January 16, 2019

SERVICE LAYER DEPENDENCY MAPPING BASED HEATMAP FOR FAULT ISOLATION

Carlos M. Pignataro

Nagendra Kumar Nainar

Alejandro Eguiarte

SERVICE LAYER DEPENDENCY MAPPING BASED HEATMAP FOR FAULT ISOLATION

AUTHORS:
Carlos M. Pignataro
Nagendra Kumar Nainar
Alejandro Eguiarte

ABSTRACT

Techniques are described herein for a two-fold process where a first Machine Learning (ML) model is engaged in building a Dependency Mapping Table (DMT) based on the data from the network (network observations). An Interior Gateway Protocol (IGP) database, service graph information, and in-band Operations, Administration and Management (iOAM) data is leveraged to feed into the ML model to build the DMT. The DMT is built with a per physical element precision (fiber optical cable in each direction). In other words, the database is built in a way to enable identification of the list of (connected/non-connected) services and protocols that potentially relies on the fiber cable. In the second fold, the ML model uses the dependency mapping built in the previous phase to identify the potential cause for an issue and prioritize the relevant alarms. In addition, it may be tied together with existing failure prediction mechanisms for any layer that in turn will be used with the above database to prioritize the alarm/notification to an operator Operations Support System (OSS) and take any necessary pre-emptive action on the above layers.

DETAILED DESCRIPTION

Modern networks are deployed with various combinations of Underlay/Overlay/Service layers that solve different use cases and business purposes. Such deployments create a lot of cross layer dependency for end-to-end service delivery and any issue in one of the lower layer may result in impacting the other layers.

As illustrated in Figure 1 below, all nodes are connected through optical fibers, enabled with Open Shortest Path First (OSPF) / Intermediate System to Intermediate System (ISIS) with Virtual Extensible Local Area Network (VxLAN) / Segment Routing (SR) / SR over Internet Protocol version 6 (SRv6) Overlay tunnels to steer the traffic over a specific virtual Network Function (vNF) graph (service chaining). When there are any

issues at the optical layer, all (or most of) the above layers may be impacted causing an end service issue.

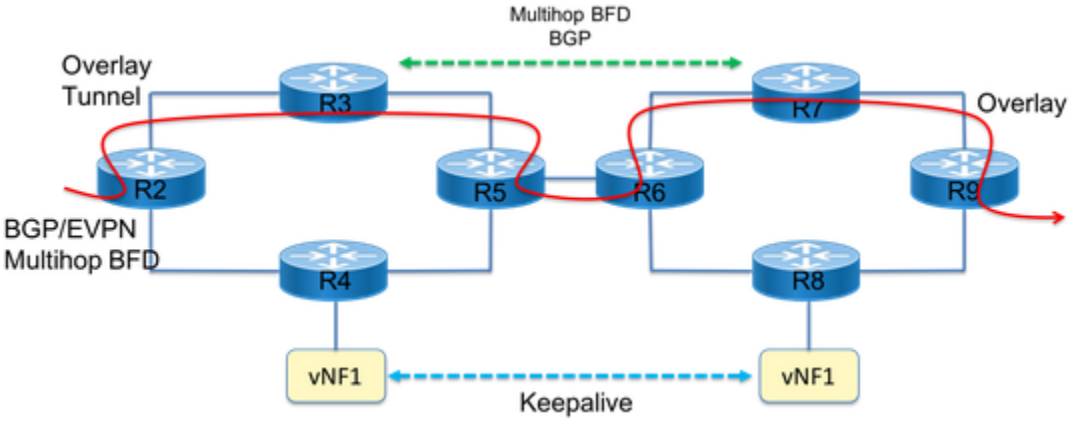

Figure 1

It is common to see Operations Support Systems (OSSs) implemented in different layers resulting in hundreds of alarms/notification from different layers for each such issue. In the above example, when the optical fiber between R5-R6 have some issues, it may result in the following set of alarms:

1. Optical related alarms from both R5 and R6.
2. Interface down errors and alarms from R5 and R6.
3. OAM protocol alarms (Bidirectional Forwarding Detection (BFD) / Seamless BFD (SBFD)) between R5 and R6.
4. Interior Gateway Protocol (IGP) alarms from R5 and R6.
5. Multihop BFD alarms from R3 and R7
6. Border Gateway Protocol (BGP) alarms from R3 and R7.
7. Overlay tunnel (SR/SRv6/VxLAN) alarms from R2 and R9
8. Keepalive alarms on vNF1 and vNF2.
9. More (depending on the services deployed).

A histogram of all the above alarms/notifications may be provided to the operators and let them isolate and narrow down the problem to the underlying optical layer. While in the above simplified topology, this results in many such notification, large networks tend to produce alarms in much higher magnitude based on the number of devices and services impacted. It would be beneficial is the above could be prioritized and the physical layer

identified as the cause that can be used for isolation such that any pre-emptive measures may be taken at the dependency layers.

Described herein is a Machine Learning (ML) based service dependency mapping that helps create a mapping between different layers. This may prioritize the relevant alarms and present the same to the operator as a cross layer heatmap on dependency layers, and use the learning to predict any issue at any layer and take pre-emptive measures at the dependency layers.

Techniques are described herein for a two-fold process where a first ML model is engaged in building a Dependency Mapping Table (DMT) based on the data from the network (network observations). An IGP database, service graph information, and in-band Operations, Administration and Management (iOAM) data is leveraged to feed into the ML model to build the DMT. The DMT is built with a per physical element precision (fiber optical cable in each direction). In other words, the database is built in a way to enable identification of the list of (connected/non-connected) services and protocols that potentially relies on the fiber cable. In the second fold, the ML model uses the dependency mapping built in the previous phase to identify the potential cause for an issue and prioritize the relevant alarms. In addition, it may be tied together with existing failure prediction mechanisms for any layer that in turn will be used with the above database to prioritize the alarm/notification to an operator OSS and take any necessary pre-emptive action on the above layers.

Phase one may involve dependency mapping generation. As illustrated in Figure 2 below, network data from different sources may be pre-processed to convert into an ML feature set which will be fed to the ML algorithm. In the above example, data from BGP Link State (BGP-LS) (topology), Link Layer Discovery Protocol (LLDP) (connectivity), optical, overlay, Path Computation Element (PCE) states, iOAM, Netflow, and telemetry data may be fed into the ML agent after pre-processing.

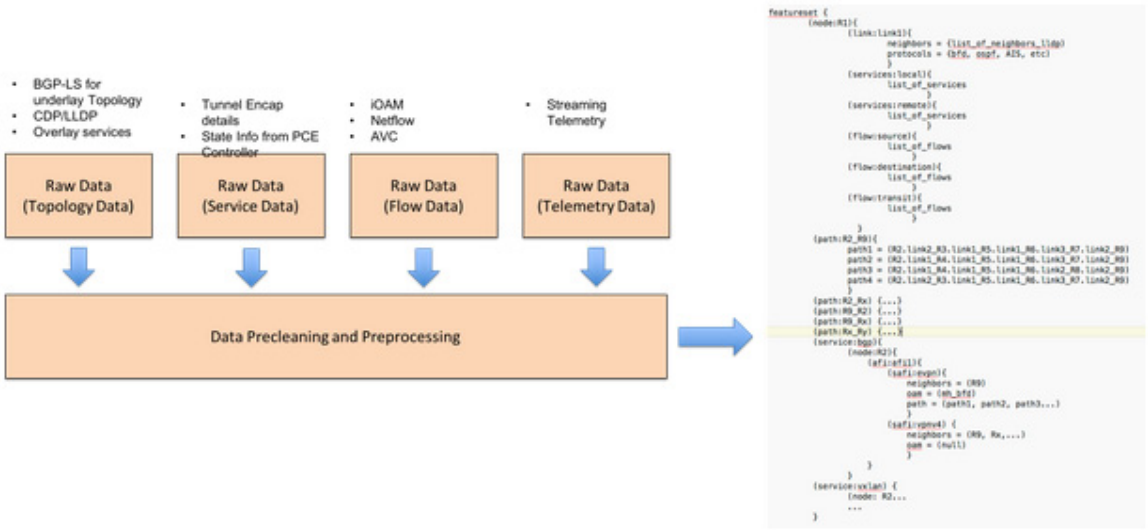

Figure 2

As shown, data from BGP-LS (topology), LLDP (connectivity), optical, overlay, PCE states, iOAM, Netflow, telemetry and timestamps may be cleaned and pre-processed to create an ML feature set suitable to be used as the inputs to train two ML agents: an Unsupervised ML (UML) agent and a Supervised ML (SML) agent.

Both ML agents may process data in two modes: Training Mode and Inference Mode. While in Training Mode, an ML model may be built using an ML algorithm where either the user or a training scheduler may define the start and end times for the training.

In Training Mode, the UML agent may be used to group the data into subgroups or clusters of similar or correlated of network observations. Each cluster may be assigned labels. The SML agent may use the Cluster Group Identifier (ID) (obtained via inference from the UML agent) and the ML feature set to train or retrain the ML model using an ML algorithm to classify the current state by providing the priority or severity.

In Inference Mode, the ML feature set may be input to the UML agent to determine the subgroup or cluster to which the given state of network observations belongs, assigning a Cluster Group ID. This Cluster Group ID may be fed to the SML agent to generate the DMT. This approach may require an ML architecture that accommodates a multi-class output vector.

Figure 3 below illustrates an example flow for creating the DMT.

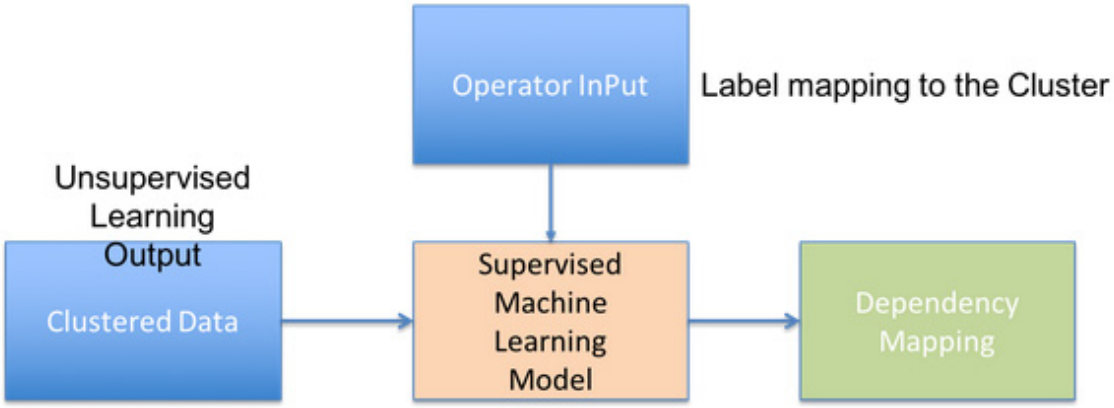

Figure 3

Figure 4 below illustrates an example (simplified) dependency for link R2_R3.

```
dependency_map {
      (link:R3_R5) {
            protocol = {ospf, bfd}
            services = (R2_R9.bgp.path1, R2.vxlan...)
            ...
```

Figure 4

In the above example, BGP between R2 to R9 over path1 depends on the above link and so on. As illustrated in Figure 5 below, this may be used to create a layered DMT.

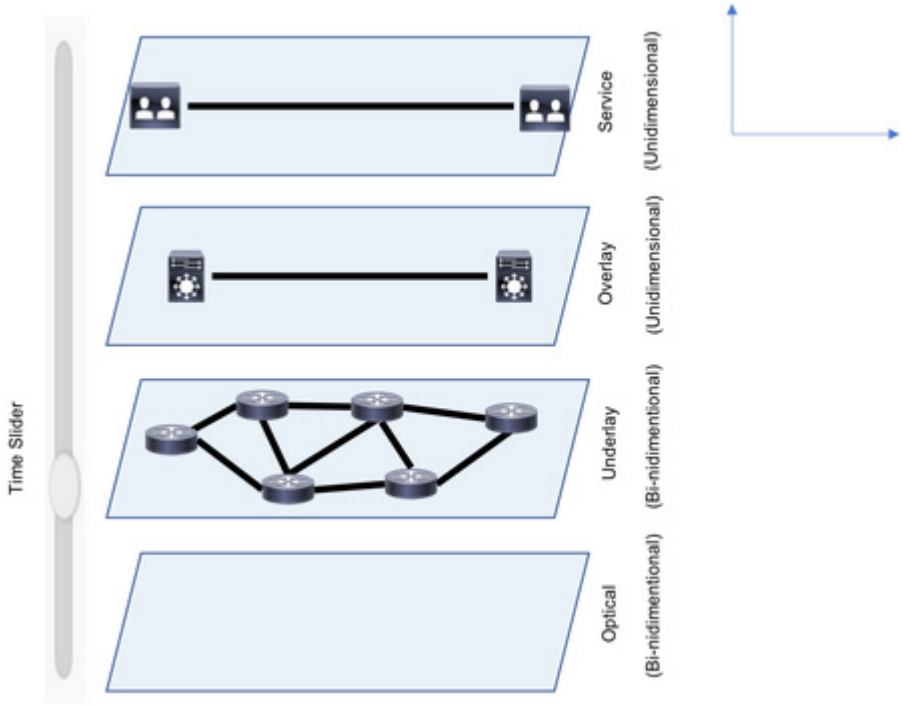

Figure 5

Phase two involves log prioritization and reporting. As illustrated in Figure 6 below, upon receiving any logs, it may be fed into a ML model.

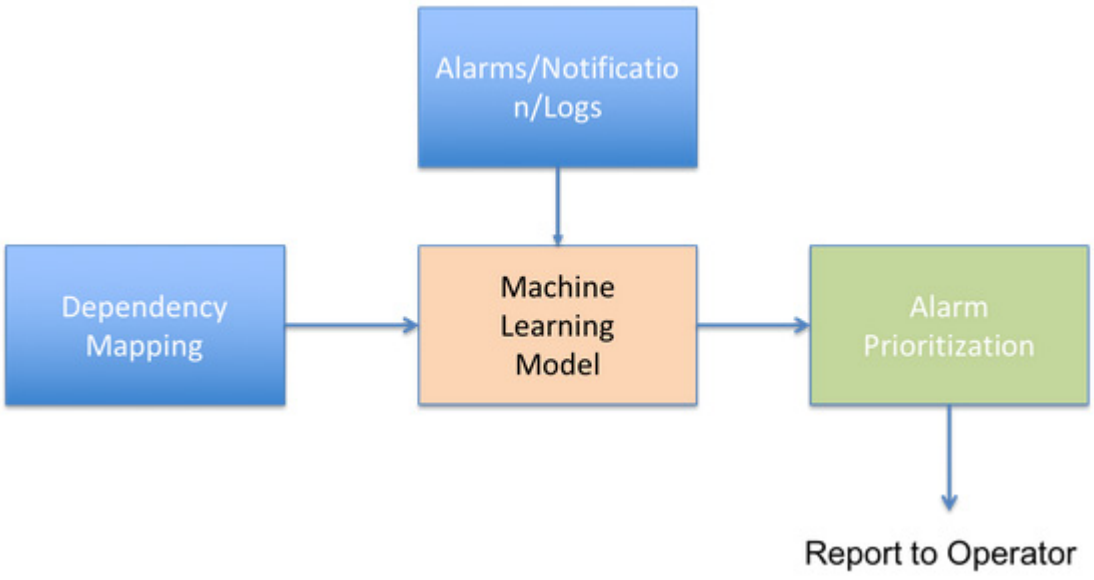

Figure 6

The DMT, event logs, alarms, and notifications may be the inputs for a ML agent that outputs a multi-class output vector predicting the overall priority of the given list of alarms/notifications/logs and indicating the affected layer. This may be presented to an operator as a histogram that highlights the events on each layer. Figure 7 below illustrates how link failure happens at time 0, causing alarms at underlay nodes at time 1.

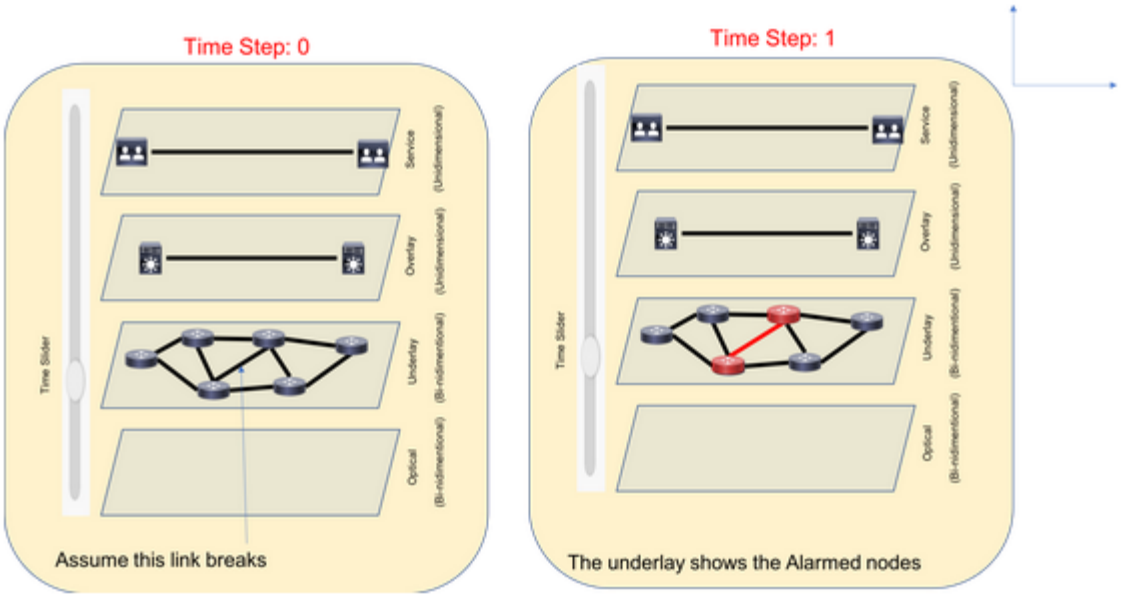

Figure 7

Figure 8 below illustrates additional notifications/alarms from relevant nodes impacted by the path followed by overlay alarms.

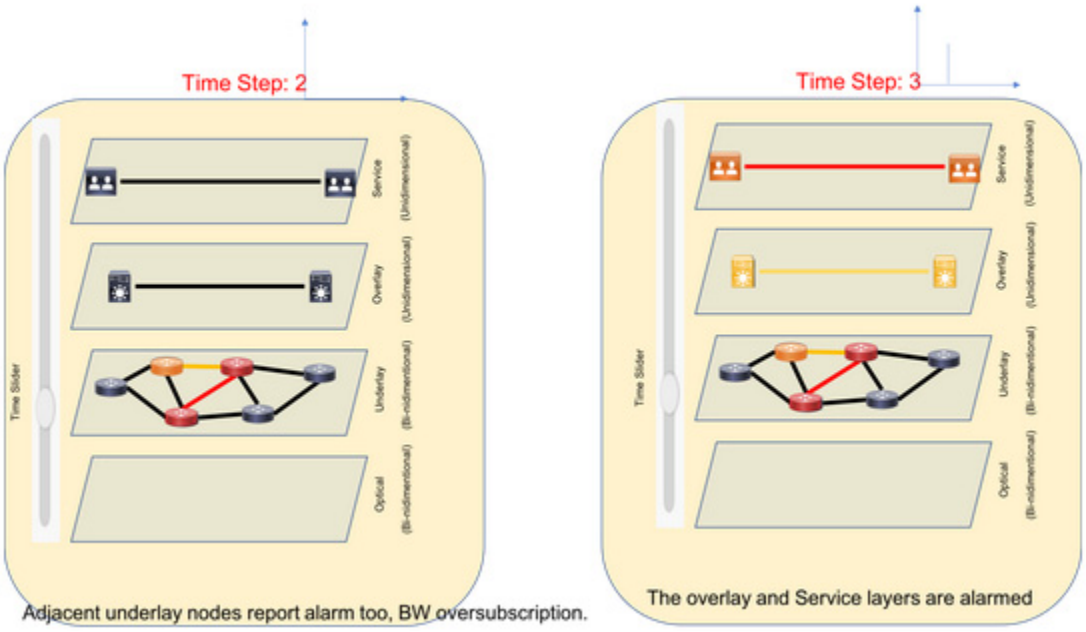

Figure 8

Figure 9 below illustrates an example heatmap that shows red for the underlay and orange for the dependency layers.

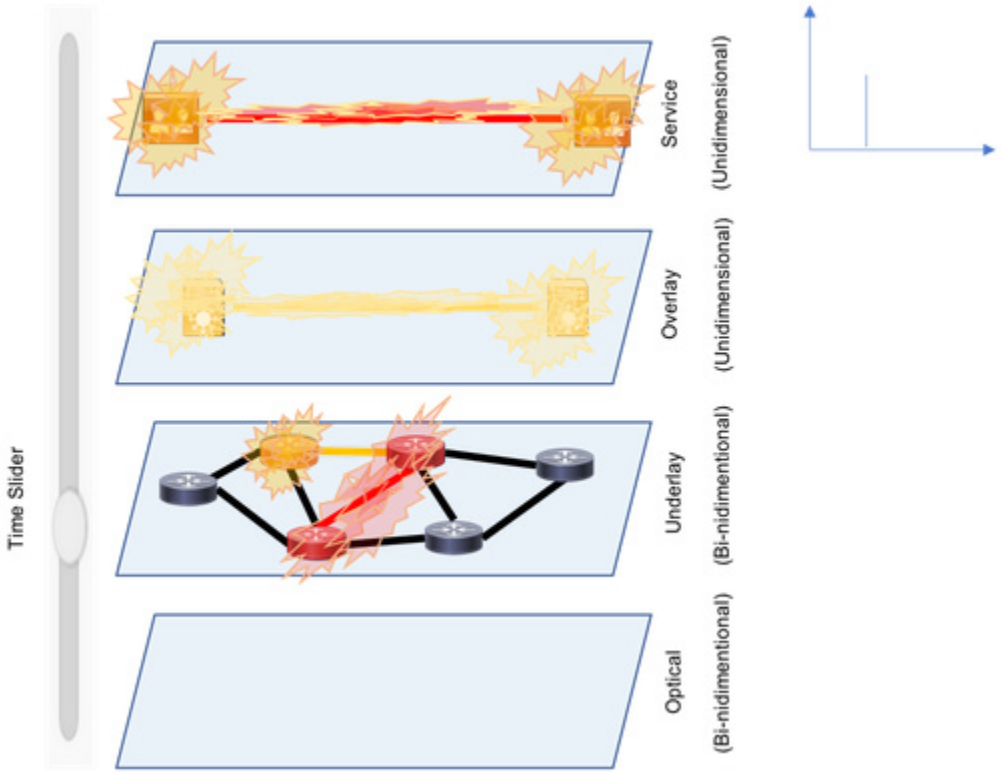

Figure 9

Figure 10 below illustrates how service layer alarms may be present at step 4. When the path converges at the underlay layer, services may be recovered and return to normal. This may occur while the alarms related to the circuit that failed are still present.

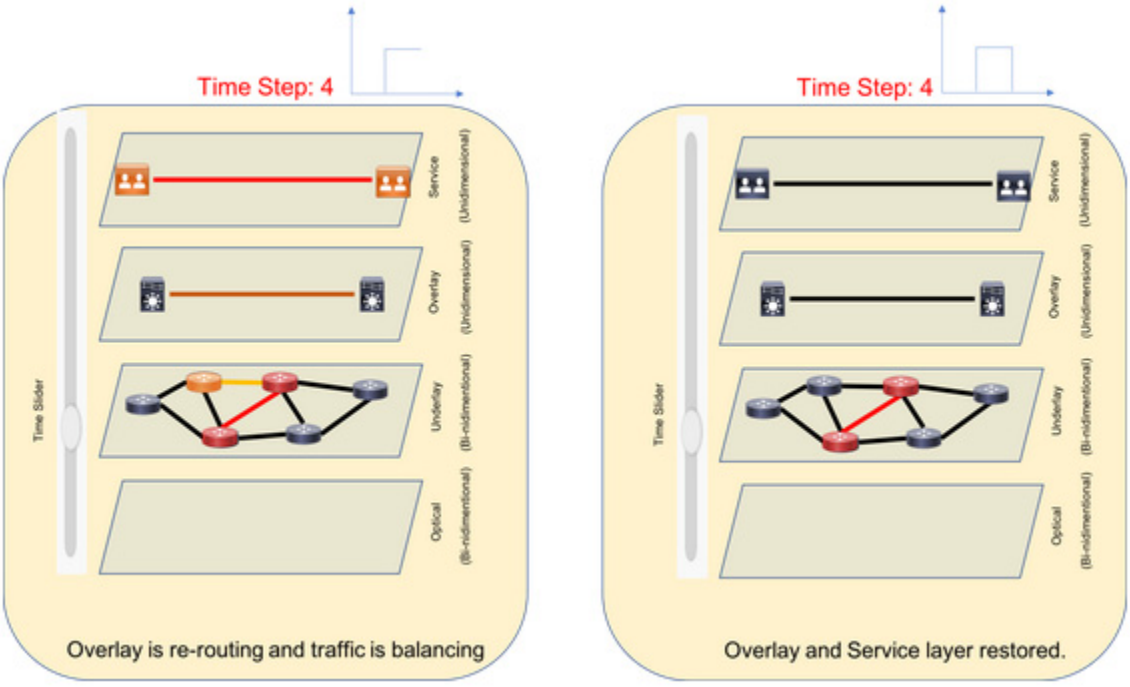

Figure 10

In another embodiment, any existing failure prediction models applied at each layer as well as the DMT may be leveraged to predict the issues/alarms at upper layer and trigger the necessary course of action.

Techniques described herein provide multi-layer configuration and operation data collection and clustering. Clustering the multi-layer data may create the layer DMT in a two-fold learning phase. A bottom-up DMT may be created from Layer 1. The network level alarm/logs fed into the ML agent along with the DMT. Identification and clustering of multi-layer alarms, telemetry, and data may be traced to an atomic event. Said cluster may be mapped to a unique atomic event. The cluster may be vectored against a singular intent.

Moreover, report visualization and reporting as described herein may involve providing a multilayer heatmap in which high issue identification may be tracked to its appropriate layer. A multilayer based time slider may enable a heatmap based time machine. Each layer may have a different dimension (e.g., 1D, 2D, 3D). The relevant alarm that is

the root cause for the incident may be prioritized and/or visualized, and necessary corrective action may be taken.

In summary, techniques are described herein for a two-fold process where a first ML model is engaged in building a DMT based on the data from the network (network observations). An IGP database, service graph information, and iOAM data is leveraged to feed into the ML model to build the DMT. The DMT is built with a per physical element precision (fiber optical cable in each direction). In other words, the database is built in a way to enable identification of the list of (connected/non-connected) services and protocols that potentially relies on the fiber cable. In the second fold, the ML model uses the dependency mapping built in the previous phase to identify the potential cause for an issue and prioritize the relevant alarms. In addition, it may be tied together with existing failure prediction mechanisms for any layer that in turn will be used with the above database to prioritize the alarm/notification to an operator OSS and take any necessary pre-emptive action on the above layers.