Troubleshoot AppQoE TCP Optimization and DRE in Cisco Catalyst SD-WAN

Contents

Introduction

Background Information

Purpose, Scope, and Safety

Start Here: 10-minute Triage

- 1. Verify AppQoE State and Recent Errors
- 2. Verify Policy and Service-node Assignment in Both Directions
- 3. Inspect One Known Flow and Trace it End-to-end
  - Tracing a Single Flow
- 4. Verify Service-node Resources and DRE
- 5. Capture Traffic between Service Controller (SC) and Service Node (SN)
  - Method 1: CLI (Embedded Packet Capture — EPC)
  - Method 2: GUI (Cisco SD-WAN Manager)
- 6. Choose the Next Section

How TCP Optimization and DRE Process a Flow

No Optimization or Diversion

- Confirm
- Interpret
- Correct and Verify

Diversion Failure and Bypass

Drops after Diversion

- Confirm
- Interpret
- Correct and Verify

Service-node Health and Assignment

CFT and Capacity

SYN Policer and Connection Rate

MTU and MSS

- Confirm

DRE is Active but Reduction is Weak

- Confirm DRE State and Peers
- Measure Correctly

Evidence Collection and Escalation

Related Cisco Documentation

Introduction

This document describes how to troubleshoot AppQoE TCP Optimization and DRE in Cisco Catalyst SD-WAN.

Background Information

Use this guide when an AppQoE-enabled TCP flow is not optimized, is bypassed, resets after diversion, reports an unhealthy service node, or shows fewer Data Redundancy Elimination (DRE) reduction than expected.

Start with the 10-minute triage. It separates policy and path problems from service-node health, flow-state, capacity, TCP transport, and DRE-effectiveness problems. Continue to a symptom section only after you know where the flow stops behaving as expected.

Purpose, Scope, and Safety

This guide covers Cisco IOS® XE Catalyst SD-WAN devices using TCP optimization or DRE with integrated or external AppQoE service nodes.

Validation Item	Status
Public behavior and operational CLI	Aligned with the current Cisco Catalyst SD-WAN AppQoE 26.x documentation
Internal counter semantics	Rechecked against current Cisco IOS XE source on 2026-07-15
Exact release, platform, and topology used for publication	Lab capture: Cisco IOS XE Catalyst SD-WAN 17.18.2 on C8000v, external service-node AppQoE. AppNav controller c8000v-appqoe-3, service node c8000v-appqoe-4, external SN appqoe-service-node (SN IP 15.15.15.2). SNG SNG-APPQOE, data policy _vpn-10_appqoe-policy (TCP + DRE) on VPN 10. Captured 2026-07-15.

Command availability and output vary by software release, platform, and role. Confirm syntax with command help on the target device. The low-level QFP commands in this guide are read-only, but they are platform-and release-specific; use them only when the command is present.

Key release checkpoints are shown below; they do not replace platform-specific support and scale documentation.

Capability	Minimum Release Checkpoint
DRE and automated service-controller/service-node MTU handling	Cisco IOS XE Catalyst SD-WAN 17.5.1a
Enhanced AppQoE troubleshooting and sub-service health	17.6.1a

Capability	Minimum Release Checkpoint
Expanded flow-detail troubleshooting	17.9.1a
SSL proxy with TLS 1.3	17.13.1a/Manager 20.13.1
DRE through configuration groups	17.14.1a/Manager 20.14.1

DRE requires supported service nodes at both ends and symmetric flow handling. It does not run on a device in service-controller-only role, and AppQoE cannot be combined with Packet Duplication on the same connection. Encrypted traffic requires the supported SSL/TLS handling if its payload is to be optimized.

Before changing policy, clearing statistics, restarting a service, or enabling debug, capture:

- Software release, platform, and AppQoE role at both ends
- Source and destination IP addresses, ports, protocol, VPN, and UTC test time
- Expected policy sequence and service-node assignment
- Whether the symptom affects one flow, one site pair, or all AppQoE traffic
- Two counter snapshots over the same test interval



Note: Do not clear the DRE cache during initial troubleshooting. Clearing it restarts DRE and destroys the warm cache, which changes the condition being measured.

Start Here: 10-minute Triage

1. Verify AppQoE State and Recent Errors

Run on the participating edge devices or service controllers:

```
show sdwan appqoe status
show sdwan appqoe error recent
```

Lab sample:

c8000v-appqoe-4 (service node, C8000v, IOS XE 17.18.2).

```
c8000v-appqoe-4#show sdwan appqoe status
APPQOE Status : YELLOW
Service Status:
  SSLPROXY : YELLOW
  TCPPROXY : GREEN
```

```
  SERVICE CHAIN : GREEN
  RESOURCE MANAGER : GREEN

c8000v-appqoe-4#show sdwan appqoe error recent
Appqoe Statistics Recent
------------------------
Label                        Current value        Value(30 sec bfr)      Value(60 sec bfr)
RM TCP used sessions                 0                    0                    0
RM TCP session allocated             21516                21516                21516
TCP number of connections            21215                21215                21215
TCP failed connections               298                  298                  298
vPath drop due to pps                0                    0                    0
vPath new connection failed          0                    0                    0
BBR Active connections               1                    1                    1
Syn Drop Max PPS Reached             0                    0                    0
... (output truncated)
```

Here overall status is YELLOW only because the SSL AO is not in use (SSL proxy is in clear mode); TCP, service-chain, and resource-manager subservices are GREEN. No PPs drops or new-connection failures.

Look for the enabled services, current service-node state, recent flow errors, and any reason that directly explains bypass or drop behavior.

2. Verify Policy and Service-node Assignment in Both Directions

```
show sdwan policy from-vsmart
show service-insertion type appqoe service-node-group
```

Lab sample:

c8000v-appqoe-3 (AppNav controller), 2026-07-15.

```
c8000v-appqoe-3#show sdwan policy from-vsmart
from-vsmart data-policy _vpn-10_appqoe-policy
 direction all
 vpn-list vpn-10
  sequence 1
   match
    source-ip 31.31.31.0/24 41.41.41.0/24
   action accept
    tcp-optimization
    dre-optimization
    service-node-group SNG-APPQOE
  default-action accept
from-vsmart lists vpn-list vpn-10
 vpn 10

c8000v-appqoe-3#show service-insertion type appqoe service-node-group
Service Node Group name         : SNG-APPQOE
    Service Context             : appqoe/1
    Member Service Node count   : 1
```

```
Service Node (SN)                   : 15.15.15.2
Auto discovered                     : No
SN belongs to SNG                   : SNG-APPQOE
Current status of SN                : Alive
System IP                           : 10.20.0.1
Site ID                             : 30
Time current status was reached     : Fri Jun 12 07:45:14 2026

Cluster protocol VPATH version               : 2 (Bitmap recvd: 3)
Cluster protocol incarnation number          : 3

Health Markers:
              AO       Load State
             tcp           GREEN 0%
             ssl  RED/NOT AVAILABLE
             dre           GREEN 0%
            http  RED/NOT AVAILABLE
        utd chnl  RED/NOT AVAILABLE
```

The action accept carries both tcp-optimization and dre-optimization pointing at SNG-APPQOE, and the SN is alive with tcp/dre GREEN. ssl/http/utd show RED/NOT AVAILABLE because those AOs are not configured — expected for a TCP+DRE-only test.

Confirm that the intended sequence matches both directions of the test flow, includes the expected TCP/DRE actions, and points to the intended service-node group. DRE requires both ends and symmetric flow handling.

3. Inspect One Known Flow and Trace it End-to-end

```
show sdwan appqoe flow vpn-id <vpn-id> server-port <port>
show sdwan appqoe flow flow-id <flow-id>
show sdwan appqoe flow closed all
```

Tracing a Single Flow

First, run **show sdwan appqoe flow vpn-id <vpn-id> server-ip <server-ip> server-port <port>** on both the edge and DC routers. This command returns the flow ID. Once you have the flow ID, run **show sdwan appqoe flow flow-id <flow-id>** on both routers.

```
show sdwan appqoe flow vpn-id <vpn-id> server-ip <server-ip> server-port <port>
show sdwan appqoe flow flow-id <flow-id>
```

Lab sample:

c8000v-appqoe-4 (service node), 2026-07-15. No flows were active at capture time, so the historical

(closed) table is shown.

```
c8000v-appqoe-4#show sdwan appqoe flow all

Active Flows: 0

T:TCP, S:SSL, U:UTD, D:DRE, H:HTTP
Flow ID      VPN  Source IP:Port          Destination IP:Port          Service
No Matching Flows

c8000v-appqoe-4#show sdwan appqoe flow closed all
Current Historical Optimized Flows: 100

Optimized Flows
---------------
T:TCP, S:SSL, U:UTD, D:DRE, H:HTTP
RR: DRE Reduction Ratio

Flow ID      VPN  Source IP:Port         Destination IP:Port        Service  RR%
91989394759551 10   41.41.41.2:50748       185.125.190.99:80          T        -
91990285862945 10   41.41.41.2:36804       185.125.190.100:80         T        -
91996708679695 10   41.41.41.2:54614       91.189.91.97:80            T        -
92002614507991 10   41.41.41.2:57534       91.189.91.97:80            T        -
92101839402141 10   41.41.41.2:36856       185.125.188.54:443         T        -
... (95 more rows truncated)

++++ Tracing single flow end to end, run below command on both edge and DC routers +++++++
================================================================================================

c8000v-appqoe-4#show sdwan appqoe flow vpn-id 10 server-ip 41.41.41.2 server-port 21
T:TCP, S:SSL, U:UTD, D:DRE, H:HTTP

Flow ID      VPN  Source IP:Port               Destination IP:Port           Service
93741048628578 10   31.31.31.2:37632             41.41.41.2:21                 TD

c8000v-appqoe-4#show sdwan appqoe flow flow-id 93741048628578
Flow ID: 93741048628578

VPN: 10  APP: 0 [Client 31.31.31.2:37632 - Server 41.41.41.2:21]

HTTP Connect: 0
TCP stats
---------
Client Bytes Received   : 213
Client Bytes Sent       : 363
Server Bytes Received   : 176
Server Bytes Sent       : 36

Client Bytes sent to SSL: 165
Server Bytes sent to SSL: 176

... (195 more rows truncated)

TCP Flow Events
   1. time:303.932637   ::   Event:TCPPROXY_EVT_FLOW_CREATED
   2. time:303.932696   ::   Event:TCPPROXY_EVT_AD_RX_SYN_WITH_OPTIONS
   3. time:303.932741   ::   Event:TCPPROXY_EVT_SYNCACHE_ADDED
   4. time:303.932759   ::   Event:TCPPROXY_EVT_AD_TX_CORE_SYNACK
   5. time:303.933496   ::   Event:TCPPROXY_EVT_AD_RX_CORE_ACK_WITH_OPTIONS
   6. time:303.933594   ::   Event:TCPPROXY_EVT_ACCEPT_DONE
```

```
   7. time:303.933650   ::   Event:TCPPROXY_EVT_AD_TX_CORE_SYN_NO_OPTIONS
   8. time:303.933657   ::   Event:TCPPROXY_EVT_CONNECT_START
   9. time:303.933993   ::   Event:TCPPROXY_EVT_AD_RX_CORE_SYNACK
  10. time:303.934022   ::   Event:TCPPROXY_EVT_AD_TX_CORE_ACK_NO_OPTIONS
  11. time:303.934024   ::   Event:TCPPROXY_EVT_CONNECT_DONE
  12. time:303.934049   ::   Event:TCPPROXY_EVT_FLOW_CREATE_DRE_SENT
  13. time:303.934198   ::   Event:TCPPROXY_EVT_FLOW_CREATE_DRE_RSP_SUCCESS
  14. time:303.934222   ::   Event:TCPPROXY_EVT_FLOW_CREATE_SSL_DONE
  15. time:303.934232   ::   Event:TCPPROXY_EVT_DATA_ENABLED_SUCCESS

... (95 more rows truncated)
```

Service = T means these flows were TCP-optimized; RR% is blank because the DRE reduction ratio is reported per DRE-optimized flow (see the DRE sections). Use **flow flow-id <id>** on a live flow in order to read its recorded optimized/bypass status directly. The traced flow above shows Service = TD (TCP + DRE) and a complete proxy event sequence — SYN options exchange, accept/connect, DRE flow-create success, and data enabled — confirming full end-to-end optimization.

Classify the flow as optimized, bypassed/passthrough, or failed. Prefer the recorded status of the flow or passthrough reason over an inference from one aggregate counter.

4. Verify Service-node Resources and DRE

Run the commands that apply to the device role:

```
show sdwan appqoe rm-resources
show sdwan appqoe dreopt status detail
show sdwan appqoe dreopt statistics detail
show sdwan appqoe dreopt statistics peer
```

Lab sample:

c8000v-appqoe-4 (service node), 2026-07-15. Long DRE output trimmed to the health/capacity fields.

```
c8000v-appqoe-4#show sdwan appqoe rm-resources
============================================================
              RM Resources
============================================================
RM Global Resources :
 System Memory Status        : GREEN
 Num sessions Status         : GREEN
 Overall HTX health Status   : GREEN
Registered Service Resources :
TCP Resources:  Max Sessions : 40000   Used Sessions : 0
SSL Resources:  Max Sessions : 40000   Used Sessions : 0
DRE Resources:  Max Sessions : 750     Used Sessions : 0

c8000v-appqoe-4#show sdwan appqoe dreopt status detail
DRE ID              : 52:54:dd:77:4a:a7-019cdaae7bca-9a025f66
DRE uptime          : 126:13:46:58
```

```
Health status       : GREEN
DRE cache status    : Active
Disk cache usage    : 29%
Disk latency        : 2 ms
Active alarms:      None
Configuration:
  Profile type            : S
  Maximum connections     : 750
  Disk size               : 60 GB
  Compression type        : DRE-LZ

c8000v-appqoe-4#show sdwan appqoe dreopt statistics detail
Total connections                  : 48
Max concurrent connections         : 2
Current active connections         : 0
Total original bytes               : 63254 MB
Total optimized bytes              : 32691 MB
Overall reduction ratio            : 48%
Disk size used                     : 29%
Cache details:
  Cache status  : Active   Cache Size : 59132 MB   Cache used : 29%
... (per-connection reset/EBP/encode/decode detail truncated)

c8000v-appqoe-4#show sdwan appqoe dreopt statistics peer
  Peer No.  System IP          Hostname    Active connections    Cummulative connections
-----------------------------------------------------------------------------------------
         0  10.30.0.1        c8000v-app                     0                         22
         1  10.20.0.1        appqoe-ser                     0                         26
```

Health GREEN, no alarms, disk latency 2 ms, and a healthy 48% overall reduction ratio. The peer table confirms both DRE peers are reachable and version-compatible (see **aoim-statistics**).

Check health, maximum and active connections, peer compatibility, cache state, disk latency or alarms, and original-versus-optimized byte deltas.

5. Capture Traffic between Service Controller (SC) and Service Node (SN)

Take a monitor capture on the tunnel interface between the SC and SN. It provides clear, non-encapsulated data.

Method 1: CLI (Embedded Packet Capture — EPC)

You can use the Cisco IOS XE Embedded Packet Capture (EPC) feature directly on the device CLI. Here is the step-by-step configuration using Tunnel2000000001 as the example:

```
monitor capture APPQOE_CAP interface Tunnel2000000001 both
monitor capture APPQOE_CAP match <ipv4 or any or access-list>
monitor capture APPQOE_CAP start
show monitor capture APPQOE_CAP
monitor capture APPQOE_CAP stop
monitor capture APPQOE_CAP export bootflash:appqoe_clear_data.pcap
```

Method 2: GUI (Cisco SD-WAN Manager)

Alternatively, you can perform this capture directly from the Cisco SD-WAN Manager (formerly vManage) GUI, which will automatically output a **.pcap** file for you to download:

1. Navigate to **Monitor > Devices**.
2. Choose your device (**c8000v-appqoe-4**).
3. Click **Troubleshooting** in the left pane.
4. Under the **Traffic** section, choose **Packet Capture**.
5. In the interface selection dropdown, choose the AppQoE tunnel interface (for example, **Tunnel2000000001**).
6. (Optional) Apply any Source/Destination IP or Port filters.
7. Click **Start** to begin the capture, and **Stop** when finished. The system will prepare a **.pcap** file for download.

6. Choose the Next Section

First Abnormal Result	Continue To
Policy does not match in both directions	No optimization or diversion
No healthy or eligible service node is assigned	Service-node health and assignment
Flow is bypassed or has a passthrough reason	Diversion failure and bypass
Existing flow resets or packets drop	Drops after diversion
Only new connections fail during a burst	SYN policer and connection rate
CFT/FID failures increase	CFT and capacity
TCP stalls and PMTU/MSS evidence is present	MTU and MSS
Flow is optimized but reduction is weak	DRE effectiveness

How TCP Optimization and DRE Process a Flow

With dual-ended TCP optimization and DRE, the original connection is represented by three TCP

connections:

```
Client <-- LAN leg --> Edge A proxy/SN <== overlay leg + DRE ==> Edge B proxy/SN <-- LAN leg --> Server
```

DRE compresses repeated data on the overlay leg. The far-end device reconstructs the original stream before forwarding it to the destination. An external-service-node topology adds service-controller-to-service-node redirection, but the same checkpoints remain: policy, path symmetry, service-node eligibility, flow diversion, peer compatibility, and DRE state.

Term	Meaning in this Guide
Optimized	The selected AppQoE service is active for the flow.
Bypass or passthrough	Traffic continues without the chosen AppQoE service; inspect the passthrough reason.
Drop	The packet does not continue; this is user-impacting and requires a drop/error correlation.
Fail-close flow state	After a flow has been inspected/diverted, some later diversion failures cannot safely fall back to ordinary bypass.
SC	Service Controller
SN/ISN/ESN	Service Node/Integrated Service Node/External Service Node
CFT	Connection Flow Table used to track flows and their feature state

No Optimization or Diversion

Confirm

1. Confirm the policy match and actions in both directions.
2. Confirm symmetric routing through the expected pair of AppQoE devices.
3. Confirm that the service-node group and site IDs are correct.
4. Confirm that DRE is deployed and healthy at both ends.
5. Inspect the status of the target flow or passthrough reason.

Interpret

- No policy match usually means classification, direction, VPN, or attachment is wrong.
- A one-sided match can enable TCP/DRE handling on one edge but cannot provide a valid two-ended DRE path.
- A flow can correctly bypass when the traffic is already optimized, is an unsupported flow type, is asymmetric, or matches an auto-bypass condition.

Correct and Verify

Correct policy, direction, node-group, site-ID, or routing issues. Then create a new TCP connection and verify the flow on both ends. Do not use an existing connection to validate a policy change.

Diversion Failure and Bypass

Use internal QFP statistics only after the supported AppQoE flow and error commands have narrowed the problem:

```
show platform hardware qfp active feature appqoe stats global
show platform hardware qfp active feature appqoe stats all
show platform hardware qfp active feature appqoe internal all
```

Compare two snapshots; these counters are cumulative.

Lab sample:

c8000v-appqoe-3 (AppNav controller), 2026-07-15. Healthy diversion: the SN index is Green and no drop-cause counters are climbing beyond the expected SN Unhealthy transients recorded earlier.

```
c8000v-appqoe-3#show platform hardware qfp active feature appqoe stats all
 APPQOE Feature Statistics:
 Global:
     ip-non-tcp-pkts: 1354682
     cft_handle_pkt: 0
     sdvt_divert_req_fail: 1374
     appqoe_svc_on_appqoe_vpn_drop: 0
     appqoe_sng_not_configured: 0
   SDVT Global stats:
     within SDVT syn policer limit: 71660
 SNG: 0 SN Index [0 (Green)], IP: 15.15.15.2, oce_id: 221252816
     APPNAV STATS: toSN 85540403 / 75619122643 fromSN 105667460 / 109985814214
                   NoFoDrop 0 / 0
   SDVT Count stats:
     Active Connections: 4
     decaps: 58388600   encaps: 47119306
   SDVT Packet stats:
     Divert packets / bytes   47119306 / 34139250226
     Reinject packets / bytes 58388600 / 52530512355
     Pkts dropped packets / bytes 10 / 690
   SDVT Drop Cause stats:
```

```
     Packets Dropped as SN Unhealthy: 10

c8000v-appqoe-3#show platform hardware qfp active feature appqoe internal all
 APPQOE Feature Internal:
     syn_policer_rate: 2700
     Cluster Type: External
     Service chnl health : Green
     TCP sub-chnl health : Green
     SSL sub-chnl health : Red
     DREOPT sub-chnl health : Green
 Service-Node-Group: 0
     Active SN Bitmask:  0x0000000000000001
     SN Table:
         Idx | Id  | Ver | Status | DP Status | msecs ago |  IP
           0 |   1 |   2 |  Green |     Green |     27707 |  15.15.15.2
```

cft_handle_pkt: 0 and appqoe_svc_on_appqoe_vpn_drop: 0 rule out CFT/FID failure and a recursive-VPN drop. The packets dropped as SN Unhealthy: 10 is a small historical count — correlate against the SN health transitions before treating it as active impact.

Drops after Diversion

Confirm

```
show sdwan appqoe error recent
show sdwan appqoe flow closed all
show sdwan appqoe status
show service-insertion type appqoe service-node-group
```

If present on the release and platform, compare **stats global** or **stats all** before and after one controlled reproduction.

Lab sample:

Healthy baseline, c8000v-appqoe-3 (controller), 2026-07-15. No fail-close drops are accruing; the SN is Alive/Green and error recent shows vPath drop due to PPs: 0 and vPath new connection failed: 0. The datapath drop-cause snapshot is the deciding evidence:

```
c8000v-appqoe-3#show platform hardware qfp active feature appqoe stats all | include Drop|Unhealthy|NoF
   SDVT Drop Cause stats:
     Packets Dropped as SN Unhealthy: 10
   NoFoDrop 0 / 0
```

Correlate the exact reproduction time against these deltas before labeling a reset as fail-close.

Interpret

A previously inspected/diverted flow can drop when the service node becomes unusable or a later AppNav redirect fails. This protects established proxy state; the original client-to-server connection cannot always be reconstructed transparently after a proxy path disappears. Service-chain flows can also drop when ordinary bypass violates chain processing.

Do not label every reset as fail-close. Correlate the exact test time with the flow error, service-node health transition, and counter delta.

Correct and Verify

Restore a stable eligible service node and correct the path or service-chain failure. Validate with a new TCP connection, then confirm that the relevant drop counter stops increasing.

Service-node Health and Assignment

Health is role and service-specific. Liveness, resource capacity, and the health of a particular Application Optimizer (AO) are related but not interchangeable.

State	Datapath Behavior to Expect
Green	Eligible for new and existing flows
Yellow	Existing inspected non-SYN traffic can continue; new SYNs are not diverted to that node. A FULL node is one possible Yellow condition
Red or Down	New/uncommitted flows normally bypass when allowed; already-inspected/fail-close flows can drop; service-chain flows can drop

Run:

```
show service-insertion type appqoe service-node-group
show sdwan appqoe status
show sdwan appqoe rm-resources
show sdwan appqoe dreopt status detail
```

Check node membership, site ID, liveness, AO health, load/capacity, DRE alarms, and peer reachability. Before Cisco IOS XE Catalyst SD-WAN Release 17.6.1a, subservice health detail can be limited; interpret older output accordingly.

Lab sample:

Healthy node, 2026-07-15. On the controller the SN is Alive with per-AO health markers; on the node the resource manager reports Green with headroom:

```
c8000v-appqoe-3#show service-insertion type appqoe service-node-group | begin Health
Health Markers:
              AO       Load State
             tcp           GREEN 0%
             ssl  RED/NOT AVAILABLE
             dre           GREEN 0%

c8000v-appqoe-4#show sdwan appqoe rm-resources | include Status|Max Sessions|Used Sessions
 System Memory Status        : GREEN
 Num sessions Status         : GREEN
 Overall HTX health Status   : GREEN
TCP Resources:  Max Sessions : 40000   Used Sessions : 0
DRE Resources:  Max Sessions : 750     Used Sessions : 0
```

Load is 0% and used sessions are well under the 40000/750 limits, so this node is neither full nor Yellow for capacity reasons. The Yellow overall status seen in show sdwan appqoe status traces only to the unused SSL AO.

When a node is full or Yellow because it is near its configured session capacity, distribute new connections, increase the supported AppQoE resource profile where the platform allows it, or add capacity. Do not assume that adding router DRAM changes the supported hardware flow scale.

CFT and Capacity

appqoe_cft_handle_pkt is an error counter. It increments when AppQoE cannot obtain a valid flow ID from CFT handling. A rising value is evidence of a CFT/FID handling failure; a low value relative to total traffic is not proof of saturation.

```
show platform hardware qfp active infrastructure cft status
show platform hardware qfp active feature appqoe stats global
show sdwan appqoe rm-resources
```

Interpret CFT state separately from service-node capacity:

Lab sample:

c8000v-appqoe-3 (controller), 2026-07-15. Long memory-element table trimmed.

```
c8000v-appqoe-3#show platform hardware qfp active infrastructure cft status
===================================== CFT 1/1 =====================================
```

```
CFT id: 0   CFT name: GLOBAL_CFT
General Parameters:
    Max flows: 1000000
    Number of buckets in CFT hash table: 7227108
Statistics:
    Total number of flows added                : 1424672
    Total number of flows removed              : 1424664
    Total number of currently allocated flows  : 8
... (per-feature memory element table truncated)
```

Total number of currently allocated flows: 8 against a 1,000,000 max means no table pressure, and cft_handle_pkt: 0 in the AppQoE stats confirms no FID-acquisition failure. A rising cft_handle_pkt — not table occupancy alone — is what points to a CFT/FID problem.

- CFT status identifies edge flow-table or flow-ID pressure/errors.
- **rm-resources** identifies AppQoE service-node session capacity.
- Full tables or memory pressure are possible causes of FID acquisition failure, but not the only causes.

If the error increases during the test interval, capture the CFT error/state, platform scale, active connections, and node resources. Reduce connection pressure, rebalance service nodes, change the supported resource profile, or move to a platform with the required scale. Engage Cisco TAC before treating a generic CFT error as a hardware-capacity conclusion.

SYN Policer and Connection Rate

Use the full status and documented counters. "SDVT_DROP_ERROR" is an error class; by itself it is not proof that the SYN policer fired.

```
show sdwan appqoe libuinet-statistics
show sdwan appqoe error recent
show sdwan appqoe rm-resources
```

Co-relate new-connection failures with Syn Drop Max PPs reached, vPath drop due to PPs, and the same test interval. Long-lived flows remaining healthy while only new SYNs fail strengthens the policer hypothesis.

Lab sample:

c8000v-appqoe-4 (service node), 2026-07-15. libuinet-statistics is long; the policer-relevant Vpath statistics block is shown.

```
c8000v-appqoe-4#show sdwan appqoe libuinet-statistics | begin Vpath Statistics
Vpath Statistics:
 Packets In                 : 112733521
 Syn Packets                : 21516
 Syn Drop Max PPS Reached   : 0
```

```
 Flow Info Allocs           : 21516
 Flow Info Allocs Failed    : 0
 Vpath drops due to min threshhold: 0
 Failed to create new connection: 0
```

Syn Drop Max PPs reached: 0 (and vPath drop due to PPs: 0 in error recent) rules the SYN policer out here. The configured rate on the controller is syn_policer_rate: 2700 (from internal all); compare it against your offered SYN rate before acting.

Reduce the burst rate if possible, distribute new connections across healthy capacity, and confirm that the documented policer counters stop increasing. Do not tune or bypass protective limits from a generic guide; use the scale guidance of the platform and TAC when sustained rates approach supported limits.

MTU and MSS

MSS adjustment is not, by itself, a direct AppQoE SYN-drop path. The datapath calculates an MSS from the service-node adjacency MTU and encapsulation overhead. When it can, it adjusts the client MSS and stores a server-side MSS; when MTU information is unavailable, it can continue without that adjustment.

Therefore, do not use sdvt_drop_appnav_divert alone as proof of an MSS calculation failure.

Confirm

- Record the software release; automated SC-to-SN MTU handling was introduced in 17.5.1a.
- Verify a uniform supported MTU across the service-controller/service-node path and the SD-WAN underlay.
- Use DF-bit path-MTU testing and synchronized SYN/SYN-ACK captures at the relevant edges.
- Compare offered and adjusted MSS values and check for fragmentation or black-holing.

Useful platform command where supported:

```
show platform hardware qfp active feature sdwan datapath session summary
```

Lab sample:

c8000v-appqoe-3 (controller), 2026-07-15.

```
c8000v-appqoe-3#show platform hardware qfp active feature sdwan datapath session summary
Src IP          Dst IP          Src Port Dst Port   Encap    Uidb     Bfd Discrim PMTU     Flags
------          ------          -------- -------    ------   -------  ----------- -----    -----
192.168.172.19  192.168.172.6   12346    12346      IPSEC    65528    20005       1442     0x0
192.168.172.19  192.168.172.5   12346    12366      IPSEC    65528    20009       1442     0x0
192.168.172.19  192.168.172.20  12346    12346      IPSEC    65528    20006       1442     0x0
```

The IPsec overlay sessions report a uniform PMTU 1442. A consistent PMTU across the SC/SN tunnels (and no black-holing in DF-bit tests) means MSS is being derived from a stable adjacency MTU — rule this out before touching tunnel MTU.

Correct the path MTU or MSS policy only after confirming the failing hop. Do not increase a tunnel MTU unless the complete underlay path can carry it.

DRE is Active but Reduction is Weak

Low reduction does not automatically mean DRE is broken. Unique first-pass data, a cold cache, already-compressed payloads, encrypted traffic without the required SSL/TLS handling, asymmetric flows, peer incompatibility, or auto-bypass can all produce little or no reduction.

Confirm DRE State and Peers

```
show sdwan appqoe dreopt status detail
show sdwan appqoe dreopt statistics detail
show sdwan appqoe dreopt statistics peer
show sdwan appqoe dreopt auto-bypass
show sdwan appqoe ad-statistics
show sdwan appqoe aoim-statistics
show sslproxy status
```

Lab sample:

c8000v-appqoe-4 (service node), 2026-07-15. DRE status/statistics/peer appear in the triage Step 4. sample above; the remaining commands:

```
c8000v-appqoe-4#show sdwan appqoe dreopt auto-bypass
c8000v-appqoe-4#
                              (empty — no flows in DRE auto-bypass)

c8000v-appqoe-4#show sdwan appqoe ad-statistics
 [Edge] AD Negotiation Start                : 21513
 [Edge] AD Negotiation Done                 : 21215
 [Edge] Rcvd SYN-ACK w/o AD options         : 21167
 [Core] AD Negotiation Start                : 55
 [Core] AD Negotiation Done                 : 55

c8000v-appqoe-4#show sdwan appqoe aoim-statistics
 Total Number Of Peer Syncs      : 2
 Total Passthrough Connections Due to Peer Version Mismatch   : 0
 LOCAL AO Statistics:  SSL 1.3 (Y),  DRE 0.23 (Y)
 PEER 10.20.0.1:  SSL 1.3 InCompatible=N,  DRE 0.23 InCompatible=N
 PEER 10.30.0.1:  SSL 1.3 InCompatible=N,  DRE 0.23 InCompatible=N

c8000v-appqoe-4#show sslproxy status
CA TP Label                    : PROXY-SIGNING-CA
```

```
Dual-Side Optimization         : TRUE
Min TLS Ver                    : TLS Version 1
Clear Mode                     : TRUE
```

Nothing is in auto-bypass, AD negotiation is completing, and aoim-statistics shows 0 passthroughs from version mismatch with both DRE peers marked incompatible = N. sslproxy status shows Clear Mode: TRUE, so HTTPS payloads traverse without decryption — expected weak DRE reduction on already - encrypted traffic unless SSL proxy is enabled. The measured 48% reduction (triage Step 4.) is real DRE benefit on the clear-text flows.

Verify:

1. DRE policy matches both directions at both ends.
2. The flow is symmetric and uses the expected peers.
3. DRE health and cache state are active with no relevant alarm.
4. Peer versions and AO capabilities are compatible.
5. Disk latency and resource utilization are within the supported range.
6. The flow is not in auto-bypass.
7. Encrypted traffic has the required SSL/TLS decryption and dual-ended optimization configuration.

Measure Correctly

Use one representative data set and the same time interval at both ends. Capture original and optimized byte counters before and after the test. Prefer interval deltas over a lifetime reduction ratio. If testing cache benefit, document whether the run is cold-cache or warm-cache and repeat the same content.

Evidence Collection and Escalation

Use supported operational commands first. If the cause remains unclear, collect:

- UTC reproduction window and affected tuple/VPN
- One failing and one known-good flow from the same site pair
- AppQoE status, recent errors, policy, service-node group, and flow output from both ends
- DRE status, peer statistics, resource state, and interval byte deltas
- CFT status and QFP AppQoE statistics when those commands exist
- Admin-tech captured before clearing counters or restarting services

show sdwan appqoe flow all debug is expanded show output, not a license to enable broad platform debugging. It can be expensive and can expose flow tuples; use it only for a short, scoped collection when targeted flow commands are insufficient.

Do not publish a **generic platform-debug** command without a validated release/platform, a capture duration, an output destination, and a tested stop procedure. Use Cisco TAC guidance for active debug or packet-trace collection on a busy production device.

Related Cisco Documentation

- AppQoE Verification and Troubleshooting
- TCP Optimization
- Traffic Optimization with DRE
- External Service Nodes for AppQoE Services
- Catalyst SD-WAN AppQoE DRE: Topology, Configuration, Verification
- Cisco Technical Support & Downloads