

A YANG Data Model for Network Incident Management

draft-ietf-nmop-network-incident-yang-06

NMOP Meeting 2025-11

Authors:

Tong Hu(hutong@cmhi.chinamobile.com)

Luis Contreras (luismiguel.contrerasmurillo@telefonica.com)

Qin Wu(bill.wu@huawei.com)

Nigel Davis (ndavis@ciena.com)

Chong Feng(Fengchonglly@gmail.com)

Contributors:

Lionel Tailhardat (lionel.tailhardat@orange.com)

Thomas Graf (thomas.graf@swisscom.com)

Zhengqiang Li (li_zhenqiang@hotmail.com)

Yanlei Zheng (zhengyanlei@chinaunicom.cn)

Yunbin Xu (xuyunbin@caict.ac.cn)

Xing Zhao (zhaoxing@caict.ac.cn)

Chaode Yu(yuchaode@huawei.com)

Document Status

- The latest version is v-06, changes compared to previous version presented in IETF 122
 - Replace Root cause with probable root cause based on Adrian and Benoit's suggestion.
 - Replace machine learning with algorithmic techniques in service impact analysis definition.
 - Update 4 new Usage examples in the Appendix to explore the interaction between OSS and Network Controller
 - Correlation between troubleshooting ticket and network incident
 - Intent based network diagnosis interaction.
 - Multi-domain fault demarcation
 - Service Complaint triggered network diagnosis
- Other editorial comments
 - Follow RFC8407bis section 3.8.3.1 IANA template
 - Follow YANG module security considerations template
 - Add reference in YANG module
 - Address comments from Aitken Paul

Root Cause vs Probable Root Cause

• Original Definition in section 2

Probable root cause:

: A factor is considered the probable root cause of a problem if removing it fully resolves an ongoing incident -- specifically regarding network or service impairments and their related consequential failures and symptoms -- and prevents the problem from recurring. Causality is defined as the implementation of an inductive process at the network or processing element level {{ITU-T-G-7710}} and as a doubt removal

process at the Incident management system level {{ITU-T-X-733}}. The inductive process is based on the distinction made between `_primary failures_` (i.e., a failure that directly indicates the fault location and initiate a repair action, such as a broken cable or a misconnection) and `_secondary failures_` (i.e., a consequential

failure, such as an upper level service that has gone down) {{ITU-T-G-7710}} Since one Fault may give rise to another Fault or Problem, a probable root cause is commonly meant to describe the original event or combination of circumstances that is the foundation of all related Faults.

Conversely, a causal factor is a contributing action that influences the outcome of an incident or event but is not the probably cause.

• New Definition in section 2

Probable root cause:

: If removing a factor completely resolves the ongoing incident (specifically, regarding network outage or service impairments and their associated subsequent failures and symptoms) and prevents the problem from recurring, then such factor is considered as a probable root cause of a problem.

: Since one Fault may give rise to another Fault or Problem, a probable root cause is commonly meant to describe the original event or combination of circumstances that is the foundation of all related Faults.

: Conversely, a causal factor is a contributing action that influences the outcome of the incident or event but is not the probable cause.

Let us know whether definition is accurate or is there any text that needs to be simplified

Replace machine learning with algorithm tech

- Original Definition in section 2

Service impact analysis: A process that uses machine learning to evaluate whether the network service has been impacted by the network incident and map network incident to one or a set of network service, which can reduce massive fault/alarms reporting, speed up troubleshooting, and assure network service performance and availability.

- New Definition in section 2

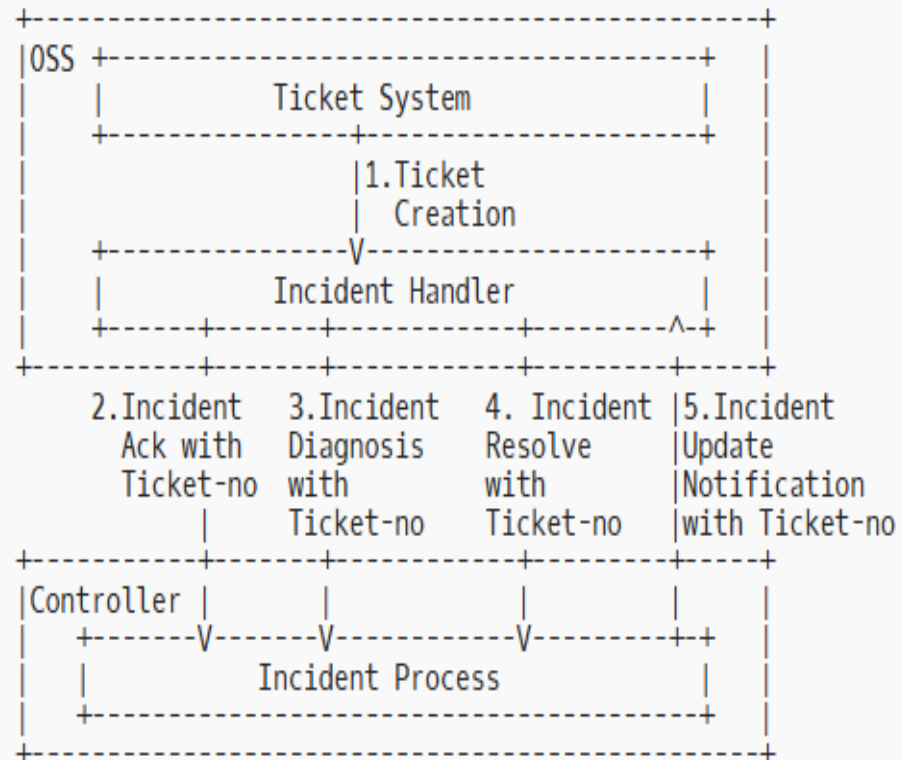
Service impact analysis: A process that uses algorithmic techniques (e.g., machine learning, automated reasoning, conformance checking, graph traversal, among others) to evaluate whether the network service has been impacted by the network incident and map network incident to one or a set of network service, which can reduce massive fault/alarms reporting, speed up troubleshooting, and assure network service performance and availability.

UC-1: Correlation with troubleshooting tickets

Use Case 1: The OSS system requires data gathering and analysis from many different tools from the controllers to demarcate whether the fault is on the User side or Network side.



Solution:
Extend RPC operations in Network Incident Model to support ticket No reporting.



```
rpcs:
+---x incident-acknowledge
| +---w input
|   +---w incident-no* incident-ref
|   +---w ticket-no? string
+---x incident-diagnose
| +---w input
|   +---w incident-no* incident-ref
|   +---w ticket-no? string
| +--ro output
|   +--ro task-id? string
+---x incident-resolve
| +---w input
|   +---w incident-no* incident-ref
|   +---w ticket-no? string

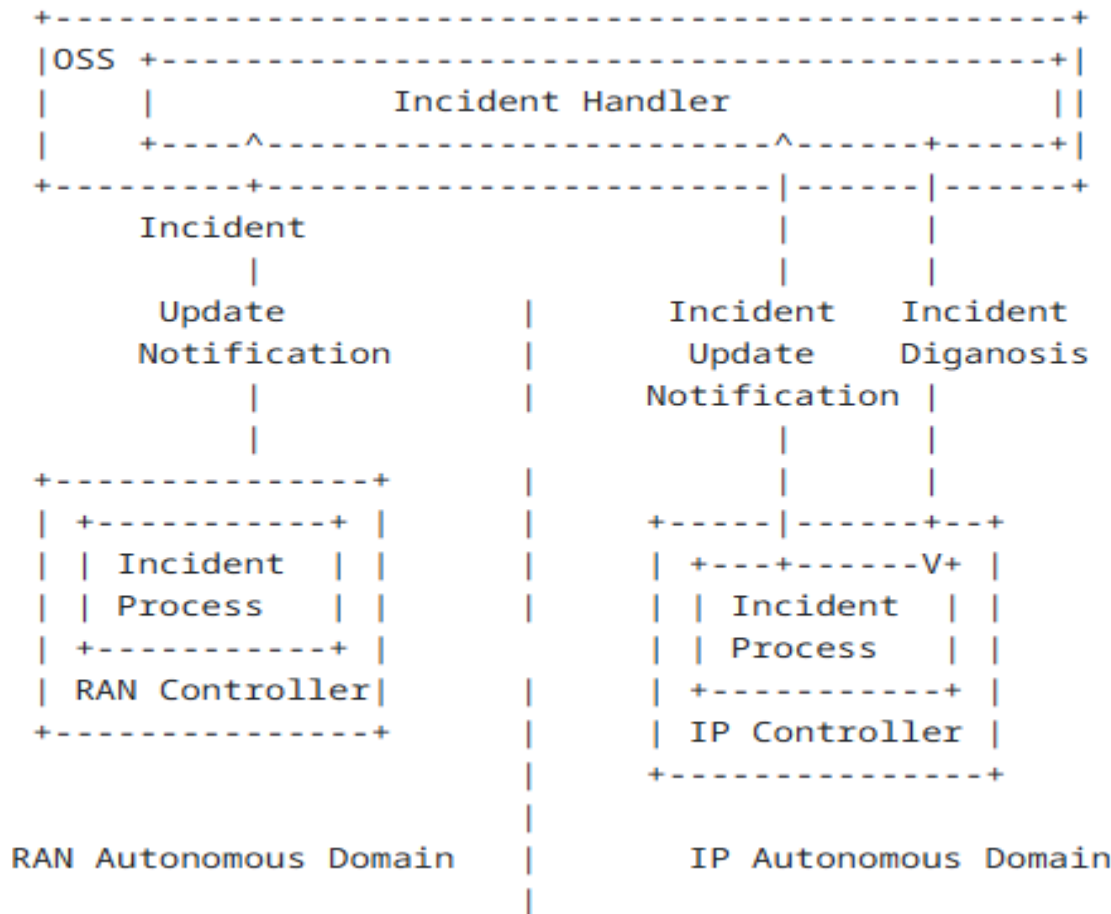
notifications:
+---n incident-notification
|   +--ro incident-no? incident-ref
|   +--ro ticket-no? string
+--
...
```

UC-2: Multi-Domain Fault Demarcation

Use Case2: The OSS system is unable to locate the fault and delegate network diagnosis to IP controller For multi-domain Demarcation

Solution:

The OSS system invokes Incident Diagnosis RPC operation In the Network Incident YANG Model to diagnose the root cause and provide repair suggestion.



```

Diagnosis Key
Parameters
{
  ticket-no, String
  incident-no, String
  occur-time, yang:date-and-time
  context? String
  related-events? leafref //List <Event>
  related-objects? leafref //List <ResourceObject>
  ....
}
  
```

Related-object can be Base Station IP address, the Related-events can be incident from RAN Domain received from Update Notification

The IP Controller correlated Network side port, Transmission path (Current Path, History Path) and Network performance along the path

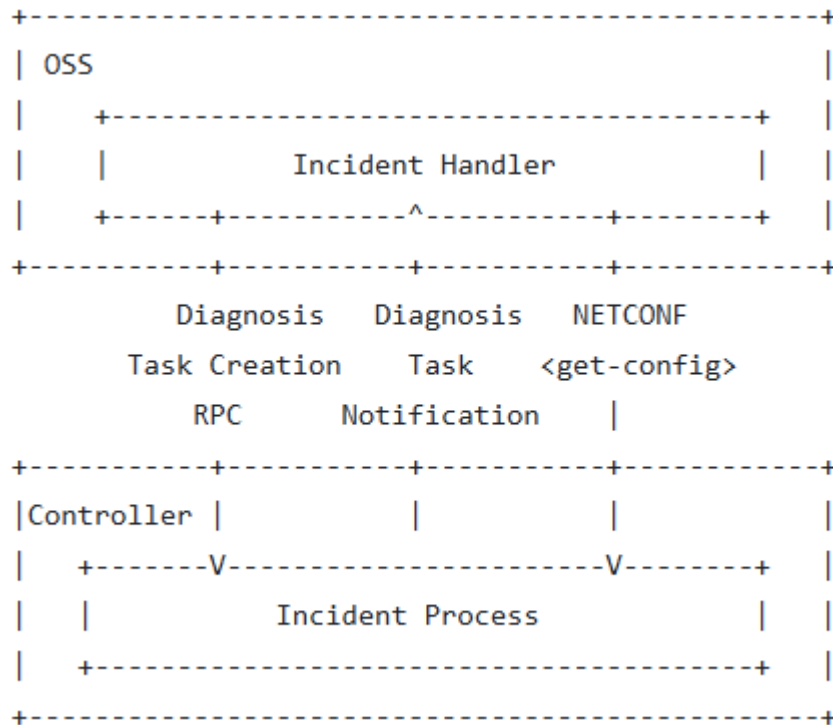
UC-3: Intent based networking with Diagnosis task list

Use Case3: The OSS system use intent based network interface to exchange the intent associated with Diagnosis task in the multiple steps interaction.



Solution:

Extend Network Incident YANG Data Model to support New RPC <Diagnosis-task-creation>, new notification <Diagnosis-task-notification> and include "incident-diagnosis-task" list.



```

+---x diagnose-task-creation
| +---w input
| | +---w incident-no? string
| | +---w ticket-no? string
| | +---w occur-time? yang:date-and-time
| | +---w context? string
| | +---w related-events
| | | +---w probable-event* []
| | | +---w type? -> ../../../../events/event/t
| | | +---w event-id? -> ../../../../events/event[t
| | +---w related-objects
| | | +---w source* [node-ref]
| | | +---w node-ref -> /nw:networks/network
| | | +---w network-ref? -> /nw:networks/network
| | | +---w resource* [name]
| | | +---w name al:resource
| +--ro output
| +--ro task-id? string

augment /inc:incidents/inc:incident:
+--ro incident-diagnosis-tasks
| +--ro incident-diagnosis-task* [task-id]
| +--ro task-id? String
| +--ro incident-no* incident-ref
| +--ro ticket-no? string
| +--ro start-time? yang:date-and-time
| +--ro end-time? yang:date-and-time
| +--ro task-state? enumeration
| +--ro diagnosis-result? enumeration
| +--ro diagnosis-result-description? String
| +--ro probable-causes leafref //List <RootCause>
...
+--ro probable-events leafref //List <Event>
...
+--ro repair-advice
| +--ro state enumeration // Incident states such as Creation, Update, Clear
...
  
```

OSS creates diagnosis task in the controller and controller allocates task-id And return back to the OSS system without task details

OSS obtain diagnosis task details via either NETCONF <get-config> on model extension or Controller reports to OSS system proactively when task gets done.

Next Step

- Address latest comments raised by Chairs
- Ready for WGLC