# PDR Deterministic Plugin

# Table of contents

# List of Figures

## Overview

The PDR deterministic plugin, overseen by the UFM, is a docker container that isolates malfunctioning ports, and then reinstates the repaired links to their previous condition by lifting the isolation. The PDR plugin uses a specific algorithm to isolate ports, which is based on telemetry data from the UFM Telemetry. This data includes packet drop rate, BER counter values, link down counter, and port temperature. Any decisions made by the plugin will trigger an event in the UFM for tracking purposes.

The PDR plugin performs the following tasks:

1. Collects telemetry data using UFM Dynamic Telemetry

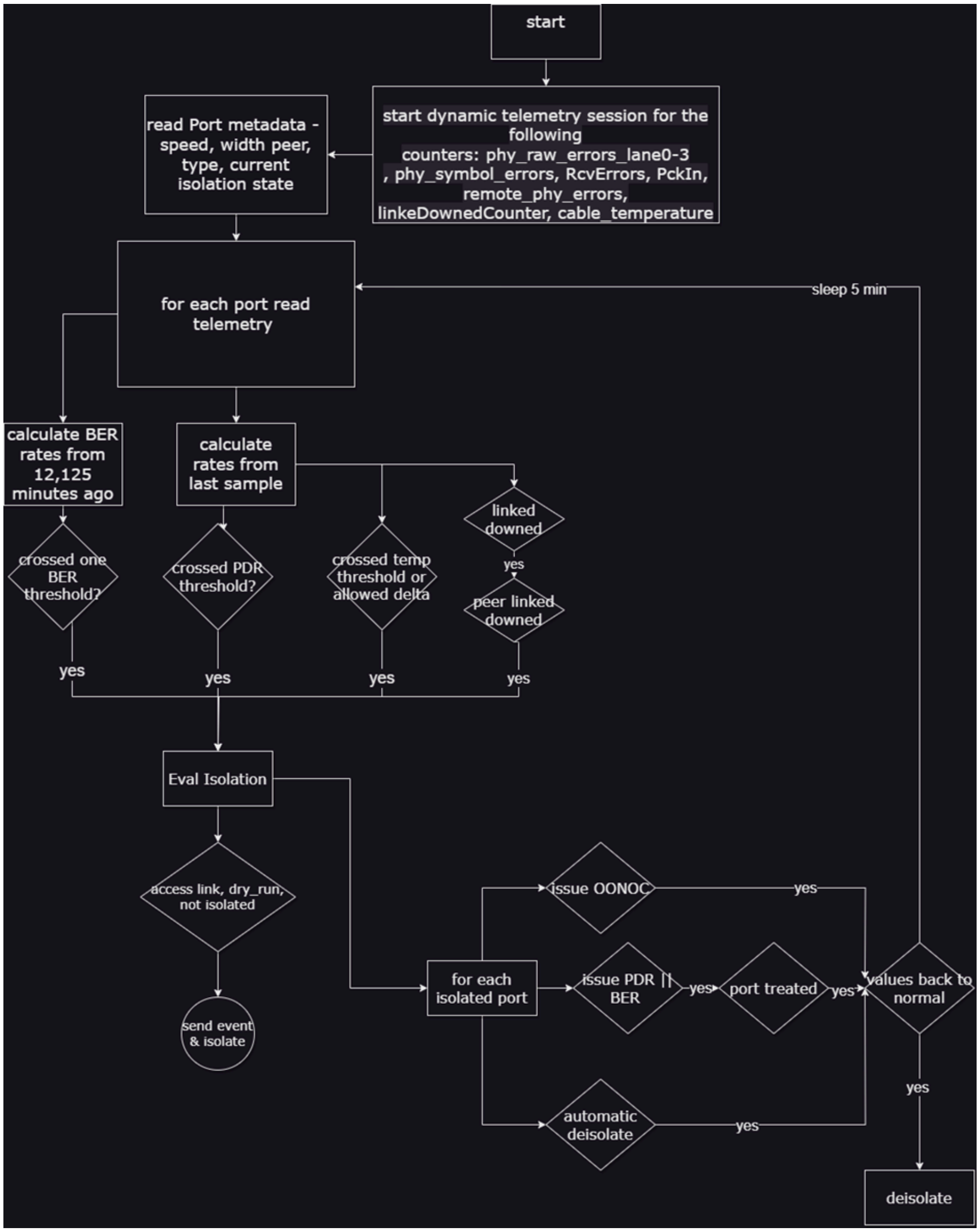
2. Identifies potential failures based on telemetry calculations and isolates them to avert any interruption to traffic flow

3. Maintains a record of maintenance procedures that can be executed to restore an isolated link

4. After performing the required maintenance, the system verifies if the ports can be de-isolated and restored to operational status (brought back online).

The plugin can simulate port isolation without actually executing it for the purpose of analyzing the algorithm's performance and decision-making process in order to make future adjustments. This behavior is achieved through the implementation of a "dry_run" flag that changes the plugin's behavior to solely record its port "isolation" decisions in the log, rather than invoking the port isolation API. All decisions will be recorded in the plugin's log.

## Schematic Flow


start
start dynamic telemetry session for the following counters: phy_raw_errors_lane0-3 , phy_symbol_errors, RcvErrors, PckIn, remote_phy_errors, linkeDownedCounter, cable_temperature
read Port metadata - speed, width peer, type, current isolation state
for each port read telemetry
sleep 5 min
calculate BER rates from 12,125 minutes ago
calculate rates from last sample
linked downed
yes
crossed one BER threshold?
crossed PDR threshold?
crossed temp threshold or allowed delta
peer linked downed
yes
yes
yes
yes
Eval Isolation
access link, dry_run, not isolated
send event & isolate
issue OONOC
yes
for each isolated port
issue PDR || BER
yes
port treated
yes
values back to normal
automatic deisolate
yes
yes
deisolate

# Deployment

To deploy the plugin, follow these steps:

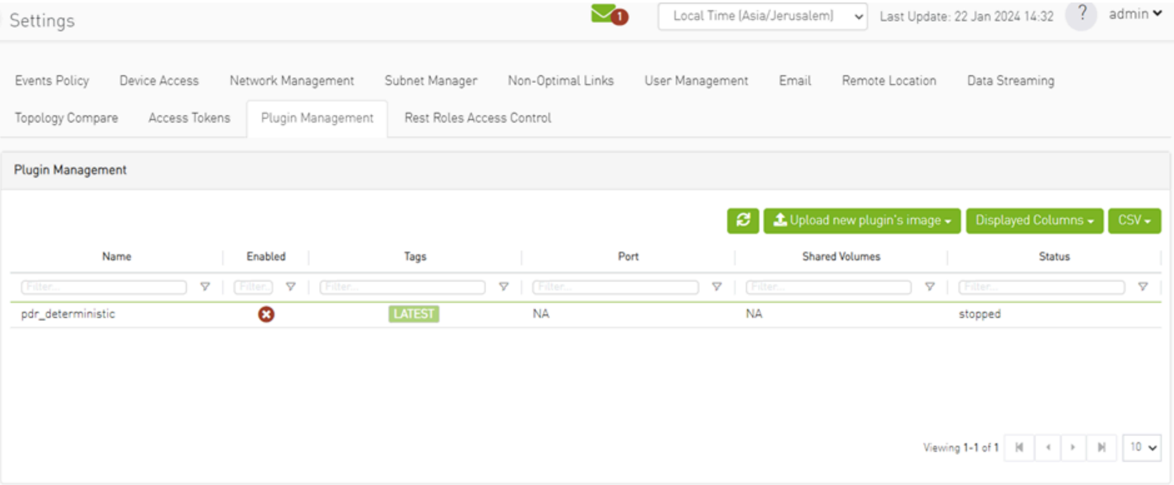
1. Download the ufm-plugin-pdr_deterministic-image from the **Docker Hub**.
2. Load the downloaded image onto the UFM server. This can be done either by using the UFM GUI by navigating to the Settings -> Plugins Management tab or by loading the image via the following instructions:

1. Log in to the UFM server terminal.

2. Run:

```
docker load -I <path_to_image>
```

3. After successfully loading the plugin image, the plugin should become visible in the plugin management table within the UFM GUI. To initiate the plugin's execution, simply right-click on the respective in the table.



Plugin Management

| Name | Enabled | Tags | Port | Shared Volumes | Status |
|---|---|---|---|---|---|
| pdr_deterministic | ☒ | LATEST | NA | NA | stopped |

# Isolation Decisions

# NDR Link Validation Procedure

Verify ports that are in INIT, ARMED or ACTIVE states only. Track the SymbolErrorsExt of every such link for at least 120m. If polling period is Pm, need to keep N=(125+Pm+1)/Pm samples. Also, two delta samples are computed: number of samples covering 12 minutes S12m = (12 + Pm + 1)/Pm and S125m = (125 + Pm + 1)/Pm. 12m_thd = LinkBW_Gbps*1e9*12*60*1e-14 (2.88 for NDR) and
125m_thd = LinkBW_Gbps*1e9*125*60*1e-15 (3 for NDR).

Check the following conditions for every port in the given set:

1. If the Delta(LinkDownedCounterExt) port is > 0 and the Delta(LinkDownedCounterExt) remote port is > 0, add it to the list of bad_ports. This condition should be ignored if the --no_down_count flag is provided.

2. If the symbol_errors[now_idx] – symbol_errors[now_idx – S12m] is > 12m_thd, add the link to the list of bad_ports, and continue with next link.

3. If the symbol_errors[now_idx] – symbol_errors[now_idx – S125m] is > 125m_thd, add the link to the list of bad_ports, continue with next linkPacket drop rate criteria

When packet drops due to the link health are detected, isolate the problematic link. To achieve this, a target packet_drop/packet_delivered ratio can be employed to include TX ports with a receiver exceeding this threshold in the list of bad_ports. However, the drawback of this method is that such links may fluctuate between bad/good state since their BER may be normal. Therefore, it is advisable to track their statistics over time and refrain from reintegrating them after their second or third de-isolation.

## Return to Service

Continuously monitoring the collection of bad_ports, the plugin persistently assess their Bit Error Rate (BER) and determines their reintegration when they successfully pass the 126m test without errors.

## Configuration

The following parameters are configurable via the plugin's configuration file. (pdr_deterministic.conf)

| Name | Description | Default Value |
|---|---|---|
| INTERVAL | Interval for requesting telemetry counters, in seconds. | 300 |
| MAX_NUM_ISOLATE | Maximum ports to be isolated. max(MAX_NUM_ISOLATE, 0.5% * fabric_size) | 10 |
| TMAX | Maximum temperature threshold | 70 (Celsius) |
| D_TMAX | Maximum allowed Temperature Delta | 10 |
| MAX_PDR | Maximum allowed packet drop rate | 1e-12 |

| Name | Description | Default Value |
|---|---|---|
| CONFIGURED_BER_CHECK | If set to true, the plugin will isolate based on BER calculations | True |
| CONFIGURED_TEMP_CHECK | If set to true, the plugin will isolate based on temperature measurements | True |
| LINK_DOWN_ISOLATION | If set to true, the plugin will isolate based on LinkDownedCounterExt measurements | False |
| SWITCH_TO_HOST_ISOLATION | If set to true, the plugin will isolate ports connected via access link | False |
| DRY_RUN | Isolation decisions will be only logged and will not take effect | False |
| DEISOLATE_CONSIDER_TIME | Consideration time for port de-isolation (in minutes) | 5 |
| DO_DEISOLATION | If set to false, the plugin will not perform de-isolation | True |
| DYNAMIC_WAIT_TIME | Seconds to wait for the dynamic telemetry session to respond | 30 |

## Calculating BER Counters

For calculating BER counters, the plugin extracts the maximum window it needs to wait for calculating the BER value, using the following formula:

$$seconds = \frac{max\_BER\_target^{-1}}{min\_port\_rate}$$

**Example:**

| Rate | | BER Target | Minimum Bits | Minimum Time in Seconds | In Minutes |
|---|---|---|---|---|---|
| HDR | 2.00E+11 | 1.00E-12 | 1.00E+12 | 5 | 0.083333 |
| HDR | 2.00E+11 | 1.00E-13 | 1.00E+13 | 50 | 0.833333 |
| HDR | 2.00E+11 | 1.00E-14 | 1.00E+14 | 500 | 8.333333 |

| Rate | | BER Target | Minimum Bits | Minimum Time in Seconds | In Minutes |
|---|---|---|---|---|---|
| HDR | 2.00E+11 | 1.00E-16 | 1.00E+16 | 50000 | 833.3333 |

BER counters are calculated with the following formula:

$$BER = \frac{error\ bits_i - error\ bits_{i-1}}{total\ bits_i - total\ bits_{i-1}} = \frac{error\ bits_i - error\ bits_{i-1}}{Link\ data\ rate * (time_i - time_{i-1})}$$

## Ports Exclusion List

You can designate specific ports to be excluded from PDR analysis, isolation, or de-isolation for an indefinite or limited period. Already excluded ports can also be removed from this list.

Ports are added to or removed from the exclusion list via the PDR plugin's REST API.

To add ports to the exclusion list (to be excluded from analysis), run:

```
curl -k -i -u <user:password> -X PUT
'https://<host_ip>/ufmRest/plugin/pdr_deterministic/excluded' -d ' [<formatted_ports_list>]' -H
"Content-Type: application/json"
```

Optionally, you can specify a TTL (time to live in the exclusion list) following the port after the comma. If zero or not specified, the port is excluded. For example:

```
-d '[["9c0591030085ac80_45"],["9c0591030085ac80_46",300]]'
```

To remove ports from the exclusion list:

```
curl -k -i -u <user:password> -X DELETE
'https://<host_ip>/ufmRest/plugin/pdr_deterministic/excluded' -d '[<comma_separated_port_names>]'
-H "Content-Type: application/json"
```

Example:

```
-d '["9c0591030085ac80_45","9c0591030085ac80_46"]'
```

To retrieve ports and their remaining exclusion times from the exclusion list:

```
curl -k -i -u <user:password> -X GET
'https://<host_ip>/ufmRest/plugin/pdr_deterministic/excluded'
```