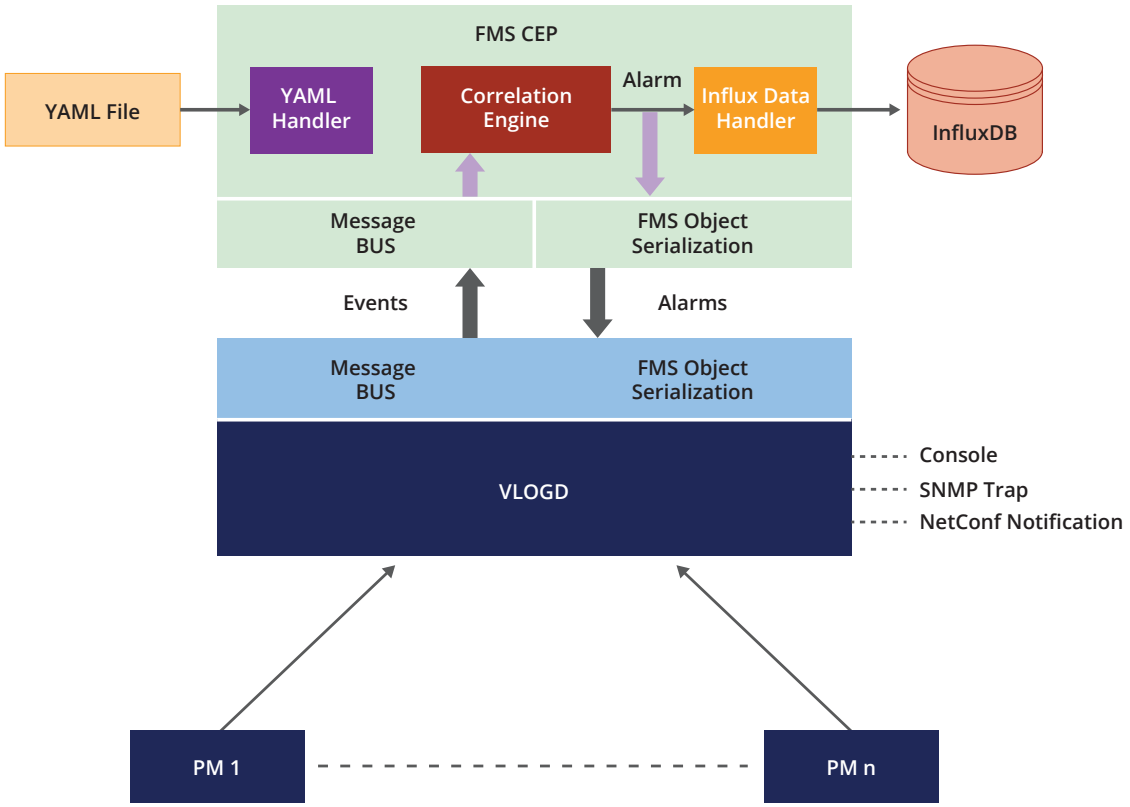
OcNOS® Fault Management System

Application Note

OcNOS 3.0 introduces a dedicated Fault Management System which can detect, filter and correlate faults generated by OcNOS powered devices and raise appropriate alarms.

The Fault Management System includes a message bus based on publish and subscribe design pattern. The different components of OcNOS register with the message bus either as publisher or subscriber of the faults. A YAML based configuration file is used to configure the action to be taken by FMS when a certain event occurs. A correlation engine examines the events, decides the relevant correlation algorithm to be used and generates the respective alarms. These alarms are then stored in an Influx database.



The Fault Management System provides unprecedented visibility about the health of a network element. Just with the help of few commands the network administrator can get all the alarm generated by the device over a specific duration of time, statistics about the alarms like total number of alarms raised and their severity, the list of current active alarms on a device etc. These alarms can also be exported to an external network wide Fault Management System like Cisco NSO with the help of an Adapter.

Alarm Definition

OcNOS consists of a number of Protocol Modules which run as separate daemons. The FMS gives provision to Protocol Modules to provide alarm definition using a configuration file written using YAML. The FMS parses these configuration files during startup. The alarm definition consists of following information:

```
#-------Template-------
#- Event_Group:
#   - ALARM_ID:                     # Integer number identifying alarm
#     EVENT:                        # Event name(oper_log)
#     GENERALIZED_EVENT_NAME:       # Event name for the Generalization Event Group
#     ALARM_DESC:                   # Alarm string which will be generated
#     CORRELATION_TYPE:             # Correlation logic type(0:No-Correlation,
#1:Generalization, 2:Timebound, 3:Counting, 4:Compression, 5:Drop-Event)
#     GENERALISED_CORRELATION_TYPE  # Correlation type, in which generalised event
will be sent
#     CORRELATION_COUNTER:          # Counter value that will be considered during
counting logic to raise alarm
#     CORRELATION_TIMER_DURATION:   # Timer duration to be considered for time bound
logic
#     CORRELATION_SEVERITY:         # Alarm Severity(1:Emergency, 2:Alert,
3:Critical, 4:Error, 5:Warning, 6:Notification, 7:Informational, 8:Debugging, 9:Cli)
#     POSITION:                     # List of positions where dynamic values present
#       STR_POSITION_1_EVENT_1:     # First position of the dynamic value in the
event
#     SNMP_TRAP:                    # SNMP TRAP (true(1) or false(0))
#     SNMP_OID:                     # OID for SNMP TRAP
#     NETCONF_NOTIFICATION:         # Netconf Notification (true(1) or false(0))
#     CLEAR_ALARM:                  # Clear Alarm (oper_log enum, Status for Alarm
will be made In-active if this event is received)
#     SNMP_TRAP_CLEAR:              # true(1) or false(0, if CLEAR_ALARM is null then
SNMP_TRAP_CLEAR will be null)
#     SNMP_CLEAR_OID:               # OID for SNMP TRAP CLEAR
#     NETCONF_CLEAR_NOTIFICATION:   # Clear Netconf Notification information
```

When FMS receives an event, it takes the action as specified by corresponding Alarm definition file.

Alarm Correlation

The correlation logic takes in multiple occurrences of the same event, examines them for duplicate information, removes redundancies and reports them as a single alarm. Alarm correlation reduces the number of reported alarms thereby reducing the network load.

Following different types of alarm correlations are supported.

Generalization

Generalization will group two or more events into a single alarm. A Generalized Alarm will further use one of the correlation types - none, time-bound, counting & compression – for applying correlation logic to the new alarm.

E.g.

```
- ALARM_ID: 1536
 EVENT: IFMGR_IF_DOWN,OSPF_OPR_LINK_DOWN
 GENERALIZED_EVENT_NAME: LINK_DOWN
 ALARM_DESC: 'Link Down Detected'
 CORRELATION_TYPE: 1
 GENERALISED_CORRELATION_TYPE: 2
 CORRELATION_COUNTER: 2
 CORRELATION_TIMER_DURATION: 10000
 CORRELATION_SEVERITY: null
 POSITION:
   STR_POSITION_1_EVENT_1: 2
   STR_POSITION_1_EVENT_2: 6
 SNMP_TRAP: 0
 SNMP_OID: null
 NETCONF_NOTIFICATION: 0
 CLEAR_ALARM: LINK_UP
 SNMP_TRAP_CLEAR: 0
 SNMP_CLEAR_OID: null
 NETCONF_CLEAR_NOTIFICATION: 0
```

Above is a generalized LINK_DOWN alarm which will be raised when IFMGR_IF_DOWN and OSPF_OPR_LINK_DOWN events occur. The GENERALISED_CORRELATION_TYPE applied to this is Time-bound correlation.

Time Bound

The Time-bound logic stipulates that when the event is received, a timer is started for that event. While the timer is running, subsequent events of the same type will be suppressed. On expiry of the timer, an Alarm will be raised for that event stating the count for the number of times that event was received in this duration.

In case CLEAR_ALARM is specified for this event type following will be the behaviour –

- If clear event is received after expiry of the timer, the IFMGR_IF_DOWN will be raised and the alarm time will be marked as Active. Once the clear event is received (IFMGR_IF_UP), the IFMGR_IF_DOWN alarm will be marked as In-active.
- If clear event is received before expiry of the timer, the IFMGR_IF_DOWN will be raised and the alarm time will be marked as In-active. The CLEAR Alarm (IFMGR_IF_UP) will be raised as usual.

E.g.

```
- ALARM_ID: 1000
    EVENT: IFMGR_IF_DOWN
    GENERALIZED_EVENT_NAME: IFMGR_IF_DOWN
    ALARM_DESC: 'Interface changed state to down'
    CORRELATION_TYPE: 2
    GENERALISED_CORRELATION_TYPE: null
    CORRELATION_COUNTER: 2
    CORRELATION_TIMER_DURATION: 10000
    CORRELATION_SEVERITY: null
    POSITION:
      STR_POSITION_1_EVENT_1: 2
    SNMP_TRAP: 0
    SNMP_OID: null
    NETCONF_NOTIFICATION: 0
    CLEAR_ALARM: IFMGR_IF_UP
    SNMP_TRAP_CLEAR: 0
    SNMP_CLEAR_OID: null
    NETCONF_CLEAR_NOTIFICATION: 0
```

For the above configuration, the CORRELATION_TIMER_DURATION is stated as 10000ms. When the event (IFMGR_IF_DOWN) is received, a 10s timer will be started. While the timer is running, the subsequent such events will be suppressed. On expiry of the 10s duration, the IFMGR_IF_DOWN alarm will be raised and the count for number of times these events came in this duration will be specified.

Counting

The Counting logic considers a specified number of similar events as one. If the event type is given Correlation type as COUNTING, the respective alarm will be raised after the event has occurred for 'count' times.

E.g.

```
- ALARM_ID: 1000
    EVENT: IFMGR_IF_DOWN
    GENERALIZED_EVENT_NAME: IFMGR_IF_DOWN
    ALARM_DESC: 'Interface changed state to down'
    CORRELATION_TYPE: 3
    GENERALISED_CORRELATION_TYPE: null
    CORRELATION_COUNTER: 2
    CORRELATION_TIMER_DURATION: 10000
    CORRELATION_SEVERITY: null
    POSITION:
      STR_POSITION_1_EVENT_1: 2
    SNMP_TRAP: 0
    SNMP_OID: null
    NETCONF_NOTIFICATION: 0
    CLEAR_ALARM: IFMGR_IF_UP
    SNMP_TRAP_CLEAR: 0
    SNMP_CLEAR_OID: null
    NETCONF_CLEAR_NOTIFICATION: 0
```

For the above configuration, the CORRELATION_TYPE is specified as 3 (Counting). The CORRELATION_COUNTER is 2. When the IFMGR_IF_DOWN event is received two times, the IFMGR_IF_DOWN alarm will then be raised by the FMS.

Compression

The Compression takes multiple occurrences of the same event, examines them for duplicate/ redundant events information, removes redundancies and reports them as a single event.

E.g.

```
- ALARM_ID: 1000
  EVENT: IFMGR_IF_DOWN
  GENERALIZED_EVENT_NAME: IFMGR_IF_DOWN
  ALARM_DESC: 'Interface changed state to down'
  CORRELATION_TYPE: 4
  GENERALISED_CORRELATION_TYPE: null
  CORRELATION_COUNTER: 2
  CORRELATION_TIMER_DURATION: 10000
  CORRELATION_SEVERITY: null
  POSITION:
    STR_POSITION_1_EVENT_1: 2
  SNMP_TRAP: 0
  SNMP_OID: null
  NETCONF_NOTIFICATION: 0
  CLEAR_ALARM: IFMGR_IF_UP
  SNMP_TRAP_CLEAR: 0
  SNMP_CLEAR_OID: null
  NETCONF_CLEAR_NOTIFICATION: 0
```

For the above configuration, on receiving the IFMGR_IF_DOWN event, the IFMGR_IF_DOWN alarm will be raised. Until IFMGR_IF_UP (configured as CLEAR_ALARM above) is received, subsequent IFMGR_IF_DOWN events will be suppressed.

No-Correlation

The No-correlation logic is for events which will not undergo correlation algorithm. For these events, matching alarms are specified in the above stated YAML file, will be raised.

By default, all events stated in the configuration YAML file will be of type NO-CORRELATION, implying that when FMS is enabled, the alarms for these events will be raised.

Drop Event

The Drop event logic is for events that are not considered for converting into alarms. The specific events are dropped or not considered for correlation.

If the correlation type is Generalization, since, the new alarm has to be processed, the event will be given to the Generalization handler. In case the GENERALISED_CORRELATION_TYPE specified for the alarm is Drop-event(5), then the event will not be considered for Generalized alarm generation also.

Using FMS

There are two different ways this Fault Management System (FMS) can be used –

1. A network operator can use CLIs to monitor and manage faults reported by devices by logging into each OcNOS powered device individually.
2. At the network level, FMS can be integrated into customer's own management system using an adapter, pushing the faults/alarms from individual network elements powered by OcNOS into a centralized fault management system

All the alarms generated by OcNOS powered devices are stored in Influx database. Network administrators can login to the device and use the CLI commands to get the current or historical status of faults in the device.

Here are some example CLI commands the Network Administrator can use to manage faults at a network element level:

```
OcNOS(config)#fault-management enable
OcNOS#sh alarm active
Active Alarms received:-
Active Alarm Count: 1
Severity     Status     Alarm Description
NOTIF        Active     OcNOS [LINK_DOWN] Link Down Detected: ce49

OcNOS#sh alarm statistics
Alarm Statistics received:-
Alarm Count: 1
Severity    Count    Alarm Description
NOTIF       1        OcNOS [LINK_DOWN] Link Down Detected: ce49
```

Above sample CLI commands shows current faults in this device which tell the user that the interface ce49 is down. User takes necessary action to bring the interface up (either by resetting the interface or by replacing faulty transceiver/cables). When the interfaces is operationally up, FMS clears this alarm.

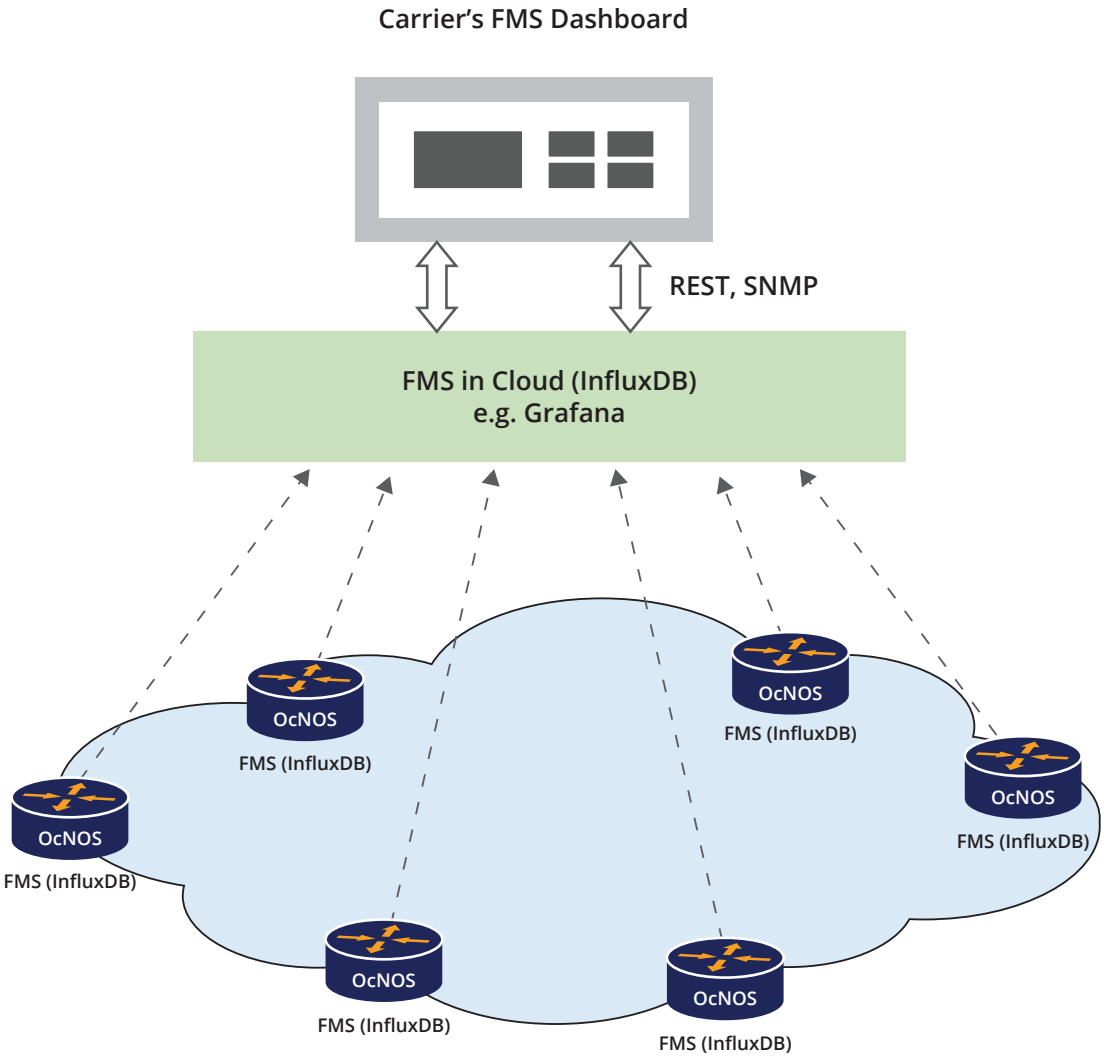
```
OcNOS#sh alarm active
Active Alarms received:-
Active Alarm Count: 0
Severity     Status     Alarm Description

OcNOS#sh alarm statistics
Alarm Statistics received:-
Alarm Count: 2
Severity    Count    Alarm Description
NOTIF       1      OcNOS [LINK_DOWN] Link Down Detected: ce49
NOTIF       1      OcNOS [LINK_UP] Link Up Detected: ce49
```

Managing Faults at Network Level

Fault Management at a network element level is useful when fault is localized to a specific device and troubleshooting needs to be performed manually at that specific device. Managing a large scale network requires a centralized Fault Management System to which individual network elements publishes its faults as and when it occurs. A centralized fault dashboard provides network operator the overall health of their network.

In order to Integrate OcNOS FMS with a third party centralized Fault Management System, an Adapter is required. This Adapter runs in the cloud and gathers faults from all OcNOS powered devices.



Conclusion

Fault Management System provides information on the health of the network by locating, detecting and correcting network problems and thereby increasing network reliability. OcNOS FMS provides a framework for managing faults at network element level which can be extended with the help of an adapter to manage the faults at the network level allowing the seamless integration of OcNOS into Carrier's existing management system.

ABOUT IP INFUSION

IP Infusion, a leader in disaggregated networking solutions, delivers enterprise and carrier-grade software solutions allowing network operators to reduce network costs, increase flexibility, and to deploy new features and services quickly. IP Infusion is headquartered in Santa Clara, Calif., and is a wholly owned and independently operated subsidiary of ACCESS CO., LTD. Additional information can be found at **http://www.ipinfusion.com**

Phone | +1 877-MYZEBOS
Email | sales@ipinfusion.com
Web | www.ipinfusion.com

U.S. (Santa Clara) | +1 408-400-1912
Japan (Tokyo) | +81 03-5259-3771
Korea (Seoul) | +82 10 2733 3016

India (Bangalore) | +91 (80) 6728 7000
China (Shanghai) | +86-186 1658 6466
EMEA | +49 (208) 8290 6464

IP Infusion | An ACCESS Company | (408) 400-3000 | www.ipinfusion.com | 3965 Freedom Circle, Suite 200 Santa Clara, CA 95054