

Troubleshoot Inter Rack Replication Failure with Error Code "424-Geo-replication Checksum Mismatch"

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[What is Geo-redundancy in SMF?](#)

[Geo-replication Pod](#)

[Identify the Active Geo Pod and Standby Geo Pod](#)

[Functionalities of GR POD](#)

[GR Pods Replicate the ETCD and Cache Pod Data across the Site](#)

[Maintain Site Local Instances Roles in ETCD](#)

[Monitor Local Site Status \(POD Status/BFD Status\)](#)

[Site Roles](#)

[GR-Triggers](#)

[CLI to Verify the GR Instance Roles on Rack](#)

[CLI to Reset Role from Standby Error to Standby](#)

[CLI to Switch Role from Standby to Standby Error](#)

[CLI to Switch Role from Standby to Primary](#)

[TCP Connection Termination](#)

[Problem](#)

[Scenario 1. Geo-replication Checksum for Instance Id 1 has IPAM Cache and NRFMgmt Cache Checksum Mismatch](#)

[Scenario 2. Geo-replication Checksum for Instance Id 2 has ETCD Checksum Mismatch](#)

[Scenario 3. TCP Connection Establishment Failure with Remote Site](#)

[Scenario 4. DIMM Error Observed on Server which Hosts Master Node](#)

[Solution](#)

Introduction

This document describes various investigation methods to troubleshoot Geo-replication Checksum Mismatch between the Local and Remote racks.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Geo-redundancy in Session Management Function (SMF)
- SMF
- Transmission Control Protocol (TCP) Connection Termination

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

What is Geo-redundancy in SMF?

- SMF supports Geographical (Geo)- redundancy (GR) in active-active mode.
- GR setup is also responsible for the replication of `etcd/cache` data to the standby Rack.
- SMF supports primary/standby redundancy in which data is replicated from the Primary to the standby instance.
- If the primary instance fails, the standby instance becomes the primary and takes over the operation.
- To achieve GR, two primary/standby pairs can be set up where each site actively processes traffic and standby acts as a backup for the remote site.

Geo-replication Pod

- Geo-replication Pod is introduced for Inter-rack/Site communication and to monitor POD/BFD within the rack
- Two instances of GR-POD run on each rack/site
- Two GR PODs function in Active-Standby mode
- GR PODs are spawned on the Proto node/VM
- GR POD uses two Virtual IP Addresses (VIPs)
- Internal-VIP for Inter-POD communication (within the rack)
- External-VIP for Inter-Rack/Site's GR POD communication
- VIPs configured for GR POD can be active on one of the Proto node/VM
- When Active GR POD restart, VIP is switched to another Proto node/VM and Standby GR POD run on the other Proto node/VM can become Active

GR Pod Reference Configuration:

```
smf# show running-config instance instance-id 1 endpoint geo
Thu Oct 20 06:25:25.319 UTC+00:00
instance instance-id 1
endpoint geo
replicas 1
nodes 2
interface geo-internal
vip-ip a.b.c.d vip-port 7001
```

```
exit
interface geo-external
vip-ip Y.Y.Y.Y vip-port 7002
exit
exit
exit
```

Identify the Active Geo Pod and Standby Geo Pod

In order to identify the active Geo pod, you need to check for errors or events in the Geo pod logs.

Active pod:

```
user@smf-ims-master-1:~$ kubectl logs georeplication-pod-0 -n smf-smfix1|tail -3
[ERROR] [grcacachepod.go:339] [gr_deferred_sync.application.app] Periodic Sync: Total time taken
to sync IPAM cache pod data: 500.563723ms"
[ERROR] [GeoAdminStreamClient.go:276] [gr_pod.geo_admin_client.app] no one waiting for received
response for txnID:CP0XXXOKCP0XXX-SMF-IMS-smfix1111163550 of host=geo-admin-pod2
```

Standby Pod:

```
user@cp0xxx-smf-ims-master-1:~$ kubectl logs georeplication-pod-1 -n smf-smfix1|tail -3
[ERROR] [gr_pod.geo_replication_client_stream] Counters => not an active geo pod
[ERROR] [gr_pod.geo_replication_client_stream] Counters => not an active geo pod
[ERROR] [gr_pod.geo_replication_client_stream] Counters => not an active geo pod
```

Functionalities of GR POD

GR Pods Replicate the ETCD and Cache Pod Data across the Site

To view replication details for ETCD and cache-pod data, use CLI:

```
[cp0xxx-smf-ims/smf-smfix1] smf# show georeplication checksum instance-id 1
Thu Oct 20 07:11:52.409 UTC+00:00
checksum-details
--  ----  -----
ID      Type      Checksum
--  ----  -----
1      ETCD      1666249907
IPAM CACHE      1666249907
NRFMGmt CACHE  1666249907
```

Maintain Site Local Instances Roles in ETCD

```
[ERROR] [gr_pod.gradmin] updateEntryInEtcd: Updating etcd entries for keys : Instance.2, with
role as PRIMARY
[ERROR] [gr_pod.gradmin] updateEntryInEtcd: Updating etcd entries for keys : Instance.1, with
role as STANDBY
```

Monitor Local Site Status (POD Status/BFD Status)

```
[cp0xxx-smf-ims/smf-smfix1] smf# show running-config geomonitor podmonitor pods smf-service
Thu Oct 20 07:36:41.280 UTC+00:00
geomonitor podmonitor pods smf-service
retryCount      2
```

```
retryInterval          900
retryFailOverInterval 500
failedReplicaPercent  60
```

Site Roles

PRIMARY : The site is ready and actively takes traffic for the given instance.

STANDBY: The site is on standby, ready to take traffic but does not take traffic for a given instance.

STANDBY_ERROR: The site is in problem, not active and not ready to take traffic for a given instance.

FAILOVER_INIT: The site has started to failover and is not in the condition to take traffic, buffer time of 2s for the application to complete its activity.

FAILOVER_COMPLETE: The site has completed the failover and attempted to inform the peer site about the failover for the given instance. buffer time of 2s.

FAILBACK_STARTED: Manual failover is triggered with a delay from the remote site for a given instance.

Note: Cache/ETCD Replication and CDL Replication would happen even in all roles. If GR links are down/periodic heartbeat fails, GR triggers are suspended.

GR-Triggers

CLI to Verify the GR Instance Roles on Rack

```
Show role instance id 1
Show role instance id 2
```

CLI to Reset Role from Standby Error to Standby

```
Geo reset-role instance-id <1/2> role standby
```

CLI to Switch Role from Standby to Standby Error

```
Geo switch-role instance-id <1/2> role standby failback-interval 0
```

CLI to Switch Role from Standby to Primary

In order to initiate this Switch role, you need to trigger the CLI from the Rack which has one of the Instances as Primary.

```
Geo switch-role instance-id <1/2> role standby failback-interval 0
```

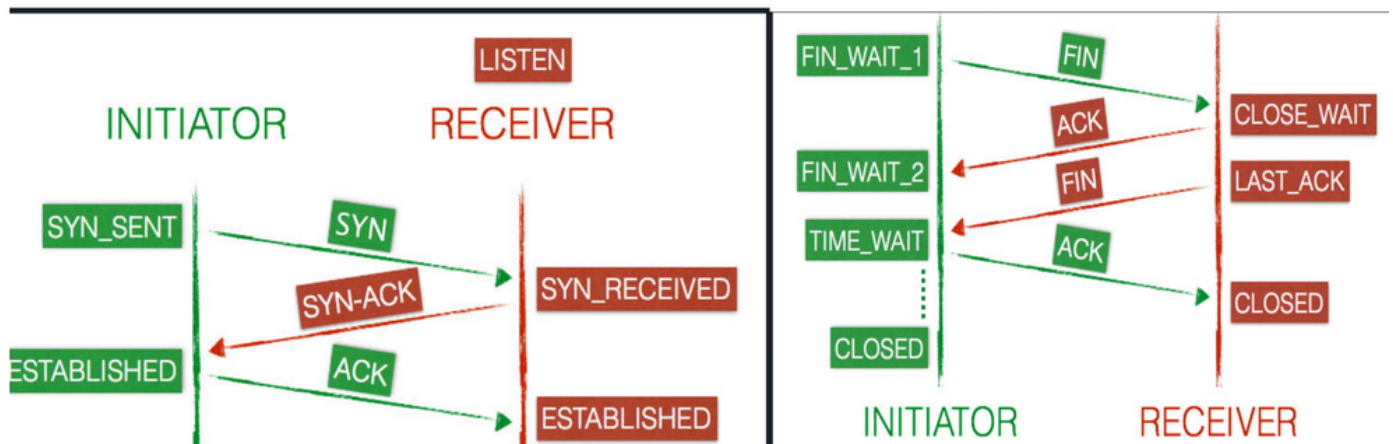
Note: Sunny Day Scenario: Rack1-Instance1-Primary, Instance2-Standby; Rack2-Instance1-StandBy, Instance2-Primary.

Rainy Day Scenario: Rack1-Instance 1 and Instance 2-Primary; Rack2-Instance 1 and Instance 2-StandBy.

TCP Connection Termination

The TCP Protocol is a connection-oriented protocol, which means that a connection is established and maintained until the application programs at each end have finished the exchange of messages. TCP works with the Internet Protocol (IP).

TCP handshake is also known as a 3-Way-Handshake. When a connection is initiated from the client machine to the server machine, the client and server exchange SYN and ACK packets before the data is transmitted.



Transmission Control Protocol :Client and Server Connection States

A connection progresses through a series of states all along its lifetime. The states are: LISTEN, SYN-SENT, SYN-RECEIVED, ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK, TIME-WAIT, and the fictional state CLOSED.

- When a new TCP connection is opened, the client (initiator) sends an **SYN** packet to the server (receiver) and updates its state to **SYN-SENT**.
- The server then sends an **SYN-ACK** in reply to the client which changes its connection state to **SYN-RECEIVED**.
- The client replies with an **ACK** and the connection is marked as **ESTABLISHED** on both end-point, now the client and the server are ready to transfer data.
- The client sends a **FIN** packet to the server and updates its state to **FIN-WAIT-1**.
- The server receives the termination request from the client and responds with an **ACK**. After the reply, the server gets into a **CLOSE-WAIT** state.
- As soon as the client receives the reply from the server, it then goes to the **FIN-WAIT-2** state.
- The server is still in the **CLOSE-WAIT** state and it independently goes with a **FIN**, which updates the state to **LAST-ACK**.
- Now the client receives the termination request and replies with an **ACK**, which results in a **TIME-WAIT** state.
- The server is now finished and sets the connection to **CLOSED** immediately.
- The client stays in the **TIME-WAIT** state for a maximum of four minutes, before the connection, is **CLOSED**.

Problem

Scenario 1. Geo-replication Checksum for Instance Id 1 has IPAM Cache and NRFMgmt Cache Checksum Mismatch

smfix1/smfix2 geo-replication status is failed (Inter rack replication to remote site failed).

ERROR: Admin command failed [pod internal-gr-pod-1, URL <http://X.X.0.0:15290/commands>] with Code 424, Message fail: replication checksum mismatch.

The issue was observed on 23rd August at 00:36:19 as "Inter rack replication failed".

From CEE alerts:

```
Inter_Rack_Replication 9ca45362a049 critical 08-23T00:36:19 System
Inter rack replication to Remote Site failed
```

From this CLI output, you can see instance-id 1 has Checksum Mismatch for IP Address Management (IPAM) and NRF Cache.

```
[cp0xxx-smf-ims/smfix1] smf# show georeplication checksum instance-id 1
```

```
Mon Sep 5 08:38:27.762 UTC+00:00
```

```
checksum-details
```

```
--  ---  -----
ID   Type           Checksum
--   ----  -----
1    ETCD          1662367102
IPAM  CACHE          1662367102
NRFMgmtCACHE  1662367102
```

```
[cp0xxx-smf-ims/smfix2] smf# show georeplication checksum instance-id 1
```

```
Mon Sep 5 08:38:30.767 UTC+00:00
```

```
checksum-details
```

```
--  ----  -----
ID   Type           Checksum
--   ----  -----
1    ETCD          1662367102
IPAM  CACHE          1661214831
NRFMgmtCACHE  1661214831
```

Scenario 2. Geo-replication Checksum for Instance Id 2 has ETCD Checksum Mismatch

```
[cp0xxx-smf-ims/smfix1] smf# show georeplication checksum instance-id 2
```

```
Mon Sep 5 08:38:37.852 UTC+00:00
```

```
checksum-details
```

```
--  ----  -----
ID   Type           Checksum
--   ----  -----
2    ETCD          1661214828
IPAM  CACHE          1662367107
NRFMgmtCACHE  1662367107
```

```
[cp0xxx-smf-ims/smfix2] smf# show georeplication checksum instance-id 2
```

```
Mon Sep 5 08:38:39.118 UTC+00:00
```

```
checksum-details
```

```
--  ----  -----
ID   Type           Checksum
--   ----  -----
2    ETCD          1662367107
IPAM  CACHE          1662367107
```

Scenario 3. TCP Connection Establishment Failure with Remote Site

Rack1-smfix1-logs:

From GR Pod logs, you can observe Update cache pod checkpoint is stopped, immediate Replication failed and No remote host is available.

```
2022/08/23 00:34:00.035 [ERROR] [grreplicationclient.go:201]
[gr_pod.geo_replication_client_stream.app] HandleImmediateReplication failed:
[RPCNoRemoteHostAvailable] No remote host available for this request
2022/08/23 00:34:02.086 [ERROR] [grreplicationclient.go:466]
[gr_pod.geo_replication_client_stream.app] Stream disconnected, closing
logQueueCounter=0xc0093b08b0
2022/08/23 00:34:04.124 [ERROR] [GeoAdminStreamClient.go:215] [gr_pod.geo_admin_client.app]
ADMIN(geo-admin-pod2) : exit outgoing request loop stream closed
2022/08/23 00:34:43.623 [ERROR] [grreplicationclient.go:270]
[gr_pod.geo_replication_client_stream.app] Update etcd checkpointing stopped for grinstance: 1
```

Rack2-smfix2-logs:

From GR Pod logs, you can observe Stream disconnected error and CACHE checksum difference is more than expected.

```
2022/08/23 00:34:06.497 [ERROR] [grreplicationserver.go:62]
[gr_pod.geo_replication_server_stream.app] Stream disconnected, closing
logQueueCounter=0xc001b85d08
2022/08/23 00:34:06.497 [ERROR] [grreplicationserver.go:314]
[gr_pod.geo_replication_server_stream.app] handleCachePodSyncRequests : Stream closed of
connection=0xc002ee08f0
2022/08/23 00:34:56.751 [ERROR] [grpodcommands.go:455] [gr_pod.cli_command.app]
compareChecksumData: CACHE checksum difference is more then expected, local checksum
[1661214831] remote checksum [1661214892]
2022/08/23 00:34:56.678 [ERROR] [etcdAuditReplHandler.go:196] [gr_pod.application.app]
SyncETCDData periodic sync : For ETCD [C.GR.1.] key, the remote site data size is: [10833]
2022/08/23 00:36:56.757 [ERROR] [grpodcommands.go:455] [gr_pod.cli_command.app]
compareChecksumData: CACHE checksum difference is more then expected, local checksum
[1661214831] remote checksum [1661215012]
```

Scenario 4. DIMM Error Observed on Server which Hosts Master Node

ECC error is seen on the master-1 node which hosts geo-replication-pod-0 around the same time as the stream disconnected error.

```
CP0XXX-Server9-02# scope sel
CP0XXX-Server9-02 /sel # show entries
Time Severity Description
-----
2022-08-23 00:33:59 UTC Informational "DDR4_P1_E1_ECC: Memory sensor, read 1 correctable ECC
errors on CPU1 DIMM E1 was asserted"
2022-08-22 22:59:45 UTC Informational "DDR4_P1_E1_ECC: Memory sensor, read 1 correctable ECC
errors on CPU1 DIMM E1 was asserted"
```

- Communication between the Geo-replication-pod on Rack1 and Geo-replication-pod on Rack2 is broken.
- DIMM Error occurs on one of the master nodes which caused the stream connection to go down between Rack1 and Rack2.

- From Rack1 Geo-replication-pod was not able to replicate or send any request to Rack2, it comes out with the error Remote Host not available.
- From netstat command output on Rack1 and Rack2 for the 7002 port found that the Rack1 socket is stuck in FIN_WAIT1 state and the Rack2 socket is stuck in SYN_RECV state.
- On the Server side, that is, on Rack2, the socket is stuck in the SYNC_RECV state, and the newly created connection also goes in the SYNC_RECV state and is not able to communicate with each other.
- The connection is in SYN_RECV state because the kernel has received an SYN packet for a port, that is, in LISTENING mode, but the other end did not reply with ACK.

smfix2-Master-2 has geo external VIP (Y.Y.Y.Y:7002) installed but the Remote host (SMFIX1) TCP connection state is stuck in the SYN_RECV state instead of the ESTABLISHED state. a.b.c.d and a.b.c.e are Master-1 and 2 IP'S of smfix1 (Rack1).

```
user@cp0xxx-smf-ims-master-2:~$ netstat -anp | grep 7002
tcp        0      0 Y.Y.Y.Y:7002        0.0.0.0:*            LISTEN      -
tcp        0      0 Y.Y.Y.Y:7002        a.b.c.e:35542        SYN_RECV    -
tcp        0      0 Y.Y.Y.Y:7002        a.b.c.d:47046        SYN_RECV    -
tcp        0      0 Y.Y.Y.Y:7002        a.b.c.e:36248        SYN_RECV    -
tcp        0      0 Y.Y.Y.Y:7002        a.b.c.d:42686        SYN_RECV    -
tcp        0      0 Y.Y.Y.Y:7002        a.b.c.e:38248        SYN_RECV    -
```

External Geo VIP TCP connection status on smfix1 (Rack1) for Remote peer is in FIN-WAIT1 state:

```
user@cp0xxx-smf-ims-master-1:~$ netstat -anp | grep 7002
tcp        0      0 a.b.c.d:60866       Y.Y.Y.Y:7002        FIN_WAIT1   -
tcp        0      1 a.b.c.d:52274       Y.Y.Y.Y:7002        FIN_WAIT1   -
tcp        0      1 a.b.c.d:59674       Y.Y.Y.Y:7002        FIN_WAIT1   -
tcp        0      1 a.b.c.d:47926       Y.Y.Y.Y:7002        FIN_WAIT1   -
```

Solution

Rack1:

- First, delete the standby Geo pod, wait for the pod to recover, and then delete the Active Geo pod. Log into Master VIP and delete the GR pod:

```
kubect1 delete pod <pod_name> -n <namespace>
```

Rack2:

- First, delete the standby Geo pod, wait for the pod to recover, and then delete the Active Geo pod.
- Check for the Geo-replication Status from CLI, post the deletion of Geo pods.

```
show georeplication-status
```

- Post the Geo pod deletion on Rack1 and Rack2, you can see the External Geo VIP IP: TCP port moves to the ESTABLISHED state.
- GeoReplacation Status "Pass".
- No checksum mismatch is seen in the replication status across the racks.

smfix2 (Rack2):

```
user@cp0xxx-smf-ims-master-1:~$ sudo netstat -anp | grep 7002 | grep -v aa
tcp        0      0 Y.Y.Y.Y:7002          0.0.0.0:*              LISTEN          36854
tcp        0      0 Y.Y.Y.Y:7002          a.b.c.d:46402         ESTABLISHED    36854/grpod
tcp        0      0 Y.Y.Y.Y:7002          1a.b.c.e:54708        ESTABLISHED    36854/grpod
tcp        0      0 Y.Y.Y.Y:7002          a.b.c.d:55152         ESTABLISHED    36854/grpod
tcp        0      0 Y.Y.Y.Y:7002          a.b.c.e:46530         ESTABLISHED    36854/grpod
tcp        0      0 10.59.0.0:7002        10.59.0.0:46532      ESTABLISHED    36854/grpod
```

smfix1 (Rack1):

```
user@cp0xxx-smf-ims-master-1:~$ sudo netstat -anp | grep 7002 | grep -v aa
tcp        0      0 a.b.c.d               0.0.0.0:*              LISTEN          53932/grpod
tcp        0      0 a.b.c.d:46530         Y.Y.Y.Y:7002         ESTABLISHED    53932/grpod
tcp        0      0 a.b.c.d:46402         Y.Y.Y.Y:7002         ESTABLISHED    53932/grpod
tcp        0      0 17 a.b.c.d:46532      Y.Y.Y.Y:7002         ESTABLISHED    53932/grpod
```

2. Geo-replication status:

```
[okcp0xx-smf-ims/smfix1] smf# show georeplication-status
result "pass"
[okcp0xx-smf-ims/smfix2] smf# show georeplication-status
result "pass"
```