January 2020

AUTOMATED AND REACTIVE MECHANISM IN CONNECTIVITY FAULT MANAGEMENT (CFM) WITH AN ON-DEVICE EXPERT SYSTEM

Karthik babu Harichandra Babu

Peter Jones

Peter Van Horne

Ananthakrishnan Rajamani

AUTOMATED AND REACTIVE MECHANISM IN CONNECTIVITY FAULT MANAGEMENT (CFM) WITH AN ON-DEVICE EXPERT SYSTEM

AUTHORS:

Karthik babu Harichandra Babu
Peter Jones
Peter Van Horne
Ananthakrishnan Rajamani

ABSTRACT

Presented herein is an "On-Device Expert System" for implementation at devices operating in accordance with Connectivity Fault Management (CFM). In particular, in addition to fault detection/verification/notification, the techniques presented herein now enable fault device/link identification. The "On-Device Expert System" performs telemetry operations inside the device itself, instead of having the same operations performed in a controller. As such, the techniques presented herein avoid the need for multiple resources usage in the controller for the same telemetry details. With the "On-Device Expert System" providing the needed telemetry details to the controller, the controller now can directly trigger the application associated with a received telemetry message and perform the remedial/enhancement/notification needed for the networks.

DETAILED DESCRIPTION

Institute of Electrical and Electronics Engineers (IEEE) Connectivity Fault Management (CFM) is an end-to-end per-service Ethernet layer Operations, Administration, and Maintenance (OAM) protocol. CFM includes proactive connectivity monitoring, fault verification, and fault isolation for large Ethernet metropolitan-area networks (MANs) and WANs. Applications have been developed to manage devices using CFM, where these applications may show the devices/hosts/endpoints and their connections in real time. Conventionally, CFM is configured in the devices in order to find and administer the faulty connections between the devices.

However, in conventional arrangements, the fault verification and fault isolation used to troubleshoot an affected network part is performed manually (i.e., it takes human intervention and, typically, a considerable amount of time). Fault verification and fault isolation are administrative actions, typically performed after fault detection, where fault verification can be used to confirm successful initiation or restoration of connectivity. In these conventional arrangements, the network administrator uses a Loopback protocol to perform fault verification. A maintenance endpoint (MEP) (i.e., an active CFM entity generating and responding to CFM protocol messages) can be ordered to transmit a Loopback Message (LBM) to another MEP or to a maintenance intermediate point (MIP) in the Maintenance Association (MA), whose MAC address can be discovered from Continuity Check Messages (CCMs) (MEPs only) or CFM linktrace messages (LTMs)/ CFM linktrace replies (LTRs) (MEPs or MIPs). The receiving MEP/MIP responds by transforming the LBM into a unicast Loopback Reply sent back to the originating MEP. That MEP records the responses for examination by the administrator. All these actions are performed by the administrators manually or through NMS scripts once the fault is detected.

Conventional controllers reply upon telemetry and these controllers involves multiple complex structures. As such, conventional controllers add a significant amount of overhead to the resources used to obtain necessary telemetry details from a managed device that enable the controller to perform the needed actions. Pushing these operations the managed devices themselves his would reduce the overall overhead involved in the controller.

In general, fault detection in an MA is done by the Continuity Check Messages (CCMs) that are transmitted between the MEPs. The inability of an MEP to receive three consecutive CCMs from any one of the other MEPs in its MA indicates either a MEP failure or a network failure.

A CCM is a message that is: (1) destination to a Media Access Control (MAC) address that is tied to the ME Level of the packet in order to make it easier to confine CCs to a particular domain (Provider, Operator, etc.), (2) indicates ME Level, Service Identifier, MEP Identifier, Transaction Identifier, and (3) has a Lifetime. Custom CCMs can be used to detect failures and network delays (using timestamps). As described below, in

accordance with the techniques presented herein, when a failure /fault condition is detected, a fault isolation action may be initiated. The techniques presented herein initiate automatic fault isolation using Loop-back messages and Link-Trace messages. Earlier existing solution is initiating the same via manual intervention.

Loop-back messages and Link-Trace messages can help the fault isolation algorithm to identify the faulty node in the CFM Domain. When a fault is isolated it is logged and administrator can use the information and automatic fault isolation reduces administrator intervention and improves user experience. This allows the network administrator to avoid SLA slippages and improved customer satisfaction. FIGs. 1 and 2, below, illustrate aspects of the techniques presented herein.

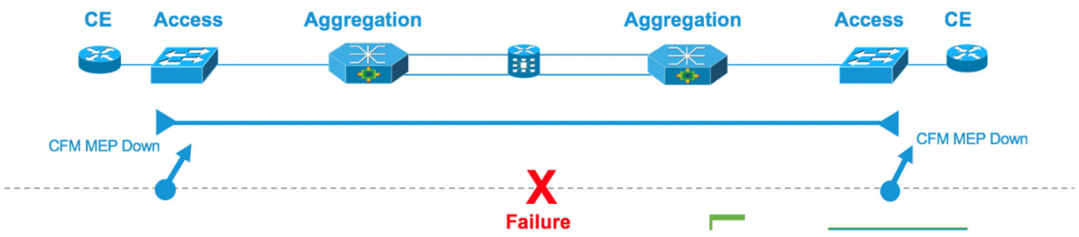

FIG. 1

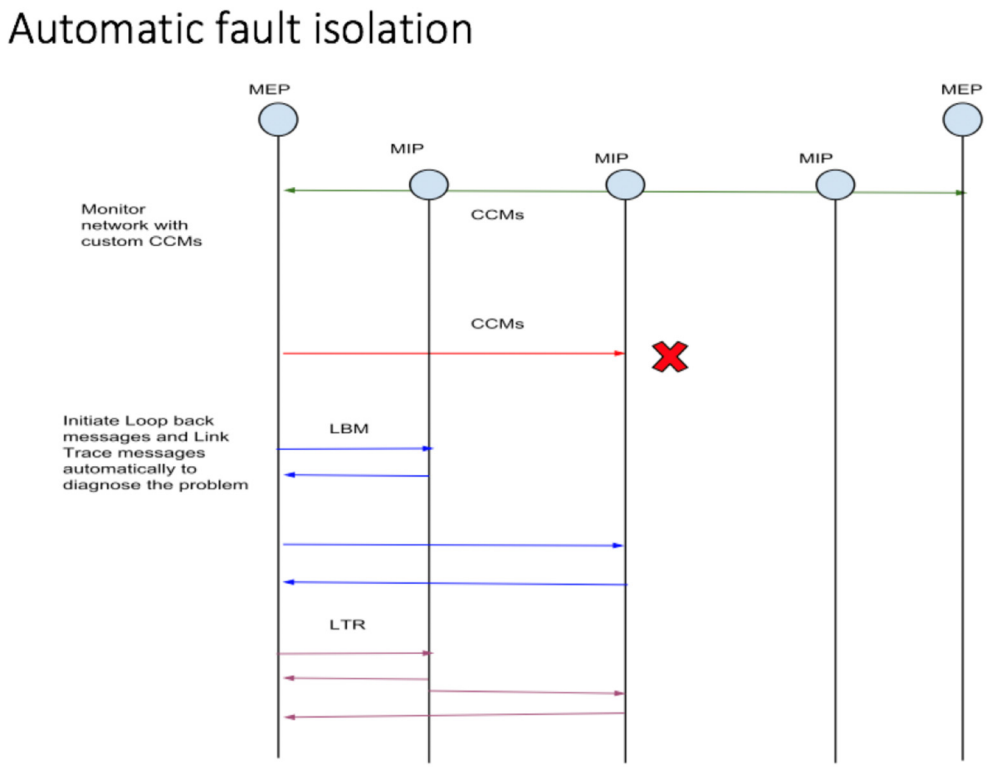

FIG. 2

In accordance with the techniques presented herein, once the fault is identified using the syslog messages (collected using telemetry invokes), the controller will proactively and automatically trigger the fault verification (loopback) and fault isolation (link trace), without any manual intervention. The CCM message already has the necessary details which will be fetched by the controller in order to trigger the link trace and loopback to the remote MEP. In addition, the existing CCM packet will be enhanced with the additional information (e.g., Host Name/Port state/L3 Management address/Node down or Link Down) to provide the exact details of the link/node which is fault. Once this information and the faulty node/link details are fetched by the controller, fault notification is triggered.

Fault notification is provided by a MEP that has detected a connectivity fault in its MA using syslog which are then collected by the pipeline which will trigger the above operations automatically and provide the administrator with the exact fault details.

A single fault may be detected by multiple MEPs and may result in the generation of multiple above triggers, a flag can be used by the APP to suppress for the particular MEP/NODE Name, such as:

*Feb 26 19:38:51.207 IST: %E_CFM-3-REMOTE_MEP_DOWN: Remote MEP mpid 10 evc evc1501 vlan 1501 MA name s1 in domain close changed state to down with event code TimeOut.

*Feb 26 19:38:51.257 IST: %E_CFM-6-REMOTE_MEP_UP: Continuity Check message is received from a remote MEP with mpid 10 evc evc1501 vlan 1501 MA name s1 domain close interface status Up event code Returning.

In accordance with the techniques presented herein, an "On-Device Expert System" is used to process and take action on events indicating a change in the state of services configured on the device. The "On-Device Expert System" can perform part or all of troubleshooting and repair workflows that currently require manual intervention by network operators. FIG. 3, below, is a schematic diagram illustrated operations in accordance with the techniques presented herein.

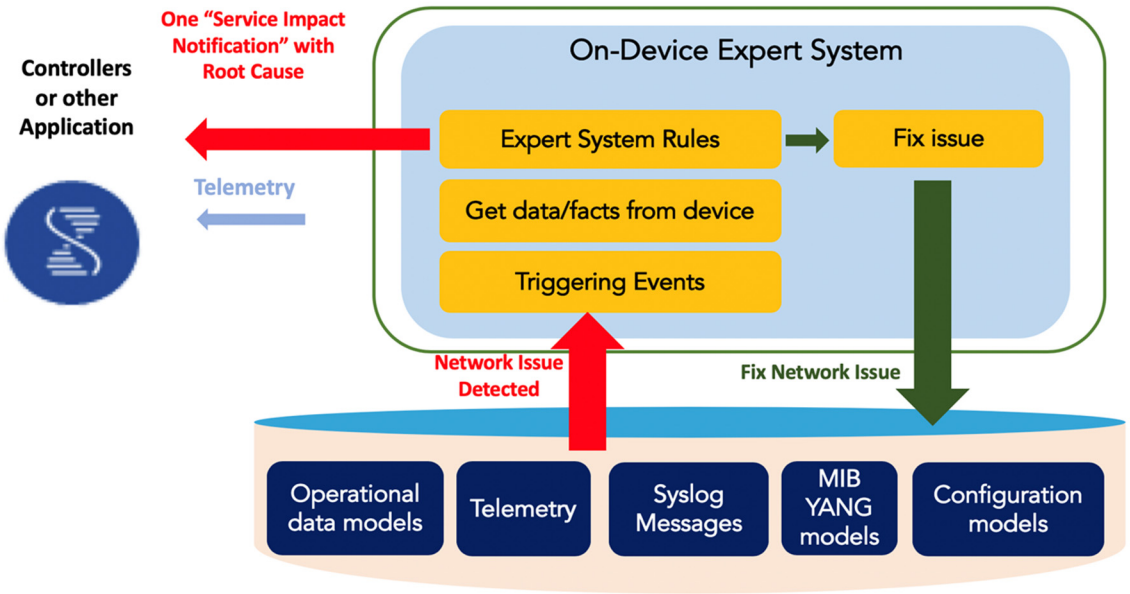

FIG. 3

In accordance with the techniques presented herein, the occurrence of the triggering events indicating that a connection failure has been detected or that a connection has been restored triggers the execution of an expert system. The expert system running on the device automatically collects data from multiple sources, including device operational state, configuration and system events. This information is evaluated by the expert system using rules that analyze the device information to determine the cause of the connection failure or the reason for session re-establishment. When the reason or cause for the connection state change is determined, the expert system can take multiple actions including:

- Send a detailed "Service Impact Notification" to the controller, which includes the identification of the connection, services impacted by the connection and causes.
- Apply configurations, if required, to fix the cause of the connectivity failure or take other action including re-directing traffic.
- Automatically enable telemetry or other monitoring capabilities to enable deeper diagnostics of the issue.
- Implement a series of automated workflow steps to perform deeper diagnostics using local processing capabilities on the device to reduce the complexity of processing required by the controller.

In general, the above is the interested telemetry data for which the application is built. Once this data is collected by the pipelines the necessary actions may be performed, including:

- Automatic fault verification/fault isolation is performed by the controller.
- Remedial action for the faulty link is done automatically as an action for fault using the controller.
- Fault identification provides the necessary details about the fault link/devices which is collected as the result of fault isolation/verification and provided to network administrator to troubleshoot and correct the faulty link/devices.

FIG. 4, below, illustrates a flow of a controller, in accordance with the techniques presented herein.

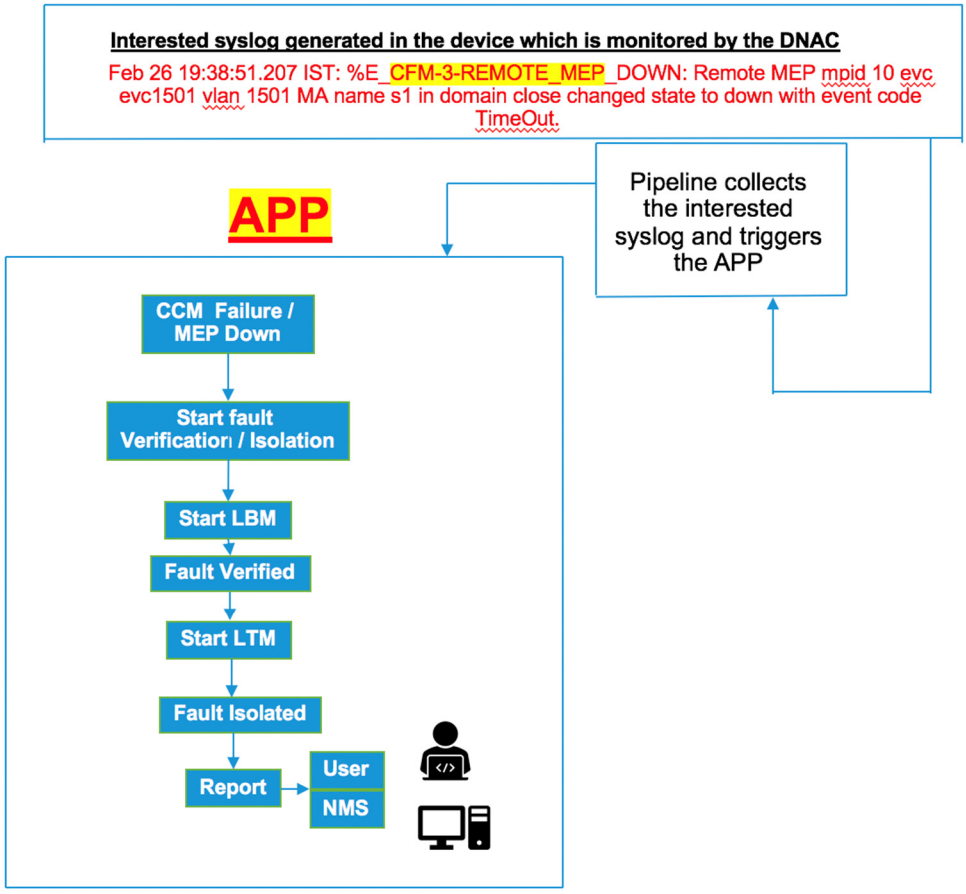

FIG. 4

As noted, conventional controllers may perform fault detection/verification/notification and, the techniques presented herein, enable a controller to also fault device/link identification. In addition, adding new fields to obtain the remote peer host name/port state/L3 management address in the protocol will be easier for the administrator to find the issue. Moreover, based on the fault identification, the required actions, such as rerouting of the traffic, applying QOS, activating the redundant link/node, etc. can be performed in the wire speed using the controller, thereby reducing the loss and time taken by the manual or script efforts to find the faulty node/link.

In addition, in accordance with the techniques presented herein, monitoring of the CFM MEPs does not require any manual/script intervention and can be taken to the next level automation to perform the EEM actions based on the situation. The proposed "On-

Device Expert System" performs the telemetry operations inside of the device itself, instead of having these same operations performed in the controller. This, accordingly, avoids the need for multiple resources usage in the controller for the same telemetry details. With the "On-Device Expert System" providing the needed telemetry details to the controller, the controller now can directly trigger the application associated with the received telemetry message to perform the remedial/enhancement/notification needed for the networks.

In general, the techniques presented herein may run on top of an existing controller or the techniques presented herein can operate as an application which will be triggered when the service impact notification is received by an existing controller or application. The action of the application can be the remedial network act. The telemetry operations described herein may be implemented through an on device machine reasoning engine to perform the root cause analysis and provide the needed telemetry to the controller or controller. This reduces the overhead currently incurred at the controller in conventional arrangements in order to fetch the telemetry details.

In certain examples, an Embedded machine Reasoning Engine (EMRE) Model-based Python-Enabled Automated Troubleshooting is proposed to perform the automated protocol consistency checking/detect and repair misconfigurations automatically. The techniques presented herein provide the ability to collect the required debug/show outputs/logs as of when the service impact notification is received. The techniques presented herein, apart from being run on top of a controller, can also run on the device inside a container waiting for the service impact to get triggered and perform the necessary remedial action. With respect to CFM technology, apart from fault detection/verification/notification which is provided now using CCM/LTM/LBM (where the LTM/LBM are manually triggered once the fault is notified), the EMRE adds the automatic fault device/link identification.

The techniques presented herein are not necessarily specific to CFM triggers alone, but instead can generic and used with any component triggers with the capability to build on many serviceability use cases.

In summary, the techniques presented herein propose the deployment of a configurable rules-based expert system on managed devices to broadly monitor device state and to make decisions on actions to take. The expert system is triggered when events that may indicate a device problem are detected, while ignoring messages that do not indicate a device problem. These decisions are made by evaluating "RULES" using "FACTS," which are multiple conditions including feature state, device state, counters, device configurations and other parameters. The expert system uses RULES defined by the designer to make decisions, which is different from the traditional approach of writing procedural code. The techniques presented herein also use the expert system to automate the implementation of workflows to, accordingly, automate troubleshooting and repair of issues.