NMOP
Internet-Draft
Intended status: Informational
Expires: 25 May 2026

T. Graf
W. Du
Swisscom
P. Francois
A. Huang-Feng
INSA-Lyon

21 November 2025

A Framework for a Network Anomaly Detection Architecture
draft-ietf-nmop-network-anomaly-architecture-06

Abstract

This document describes the motivation and architecture of a Network Anomaly Detection Framework and the relationship to other documents describing network Symptom semantics and network incident lifecycle.

The described architecture for detecting IP network service interruption is designed to be generic applicable and extensible. Different applications are described and examples are referenced with open-source running code.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Operations and Management Area Working Group Working Group mailing list (nmop@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/nmop/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-wg-nmop/draft-ietf-nmop-network-anomaly-architecture/ .

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 25 May 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Today's highly virtualized large scale IP networks are a challenge for network operation to monitor due to its vast number of dependencies. Humans are no longer capable to verify manually all the dependencies end to end in a timely manner.

IP networks are the backbone of today's society. We individually depend on networks fulfilling the purpose of forwarding IP packets from a point A to a point B at any time of the day. A loss of such connectivity for a short period of time has today manyfold implications that can range from minor to severe. An interruption can lead to being unable to browse the web, watch a soccer game, access the company intranet or, even in life threatening situations, no longer being able to reach emergency services. Further, a congestion in the network leading to delayed packet forwarding can lead to severe repercussions on real-time applications.

Networks are generally deterministic. However, the usage of networks are only somewhat. Humans, as in a large group of people, are somehow predictable. There are time of the day patterns in terms of when we are eating, sleeping, working or leisure. And these patterns are potentially changing depending on age, profession and cultural background.

1.1. Motivation

When operational or configurational changes in connectivity services are happening, it is crucial for network operators to detect interruptions within the network faster than the users utilizing the connectivity services.

In order to achieve this objective, automation in network monitoring is required. The amount of people operating the network are today simply outnumbered by the amount of people utilizing connectivity services.

This automation needs to monitor network changes holistically by supervising all 3 network planes simultaneously for a given connectivity service on the OSI (Open Systems Interconnection) layer 3. The monitoring system needs to detect whether configurational or operational State changes, an interface was shutdown by an operator versus an interface State went down due to loss of signal on the optical layer and wherever it disrupted the service, e.g. the received packets from customers are no longer forwarded to the desired destination, or not.

Management plane relates to network node entities. Where control plane in turn propagates a subset o the management plane entities, the path reachability, to its neighboring network nodes accross the network. The forwarding plane requires a previously converged network topology and received packets to export metrics.

A State change in control and management plane which are related to each other indicate a network topology State change while a State change in the forwarding plane describes how the packets are being forwarded. In other words, control and management plane State changes can be attributed to network topology State changes whereas forwarding plane State changes are related to the outcome of these network topology State changes.

Since changes in networks are happening all the time due to the vast number of dependencies, most of the changes are not negatively affecting the end to end connectivity due to redundancies in networks, a scoring system is needed to indicate how disruptive the change is considered. The scoring system needs to take into account the amount of transport sessions, the amount of affected flows and whether the detected interruptions are usual or exceptional.

1.2. Scope

Such objectives can be achieved by applying checks on network modeled time series data that contains semantics describing their dependencies across network planes. These checks can be based on domain knowledge or using outlier detection techniques. Domain-knowledge-based techniques applies the expertise of network engineers operating a network to understand whether there is an issue impacting the customer or not. On the other hand, outlier detection techniques identify measurements that deviate significantly from the norm and therefore are considered anomalous.

The described scope does not take the connectivity service intent into account nor does it verify whether the intent is being achieved all the time. Changes to the service intent causing service disruptions are therefore considered service disruptions. On monitoring systems which take the intent into account, this is considered as intended.

Also out of scope of this document are a gradual degredation of a connectivity service over a long period of time. An example would be optical fiber degredation which lead to malform packets on IP layer and therefore increases packet drops steadily. Outlier detection techniques can be applied here as well but instead of taking the network model, the component type and characterstics would be taken into context.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.1. Terminology

This document defines the following terms:

Outlier Detection: Is a systematic approach to identify rare data points deviating significantly from the majority.

Service Disruption Detection (SDD): The process of detecting a service degradation by discovering outliers in network monitoring data.

Service Disruption Detection System (SDDS): A system allowing to perform SDD.

Rules: Refers to rules defined by domain experts or artificial intelligence in context of detection strategies. See Section 3.5.1.1 for details on domain expert rules.

Additionally it makes use of the terms defined in [I-D.ietf-nmop-terminology], [I-D.ietf-nmop-network-anomaly-lifecycle] and [RFC8969].

The following terms are used as defined in [I-D.ietf-nmop-terminology] :

* Resource
* Event
* State
* Relevance
* Problem
* Symptom
* Alarm

Figure 2 in Section 3 of [I-D.ietf-nmop-terminology] shows characteristics of observed operational network telemetry metrics.

Figure 4 in Section 3 of [I-D.ietf-nmop-terminology] shows relationships between, state, relevant state, problem, symptom, cause and alarm.

Figure 5 in Section 3 of [I-D.ietf-nmop-terminology] shows relationships between problem, symptom and cause.

The following terms are used as defined in [I-D.ietf-nmop-network-anomaly-lifecycle] :

* False Positive
* False Negative
* Confidence Score
* Concern Score

The following terms are used as defined in [RFC8969] :

* Service Model

2.2. Outlier Detection

Outlier Detection, also known as anomaly detection, describes a systematic approach to identify rare data points deviating significantly from the majority. Outliers can manifest as single data point or as a sequence of data points. There are multiple ways in general to classify anomalies, but for the context of this document, the following three classes are taken into account:

Global outliers: An outlier is considered "global" if its behavior is outside the entirety of the considered data set. For example, if the average dropped packet count is between 0 and 10 per minute and, in a small time-window, the value gets to 1000, this data point is considered a global anomaly.

Contextual outliers: An outlier is considered "contextual" if its behavior is within a normal (expected) range, but it would not be expected based on some context. Context can be defined as a function of multiple parameters, such as time, location, etc. An example of a contextual outlier is when the forwarded packet volume overnight reaches levels which might be totally normal for the daytime, but anomalous and unexpected for the nighttime.

Collective outliers: An outlier is considered "collective" if the behavior of each single data point that are part of the anomaly are within expected ranges (so they are not anomalous in either a contextual or a global sense), but the group, taking all the data points together, is. Note that the group can be made within a single time series (a sequence of data points is anomalous) or across multiple types of metrics (e.g. if looking at two metrics together, the combined behavior turns out to be anomalous). In Network Telemetry time series, one way this can manifest is that the amount of network paths and interface State changes matches the time range when the forwarded packet volume decreases as a group.

For each outlier a Confidence and a Concern Score between 0 and 1 is being calculated. The higher the Confidence Score value, the higher the probability that the observed data point is an outlier. The higher the Concern Score value, the higher the probability that observed outlier is impacting the forwarding of the customer packets negatively. Combined together raising the Relevance of the observed events. Anomaly detection: A survey [VAP09] provides and discusses an overview on different anomaly detection techniques and the outlier detection approach adopted by each.

2.3. Knowledge Based Detection

Knowledge-based anomaly detection, a superset of rule-based anomaly detection and a subset of semantic-based, Knowledge-based anomaly detection: Survey, challenges, and future directions [ASNL25], is a technique used to identify anomalies or outliers by comparing them against predefined rules or patterns. This approach relies on the use of domain-specific knowledge to set standards, thresholds, or rules for what is considered "normal" behavior. Traditionally, these rules are established manually by a knowledgeable network engineer. Forward-looking, these rules can be expressed using human and machine

readable network protocol derived Symptoms and patterns defined in ontologies.

Additionally, in the context of network anomaly detection, the knowledge-based approach works hand in hand with the deterministic understanding of the network, which is reflected in network modeling. Components are organized into three network planes: the Management Plane, the Control Plane, and the Forwarding Plane [RFC9232]. A component can relate to a physical, virtual, or configurational entity, or to a sum of packets belonging to a flow being forwarded in a network.

Such relationships can be modelled in Service and Infrastructure Maps (SIMAP) to automate that process. [I-D.ietf-nmop-simap-concept] defines the concepts for the SIMAP and [I-D.havel-nmop-digital-map] defines an application of the SIMAP to network topologies.

These relationships can also be modeled in Knowledge Graphs Section 5 of [I-D.mackey-nmop-kg-for-netops] using semantic triples [W3C-RDF-concept-triples], where with ontologies, due to its declarative form, those semantic triples are machine and human readable. See Section 2.5.2 as an example for an ontology describing symptoms.

2.4. Machine Learning

Machine learning is commonly used for detecting outliers or anomalies. Typically, unsupervised learning is widely recognized for its applicability, given the inherent characteristics of network data. See [VAP09]. Although machine learning requires a sizeable amount of high-quality data and considerable advanced training, the advantages it offers make these requirements worthwhile. The power of this approach lies in its generalizability, robustness, ability to simplify the fine-tuning process, and most importantly, its capability to identify anomaly patterns that might go unnoticed to the human observer.

2.5. Data Mesh

The Data Mesh [Deh22] Architecture distinguishes between operational and analytical data. Operational data refers to collected data from operational systems. While analytical data refers to insights gained from operational data.

2.5.1. Operational Network Data

In terms of network observability, semantics of operational network metrics are defined by IETF and are categorized as described in the Network Telemetry Framework [RFC9232] in the following three different network planes:

Management Plane: Time series data describing the State changes and statistics of a network node and its Resources. For example, Interface State and statistics modeled in ietf-interfaces.yang [RFC8343].

Control Plane: Time series data describing the State and State changes of network reachability. For example, BGP VPNv6 unicast updates and withdrawals exported in BGP Monitoring Protocol (BMP) [RFC7854] and modeled in BGP [RFC4364].

Forwarding Plane: Time series data describing the forwarding behavior of packets and its data-plane context. For example, dropped packet count modelled in IPFIX entity forwardingStatus(IE89) [RFC7270] and packetDeltaCount(IE2) [RFC5102] and exported with IPFIX [RFC7011].

2.5.2. Analytical Observed Symptoms

The Service Disruption Detection process takes operational network data as input and generates analytical metrics describing Symptoms and outlier pattern of the connectivity service disruption.

The observed Symptoms are categorized into semantic triples [W3C-RDF-concept-triples]: action, reason, trigger. The object is the action, describing the change in the network. The reason is the predicate, defining why this change occured and the subject is the trigger, which defines what triggered that change.

Symptom definitions are described in Section 3 of [I-D.ietf-nmop-network-anomaly-semantics] and outlier pattern semantics in Section 8 of [I-D.ietf-nmop-network-anomaly-lifecycle]. Both are expressed in YANG Service Models.

However the semantic could also be expressed with the Semantic Web Technology Stack in RDF, RDFS and OWL definitions as described in Section 6 of [I-D.mackey-nmop-kg-for-netops]. Together with the ontology definitions described in Section 3 of [I-D.ietf-nmop-network-anomaly-semantics], a Knowledge Graph can be created describing the relationship between the network state and the observed Symptom.

3. Elements of the Architecture

The service disruption detection system architecture is aimed at detecting service disruptions and is built upon multiple components, for which design choices need to be made. In this section, we describe the main components of the architecture, and delve into considerations to be made when designing such componenents in an implementation.

The system architecture is illustrated in Figure 1 and its main components are described in the following subsections.

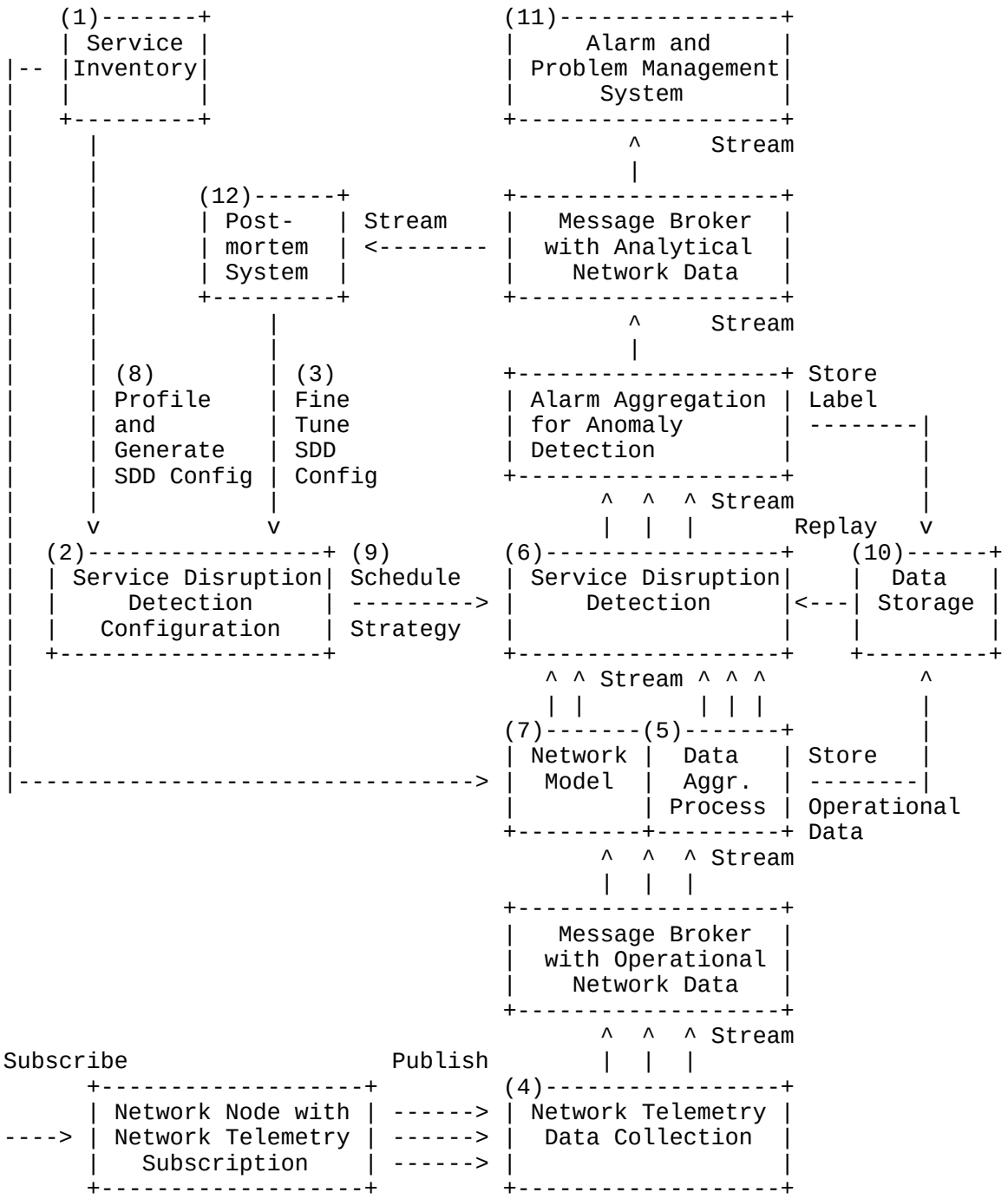
(1) Service Inventory
(2) Service Disruption Detection Configuration
(3) Fine Tune SDD Config
(4) Network Telemetry Data Collection
(5) Data Aggr. Process
(6) Service Disruption Detection
(7) Network Model
(8) Profile and Generate SDD Config
(9) Schedule Strategy
(10) Data Storage
(11) Alarm and Problem Management System
(12) Post-mortem System
Message Broker with Analytical Network Data
Alarm Aggregation for Anomaly Detection
Message Broker with Operational Network Data
Store Label
Store Operational Data
Stream
Replay
Subscribe
Publish
Network Node with Network Telemetry Subscription

Figure 1: Service Disruption Detection System Architecture

3.1. Service Inventory

A service inventory, (1) in Figure 1, is used to obtain a list of the connectivity services for which Anomaly Detection is to be performed. A service profiling process may be executed on the operational network data of the service in order to define a configuration of the service disruption detection approach and parameters to be used.

3.2. Service Disruption Detection Configuration

Based on this service list and potential preliminary service profiling, a configuration of the Service Disruption Detection, (2) in Figure 1, is produced. It defines the set of approaches that need to be applied to perform SDD, as well as parameters, grouped in templates, that are to be set when executing the algorithms performing SDD per se.

As the service lives on, the configuration may be adapted, (3) in Figure 1, as a result of an evolution of the profiling being performed. Postmortem analysis are produced as a result of Events impacting the service, or the occurrence of false positives raised by the Alarm system. These postmortem analysis can improve the deployed profiles parameters and creation of new customer profiles. See upcoming section Section 3.5.1.3 for details on profiling.

3.3. Operational Data Collection

Collection of network monitoring data, (4) in Figure 1, involves the management of the subscriptions to network telemetry on nodes of the network, and the configuration of the collection infrastructure to receive the monitoring data produced by the network.

The monitoring data produced by the collection infrastructure is then streamed through a message broker system, for further processing.

Networks tend to produce extremely large amounts of monitoring data. To preserve scaling and reduce costs, decisions need to be made on the duration of retention of such data in storage, and at which level of storage they need to be kept. A retention time need to be set on the raw data produced by the collection system, in accordance to their utility for further used. This aspect will be elaborated in further sections.

3.4. Operational Data Aggregation

Aggregation, (5) in Figure 1, is the process of producing data sets based on collected network monitoring data upon which detection of a service disruption can be performed by filtering or aggregating.

Pre-processing of collected network monitoring data is usually performed to reduce input for the Service Disruption Detection component since not all metrics are relevant for this use case. This can be achieved in multiple ways, depending on the architecture of the SDD component. As an example, the granularity or cardinality at which forwarding plane data is produced by the network may be too high for the SDD algorithms, and instead be aggregated into a coarser dimension for SDD execution.

A retention time for the operational data needs to be decided on Aggregated data and should reflect the expected further use. As example, the retention time must be set in accordance with the replay ability requirement discussed in Section 3.8.

3.5. Service Disruption Detection

Service Disruption Detection processes, (6) in Figure 1, decide whether a service might be degraded to the point where network operation needs to be alerted of an ongoing Problem within the network.

Two key aspects need to be considered when designing the SDD component. First, the way the data is being processed needs to be carefully designed, as networks typically produce extremely large amounts of data which may hinder the scalability of the architecture. Second, the algorithms used to make a decision to alert the operator need to be designed in such a way that the operator can trust that a targeted Service Disruption will be detected (no false negatives), while not spamming the operator with Alarms that do not reflect an actual issue within the network (false positives) leading to Alarm fatigue.

Two approaches are typically followed to present the data to the SDD system. Classically, the aggregated data can be stored in a database that is polled at regular intervals by the SDD component for decision making. Alternatively, a streaming approach can be followed so as to process the data while they are being consumed from the collection component.

For SDD per-se, two families of algorithms can be decided upon. First, knowledge based detection approaches can be used, mimicking the process that human operators follow when looking at the data. Second, Machine Learning based approaches to detect outliers based from prior trained operational network data.

3.5.1. Knowledge Based

Knowledge based detection is comprised of several types of knowledge sources such as domain knowledge from network engineers Section 3.5.1.1 understanding the mechanics of network protocols and their implications, knowledge from relationships in the network topology Section 3.5.1.2, knowledge derived from Section 3.5.1.3 where customer, human behavioral related aspects are taken into context and finally in Section 3.5.1.4 a combination of that knowledge is being applied.

3.5.1.1. Expert Rules

Some input to SDD is made of established knowledge from network engineers. This expertise can be used for both Service Disruption Detection Configuration or SDD, (2) and (6) in Figure 1 respectively. For example, sudden spikes in drop counters from the forwarding plane are likely to be attributed to changes in the routing topology. Or, drops in the fowarding plane can manifest in an increase of flow counts in the forwarding plane due to the implied congestion and re-establishment of application transport sessions. These network behaviours are typically sourced from the experience of operating a network infrastructure by human operators, and can be used by an SDD engine to trigger alerts.

3.5.1.2. Network Modeling

Some input to SDD is made of established knowledge of the network, (7) in Figure 1, that is unrelated to the dimensions according to which outlier detection is performed. For example, the knowledge of the network infrastructure may be required to perform some service disruption detection. Such data need to be rendered accessible and updatable for use by SDD. They may come from inventories, or automated gathering of data from the network itself.

3.5.1.3. Data Profiling

As expert rules cannot be crafted specifically for each customer because each customer has a different usage pattern, they need to be defined according to pre-established service profiles, (8) in Figure 1. Processing of monitoring data can be performed with machine learning methods in order to identify and group patterns into clusters and associate clusters with profiles. External knowledge on customer types can also help in associating clusters with profiles.

3.5.1.4. Detection Strategies

For a profile, a set of strategies is defined. Each strategy captures one approach to look at the data (as a human operator does) to observe if an abnormal situation is arising. Strategies can use both expert rule-based algorithms, as described in Section 3.5.1.1, or outlier detection algorithms, as explained in Section 2.2. Thus, a strategy defined as a combination of expert rule-based algorithms or outlier detection algorithms that together trigger an alarm when a disruption occur.

When one of the strategies applied for a profile detects a concerning global outlier or collective outlier, an Alarm MUST be raised.

Depending on the implementation of the architecture, a scheduler may be needed in order to orchestrate the evaluation of the Alarm levels for each strategy applied for a profile, for all service instances associated with such profile, as illustrated in (9) from Figure 1.

3.5.2. Storage

Storage, (10) in Figure 1, may be required to execute SDD, as some algorithms may be relying on historical (aggregated) monitoring data in order to detect anomalies. The cardinality,granularity and retention time of historical data should be carefully considered to avoid slow and costly retrieval of this information if required for SDD analysis.

3.6. Alarm

When the SDD component decides that a service is undergoing a disruption, an aggregated relevant-state change notification, taking the output of multiple Service Disruption Detection processes into account, MUST be sent to the Alarm and Problem management system as shown in Figure 4 in Section 3 of [I-D.ietf-nmop-terminology] and (11) in Figure 1. Multiple practical aspects need to be taken into account in this component.

When the issue lasts longer than the interval at which the SDD component runs, the relevant-state change mechanism should not create multiple notifications to the operator, so as to not overwhelm the management of the issue. However, the information provided along with the Alarm should be kept up to date during the full duration of the issue.

3.7. Postmortem

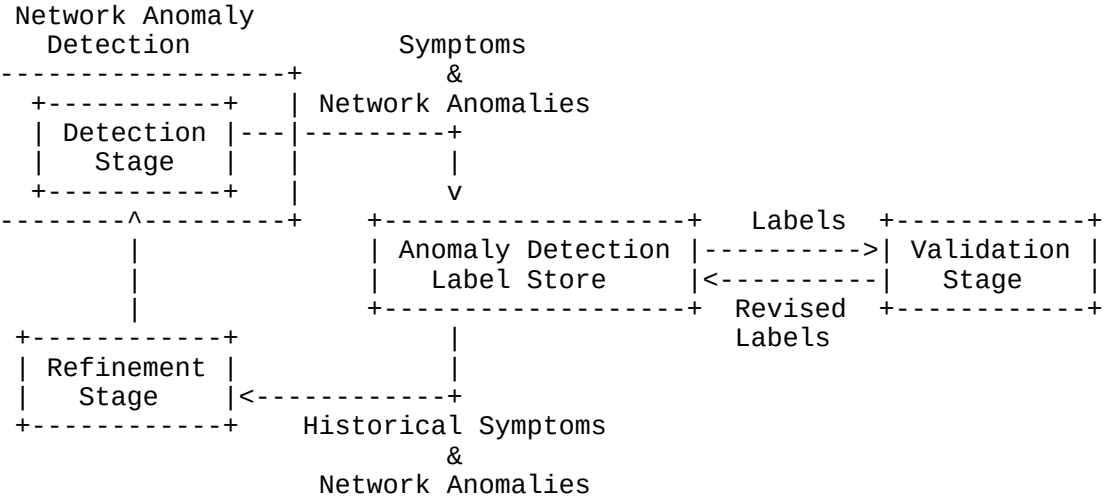

Figure 2: Anomaly Detection Refinement Lifecycle

Validation and refinement are performed during Postmortem analysis, (12) in Figure 1.

From an Anomaly Detection Lifecycle point of view, as described in [I-D.ietf-nmop-network-anomaly-lifecycle], the Service Disruption Detection Configuration evolves over time, iteratively, looping over three main phases: detection, validation and refinement.

The Detection phase produces the Alarms that are sent to the Alarm and Problem Management System and at the same time it stores the network anomaly and Symptom labels into the Label Store. This enables network engineers to review the labels to validate and edit them as needed.

The Validation stage is typically performed by network engineers reviewing the results of the detection and indicating which Symptoms and network anomalies have been useful for the identification of

Problems in the network. The original labels from the Service Disruption Detection are analyzed and an updated set of more accurate labels is provided back to the label store for version-control.

The resulting labels will be then provided back into the Network Anomaly Detection via its refinement capabilities: the refinement is about the update of the Service Disruption Detection configuration in order to improve the results of the detection (e.g. false positives, false negatives, accuracy of the boundaries, etc.).

3.8. Replaying

When a service disruption has been detected, it is essential for the human operator to be able to analyze the data which led to the raising of an Alarm. It is thus important that a SDDS preserves both the data which led to the creation of the Alarm as well as human understandable information on why the data led to the raising of an Alarm.

In early stages of operations or when experimenting with a SDDS, it is common that the parameters used for SDD are to be fined tuned. This process is facilitated by designing the SDDS architecture in a way that allows to rerun the SDD algorithms on the same input.

Data retention, as well as its level, need to be defined in order not to sacrifice the ability of replaying SDD execution for the sake of improving its accuracy.

4. Implementation Status

Note to the RFC-Editor: Please remove this section before publishing.

This section records the status of known implementations.

4.1. Cosmos Bright Lights

This architecture have been developed as part of a proof of concept started in September 2022 first in a dedicated network lab environment and later in December 2022 in Swisscom production to monitor a limited amount of 16 L3 VPN connectivity services.

At the Applied Networking Research Workshop at IRTF 117 the architecture was the first time published in the following academic paper: [Ahf23].

Since December 2022, 20 connectivity service disruptions have been monitored and 52 false positives due to time series database temporarily not being real-time and missing traffic profiling,

comparing to previous week was not applicable, occurred. Out of 20 connectivity service disruptions 6 parameters where monitored and 3 times 1, 8 times 2, 6 times 3, 2 times 4 parameters recognized the service disruption.

A real-time streaming based version has been deployed in Swisscom production as a proof of concept in June 2024 monitoring approximate >13’000 L3 VPN’s concurrently. Improved profiling capabilities are currently under development.

5. Security Considerations

Security of the retained data. Compromised data could reveal sensitive information; could prevent valid alarms from being raised; or could cause false alarms.

6. Contributors

The authors would like to thank Alex Huang Feng, Ahmed Elhassany and Vincenzo Riccobene for their valuable contribution.

7. Acknowledgements

The authors would like to thank Qin Wu, Ignacio Dominguez Martinez-Casanueva, Adrian Farrel, Reshad Rahman, Ruediger Geib, Paul Aitken and Yannick Buchs for their review and valuable comments.

8. References

8.1. Normative References

[I-D.havel-nmop-digital-map]
Havel, O., Claise, B., de Dios, O. G., Elhassany, A., and T. Graf, "Modeling the Digital Map based on RFC 8345: Sharing Experience and Perspectives", Work in Progress, Internet-Draft, draft-havel-nmop-digital-map-02, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-havel-nmop-digital-map-02>.

[I-D.ietf-nmop-network-anomaly-lifecycle]
Riccobene, V., Graf, T., Du, W., and A. H. Feng, "An Experiment: Network Anomaly Lifecycle", Work in Progress, Internet-Draft, draft-ietf-nmop-network-anomaly-lifecycle-03, 8 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-network-anomaly-lifecycle-03>.

[I-D.ietf-nmop-network-anomaly-semantics]
Graf, T., Du, W., Feng, A. H., and V. Riccobene, "Semantic Metadata Annotation for Network Anomaly Detection", Work in Progress, Internet-Draft, draft-ietf-nmop-network-anomaly-semantics-03, 8 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-network-anomaly-semantics-03>.

[I-D.ietf-nmop-simap-concept]
Havel, O., Claise, B., de Dios, O. G., and T. Graf, "SIMAP: Concept, Requirements, and Use Cases", Work in Progress, Internet-Draft, draft-ietf-nmop-simap-concept-07, 18 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-simap-concept-07>.

[I-D.ietf-nmop-terminology]
Davis, N., Farrel, A., Graf, T., Wu, Q., and C. Yu, "Some Key Terms for Network Fault and Problem Management", Work in Progress, Internet-Draft, draft-ietf-nmop-terminology-23, 18 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-terminology-23>.

[I-D.mackey-nmop-kg-for-netops]
Mackey, M., Claise, B., Graf, T., Keller, H., Voyer, D., Lucente, P., and I. D. Martinez-Casanueva, "Knowledge Graph Framework for Network Operations", Work in Progress, Internet-Draft, draft-mackey-nmop-kg-for-netops-03, 2 September 2025, <https://datatracker.ietf.org/doc/html/draft-mackey-nmop-kg-for-netops-03>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8969] Wu, Q., Ed., Boucadair, M., Ed., Lopez, D., Xie, C., and L. Geng, "A Framework for Automating Service and Network Management with YANG", RFC 8969, DOI 10.17487/RFC8969, January 2021, <https://www.rfc-editor.org/info/rfc8969>.

[RFC9232] Song, H., Qin, F., Martinez-Julia, P., Ciavaglia, L., and A. Wang, "Network Telemetry Framework", RFC 9232, DOI 10.17487/RFC9232, May 2022, <https://www.rfc-editor.org/info/rfc9232>.

8.2. Informative References

[Ahf23] Huang Feng, A., "Daisy: Practical Anomaly Detection in large BGP/MPLS and BGP/SRv6 VPN Networks", IETF 117, Applied Networking Research Workshop, DOI 10.1145/3606464.3606470, July 2023, <https://hal.science/hal-04307611>.

[ASNL25] Qadir Khan, A., El Jaouhari, S., Tamani, N., and L. Mroueh, "Knowledge-based anomaly detection: Survey, challenges, and future directions", DOI 10.1016/j.engappai.2024.108996, May 2025, <https://hal.science/hal-05055886>.

[Deh22] Dehghani, Z., "Data Mesh", O'Reilly Media, ISBN 9781492092391, March 2022, <https://www.oreilly.com/library/view/data-mesh/9781492092384/>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC5102] Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, "Information Model for IP Flow Information Export", RFC 5102, DOI 10.17487/RFC5102, January 2008, <https://www.rfc-editor.org/info/rfc5102>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/info/rfc7011>.

[RFC7270] Yourtchenko, A., Aitken, P., and B. Claise, "Cisco-Specific Information Elements Reused in IP Flow Information Export (IPFIX)", RFC 7270, DOI 10.17487/RFC7270, June 2014, <https://www.rfc-editor.org/info/rfc7270>.

[RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP Monitoring Protocol (BMP)", RFC 7854, DOI 10.17487/RFC7854, June 2016, <https://www.rfc-editor.org/info/rfc7854>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[VAP09] Chandola, V., Banerjee, A., and V. Kumar, "Anomaly detection: A survey", ACM Computing Surveys 41, DOI 10.1145/1541880.1541882, July 2009, <https://www.researchgate.net/publication/220565847_Anomaly_Detection_A_Survey>.

[W3C-RDF-concept-triples] Cyganiak, R., Wood, D., and M. Lanthaler, "W3C RDF concept semantic triples", W3 Consortium, February 2014, <https://www.w3.org/TR/rdf-concepts/#section-triples>.

Authors' Addresses

Thomas Graf
Swisscom
Binzring 17
CH-8045 Zurich
Switzerland
Email: thomas.graf@swisscom.com

Wanting Du
Swisscom
Binzring 17
CH-8045 Zurich
Switzerland
Email: wanting.du@swisscom.com

Pierre Francois
INSA-Lyon
Lyon
France
Email: pierre.francois@insa-lyon.fr

Alex Huang Feng
INSA-Lyon
Lyon
France
Email: alex.huang-feng@insa-lyon.fr