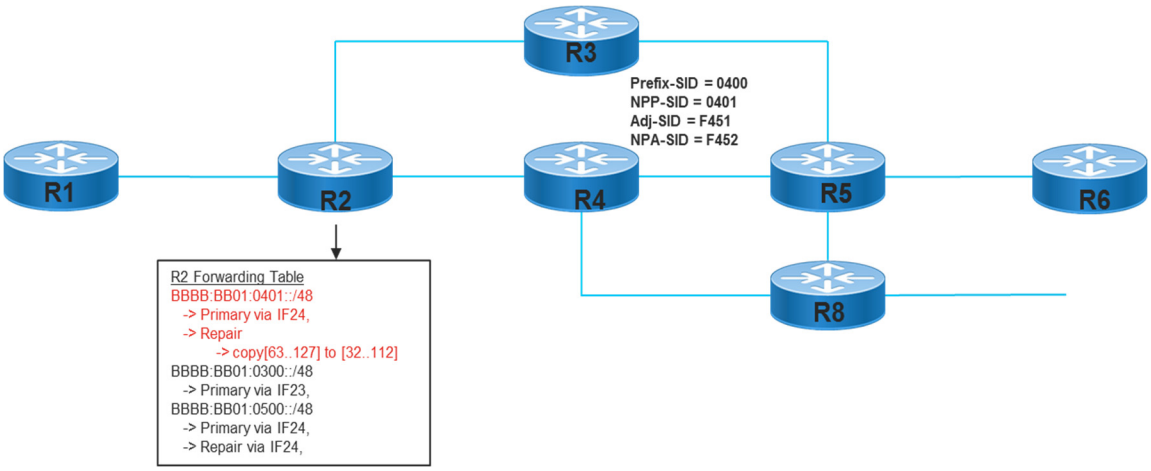

Figure 4: Example Forwarding Semantic Configuration Details

Consider various additional details involving configuring the forwarding semantic in protected nodes. In the present example, node R4 is the protected node. Thus, the forwarding semantic for the NP-SID (NPP and NPA SIDs) is illustrated in Figure 4 relative to node R4. Techniques of this proposal involve two types of lookup semantics, including:

- A single Longest Prefix Match (LPM) lookup approach (Used to protect a local SID); and
- A two-lookup approach (Used to protect any SID).

For the single LPM lookup approach involving the above example topology, node R4 can concatenate the Prefix NPA-SID and the Prefix-SID of the Adjacency to which NPA-SID is assigned as one LPM entry. For example, node R4 assigns F451 as the NPA-SID towards node R5. Thus, node R4 concatenates '<uSID-Block><R4-NPP-SID><R4R5-NPA-SID><R5-Prefix-SID>::/80' as one LPM entry in the forwarding table with the following semantic:

```
Prefix::/mask
--> COPY[80..127][32..112] && via IF45
```

As shown in Figure 5, below, node R4 installs 'BBBB:BB01:0401:F452:0500::/80' with a semantic of shifting 3 SIDs by copying bits 80..127 to 32..112 and strictly forwarding packets over IF45.

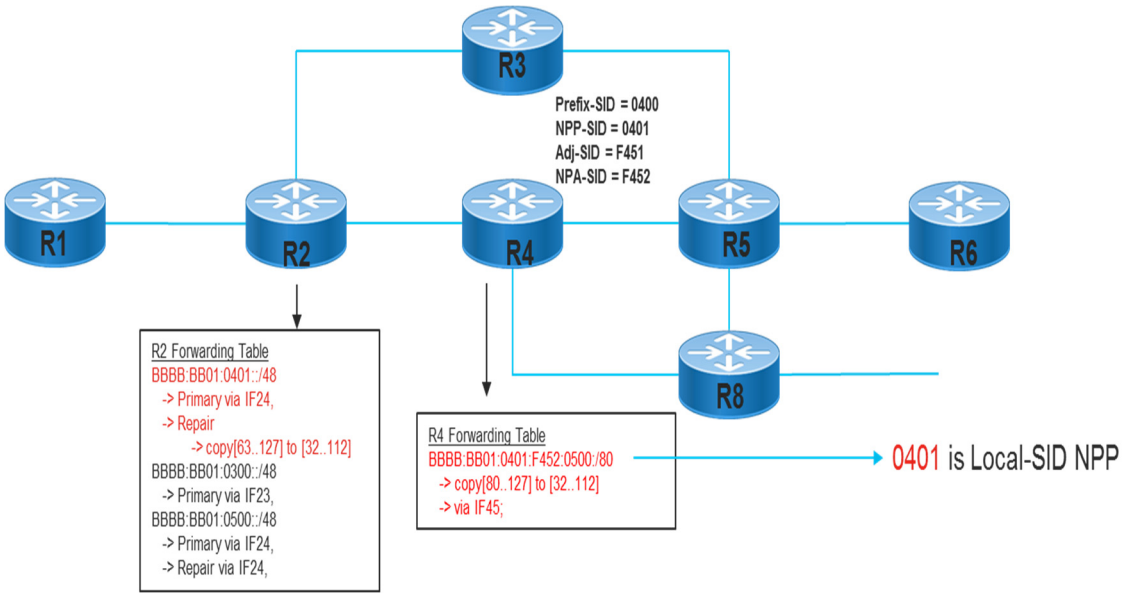

Figure 5: Single LPM Lookup Example Details

For the two-lookup approach involving the above example topology, node R4 can concatenate the uSID-Block to NPP-SID and configure an entry with a semantic involving shifting 2 SIDs and performs the second lookup as follows:

```
Prefix::/mask
--> COPY[64..127] to [32..112] && lookup
```

As shown in Figure 6, below, node R4 installs 'BBBB:BB01:0402::/48' with a semantic of shifting 2 SIDs by copying the bits 64..127 to 32..112 and then performing the second lookup.

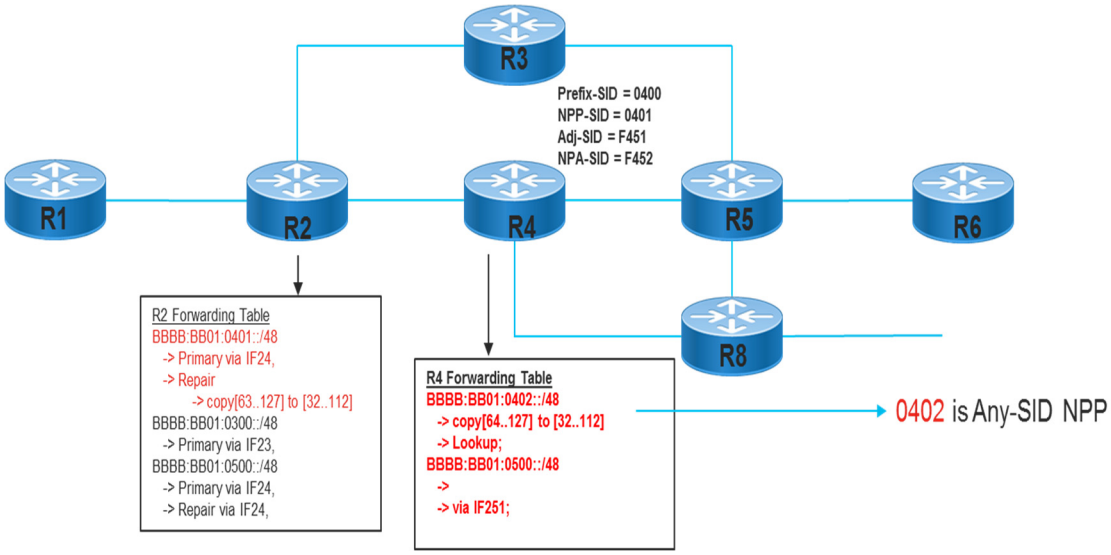

Figure 6: Two-Lookup Example Details

In various implementations, the single LPM lookup approach may be utilized when a NPP is used to protect the local SID while the two-lookup approach may be utilized when a NPP is used to protect any global SID.

Consider, for example, a first node protection implementation involving protecting a local SID, which is discussed with reference to Figures 7 and 8, below. In order to protect the local SID of node R4 (e.g., Adj-SID of node R4 to node R5), the NPP-SID of node R4, (0401) is utilized. The node protected segment stack is concatenated as '<uSID-Block><NPP-SID><NPA-SID><Prefix-SID-of-NPA>::'. In the example topology, when Adj-SID F451 is used, it is appended by the Prefix-SID of node R5 (0500) and prepended by '0401' (the NPP-SID of node R4).

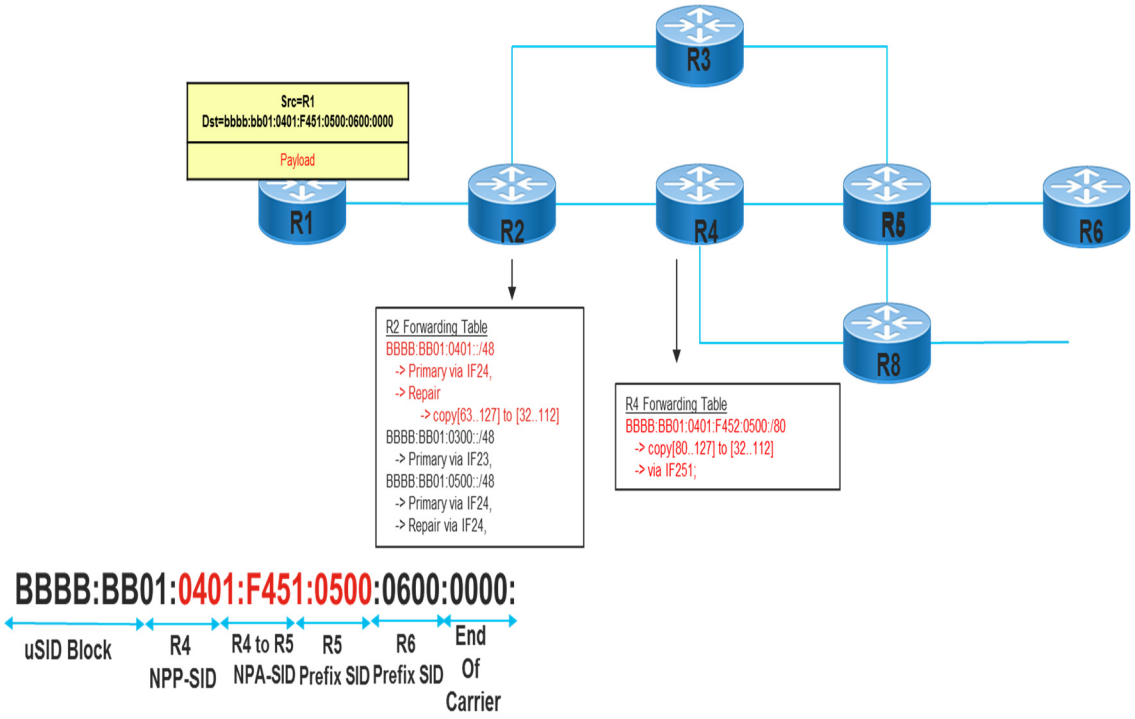

Figure 7: Normal Operation Details When Protecting a Local SID

During normal working conditions, as illustrated above in Figure 7, node R2 may receive a packet and match the primary path for 'BBBB:BB01:0401::/48' and forward the packet to node R4. Upon receiving the packet, node R4 will match 'BBBB:BB01:0401:F451:0500::/80', shift 3 SIDs, and forward the packet over IF45.

During a failure condition, as illustrated below in Figure 8, node R2 may receive a packet and can match the repair path for 'BBBB:BB01:0401::/48'. Additionally, node R2 copies [63..127] to [32..112] such that the segment stack becomes 'BBBB:BB01:0500::' and R2 can perform another lookup in the table and forward the packet towards node R5 (via node R3).

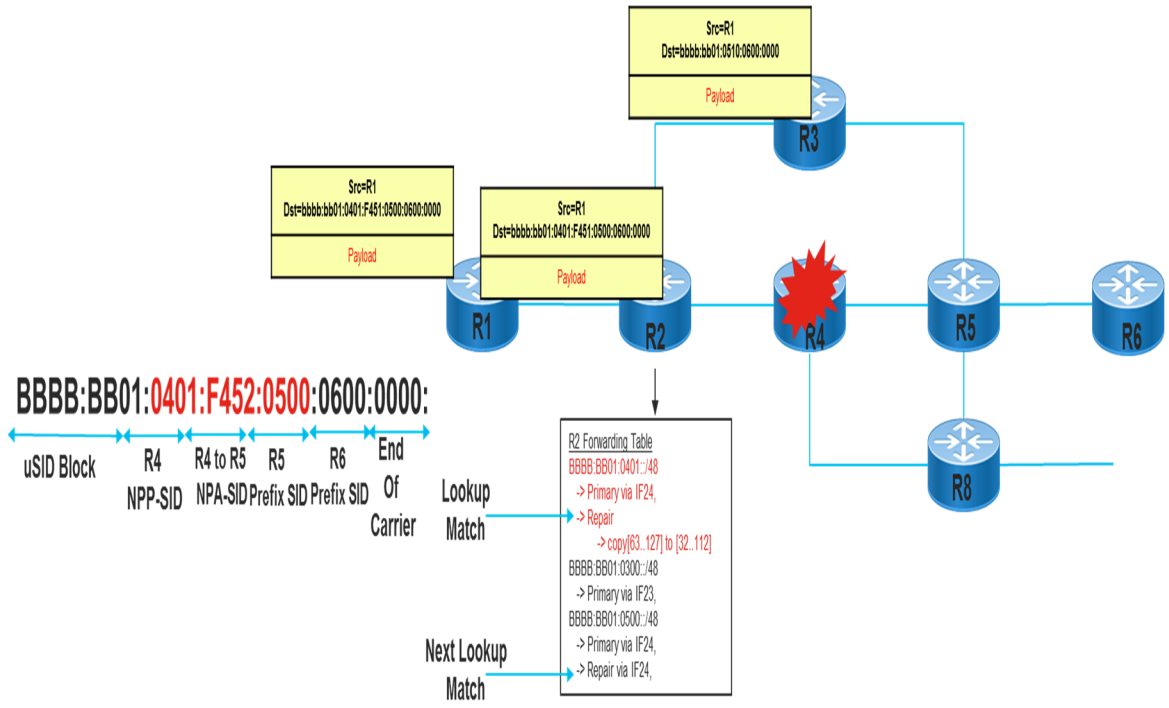

Figure 8: Failure Condition Details When Protecting a Local SID

Consider another example node protection implementation involving protecting any global SID, discussed with reference to Figures 9, 10, and 11, below.

In order to protect any global SID of node R4 (that follows R4), for example, the Prefix-SID of node R5, the NPP-SID of node R4 (0402) is utilized. In this example, the node protected segment stack is concatenated by copying the global SID twice after the NPP-SID as '<uSID-Block><NPP-SID><Prefix-SID><Prefix-SID>::.' In the example topology illustrated in Figure 9, below, when the Prefix-SID '0500' is inserted after the NPP-SID, it is appended once by the same SID (0500:0500) and prepended by '0402' (the NPP-SID of node R4).

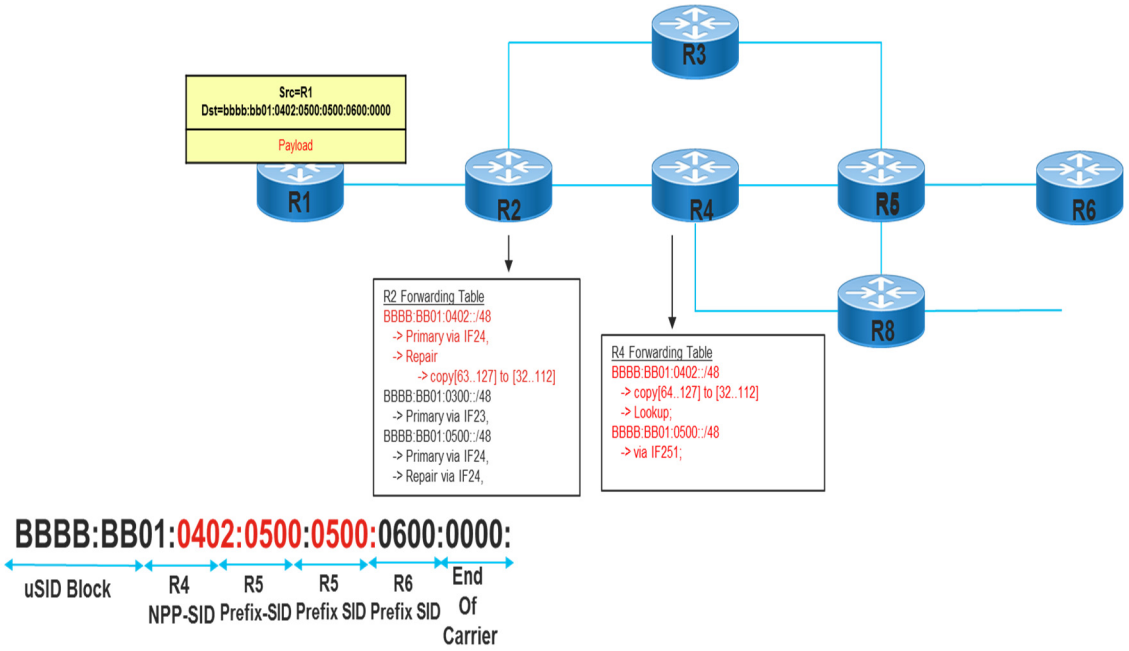

Figure 9: Example Configuration Details When Protecting any Global SID

During normal working conditions, as illustrated via Figure 10, below, when node R2 receives a packet, it can match the primary path for 'BBBB:BB01:0402::/48' and forward the packet to node R4. Upon receiving the packet, node R4 can match 'BBBB:BB01:0402::/48', shift 2 SIDs such that the resulting stack is 'BBBB:BB01:0500:0600::', perform a lookup to match 'BBBB:BB01:0500::/48' and forward the packet towards node R5.

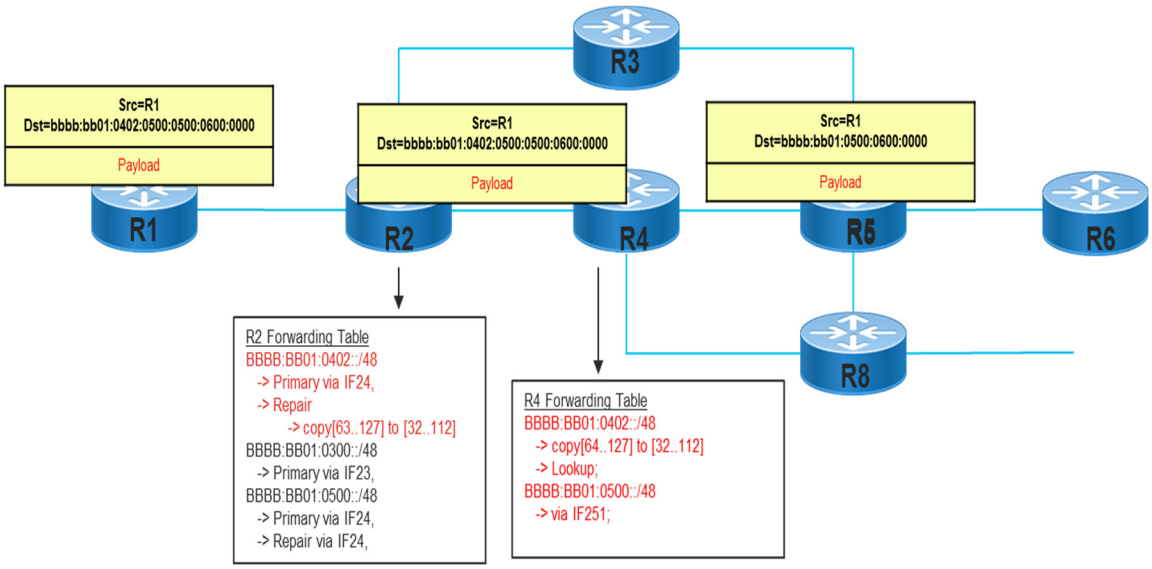

Figure 10: Normal Operation Details When Protecting a Local SID

During a failure condition, as illustrated below in Figure 11, node R2 may receive a packet and can match the repair path for 'BBBB:BB01:0402::/48', shift 2 SIDs such that the resulting stack is 'BBBB:BB01:0500::', perform another lookup, and then forward the packet via node R3 towards node R5.

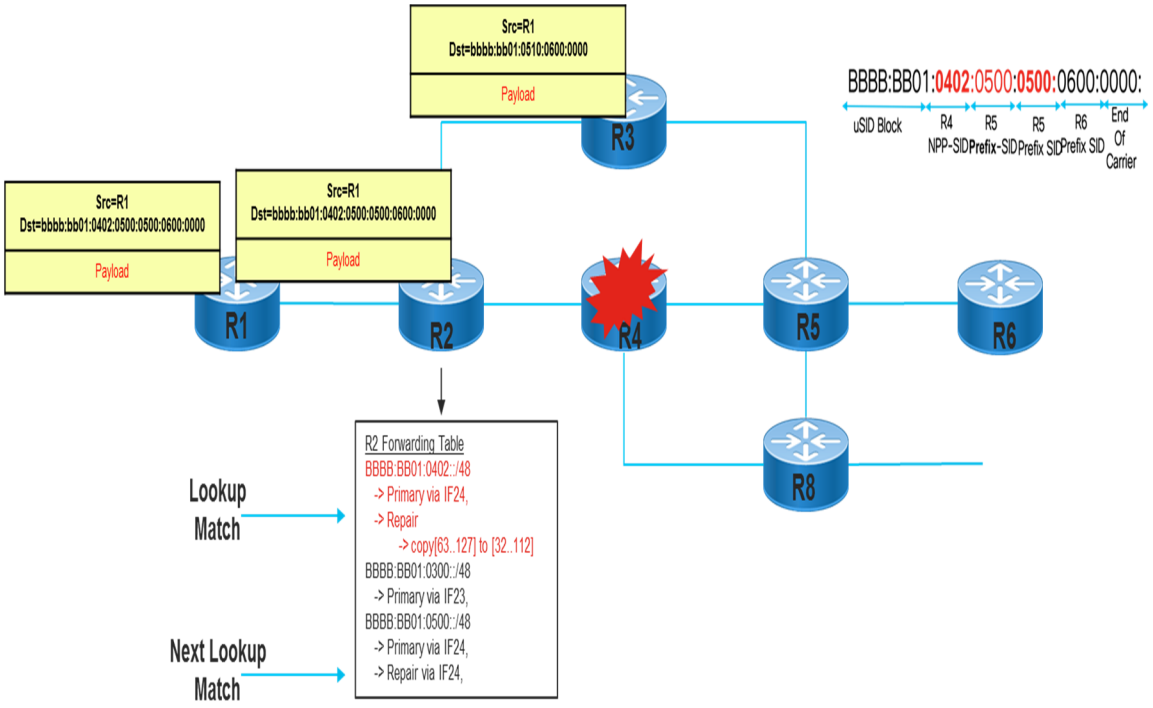

Figure 11: Failure Condition Details When Protecting any Global SID

In summary, techniques herein improve TI-LFA with node protection by embedding backup path information (on a per segment basis) in the stack itself. In particular, techniques herein introduce the NP-SID with a new (yet simple) forwarding semantic that can be used to embed backup path information directly in a segment stack. The NP-SID is always followed by the backup node information in the segment stack and the forwarding semantic to the NP-SID will involve a lookup on the backup information only if a protected node fails. Techniques herein may provide a benefit of being purely stateless by not involving context tables on nodes.