Cisco Customer Experience

NBAR and SD-AVC Operations and Troubleshooting in Cisco SDWAN

Luis Daniel Martinez Mercado
EMEA TAC SDWAN Lead – Technical Consultant Engineer
BRKENT-2336

cisco Live!

What is the struggle?

When implementing application classification

- Application visibility
- Application Firewall
- Traffic prioritization
- Transport selection

- We do not understand why traffic is misclassified
- If traffic is clasified properly, why is it misforwarded, not prioritized, or dropped?

Webex App

Questions?

Use the Webex app to chat with the speaker after the session

How

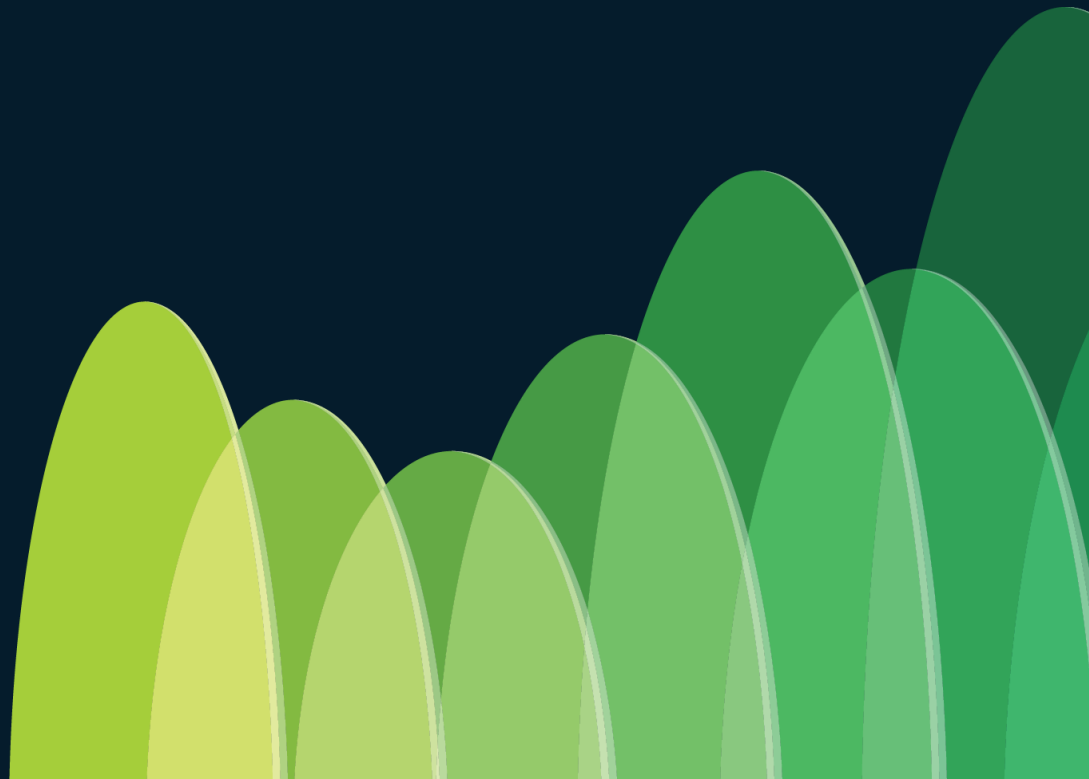
1. Find this session in the Cisco Events mobile app
2. Click “Join the Discussion”
3. Install the Webex app or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until February 28, 2025.

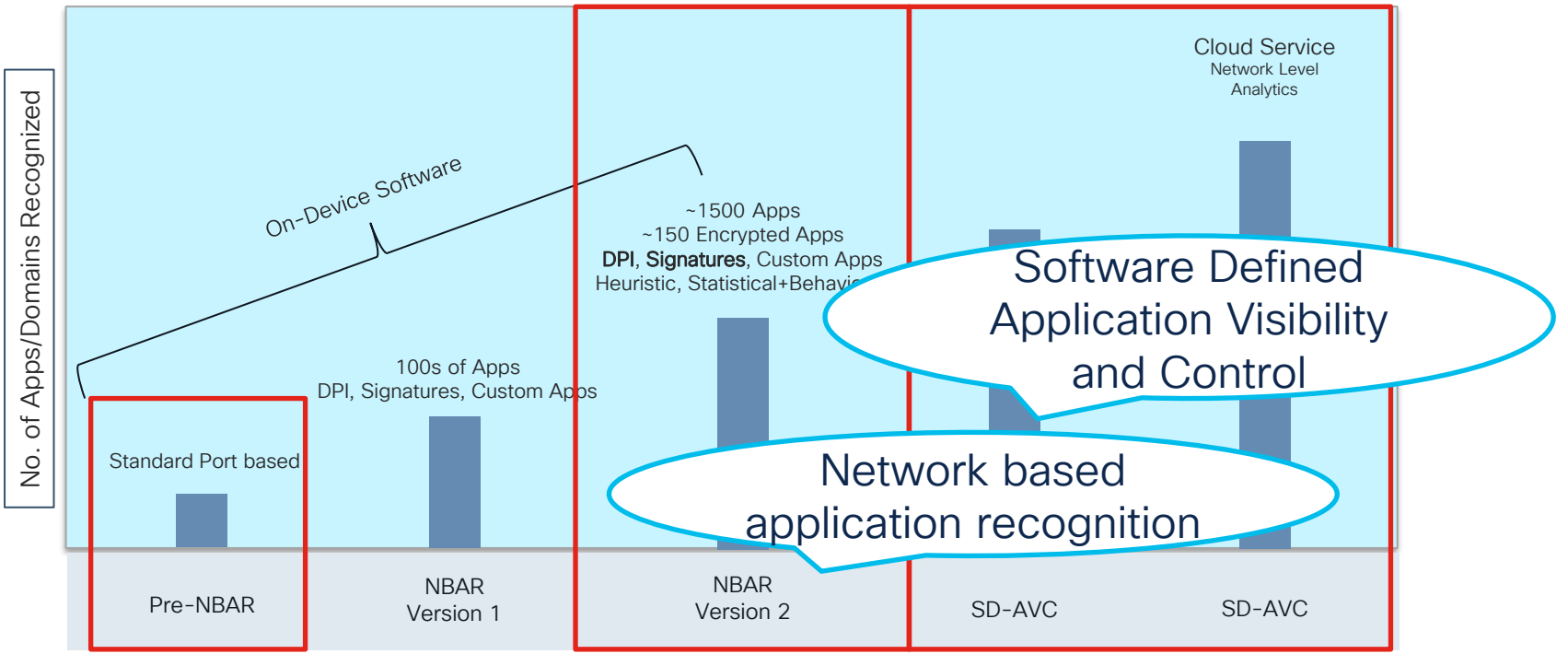
Agenda

- **Application Recognition**
  - On what basis are applications recognized?
  - Cisco SAIE
- NBAR Flow Processing
  - ✓ What is the tricky part?
- SD-AVC Integration and Order of precedence
  - ✓ What is the tricky part?
- Cisco SDWAN Manager role
- Key Takeaways

Application recognition



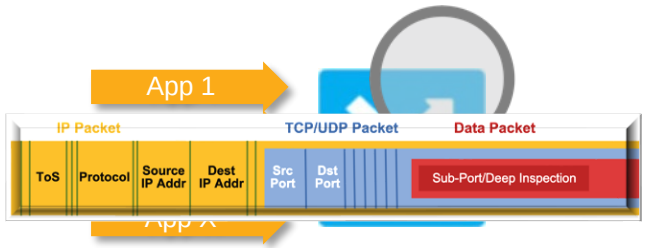
SDWAN Application Intelligence Engine



What does NBAR2 DPI do?

Examines and analyses the data payload in the packet and identifies application layer protocols by matching them against an **Application Signature**.

Deep Packet Inspection

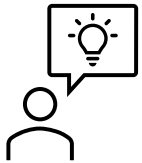


WAN Edge Router

What is an Application Signature?

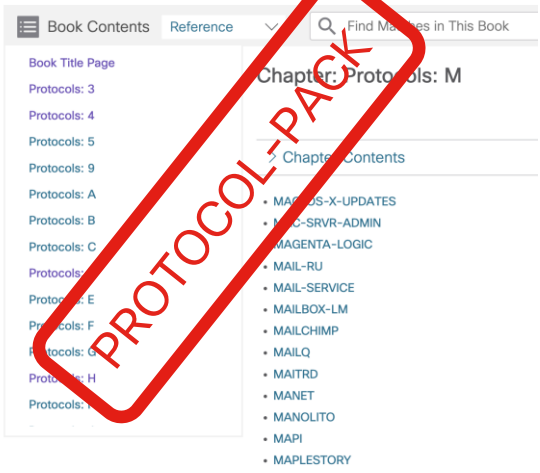
The specific traffic characteristics of a network application

What is a Protocol-Pack?



A collection of application signatures

~~Cisco NBAR2 Protocol Reference~~



MS-TEAMS

Name/CLI Keyword	ms-teams
Full Name	Microsoft Teams
Description	Microsoft Teams is a platform that combines workplace chat, meetings, notes, and attachments. The service integrates with the company's Office 365 subscription office productivity suite, including Microsoft Office and Skype.
Introduced	37.0.0
Deprecated	-
Reference	https://products.office.com/en-us/microsoft-teams/group-chat-software
Global ID	13:1208
ID	1208
IPv4 Support	Yes
IPv6 Support	Yes
Application Group	ms-cloud-group
Business Relevance	business-relevant
Category	voice-and-video
Sub Category	enterprise-media-conferencing
P2P Technology	True
Encrypted	True
Traffic-class	multimedia-conferencing
Tunnel	False

Verifying Protocol-pack version

Protocol-pack file naming convention,
pp-adv-<platform-type>-<OS>-<engine-id>-<protocol-pack-version>.pack

Default nbar version and protocol pack in ios-xe version 17.9.5a

```
CSRSD-WAN-1#show ip nbar version
NBAR software version:  46
NBAR minimum backward compatible version:  46
NBAR change ID:  BLD_NBAR_XE179_20231127_144921

Loaded Protocol Pack(s):
  Name:                           Advanced Protocol Pack
  Version:                        61.0
  Publisher:                      Cisco Systems Inc.
  NBAR Engine Version:            46
  State:                          Active
```

Example for output with protocol Pack updated through CLI:

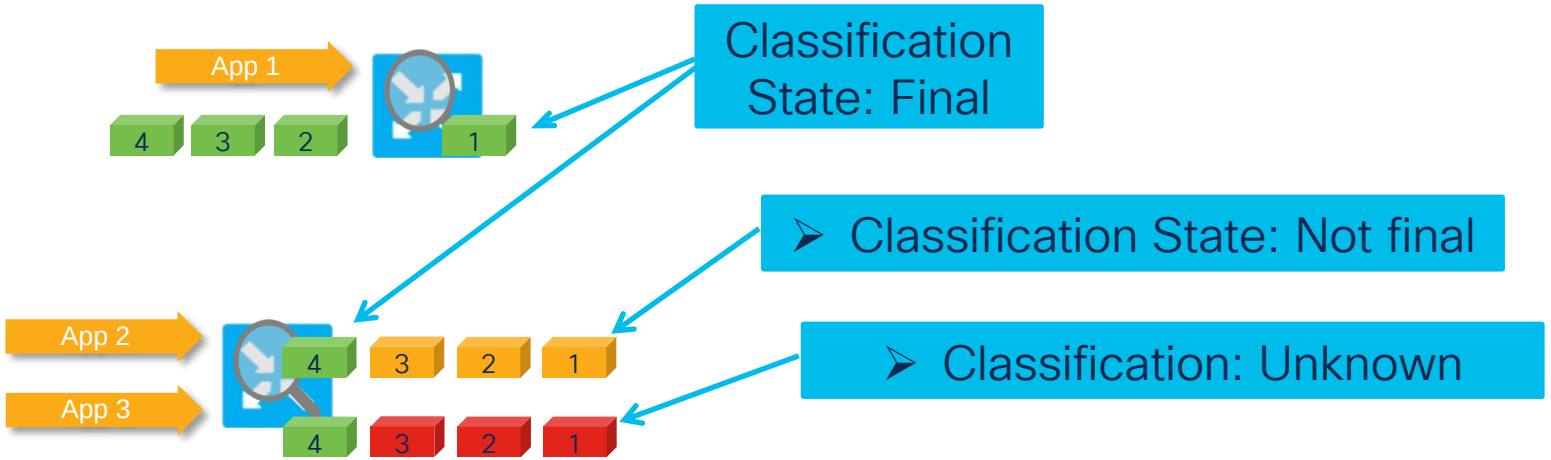
(config)#ip nbar protocol-pack bootflash:<path>/<filename>

```
cEdge#show ip nbar version
NBAR software version: 46
NBAR minimum backward compatible version: 46
NBAR change ID: BLD_NBAR_XE179_20230201_235301

Loaded Protocol Pack(s):
Name: Advanced Protocol Pack
Version: 67.0
Publisher: Cisco Systems Inc.
NBAR Engine Version: 46
Creation time: Thu Sep 7 16:17:25 UTC 2023
File: bootflash:/vmanage-admin/pp-adv-cat8k-179.1a-46-67.0.0.pack
State: Active
cEdge#
```

NBAR Flow processing

First Packet Match

We call **FPM** the capability of recognising and application flow successfully from the very first packet or **First in Flow (FIF)** packet.



What is the tricky part?

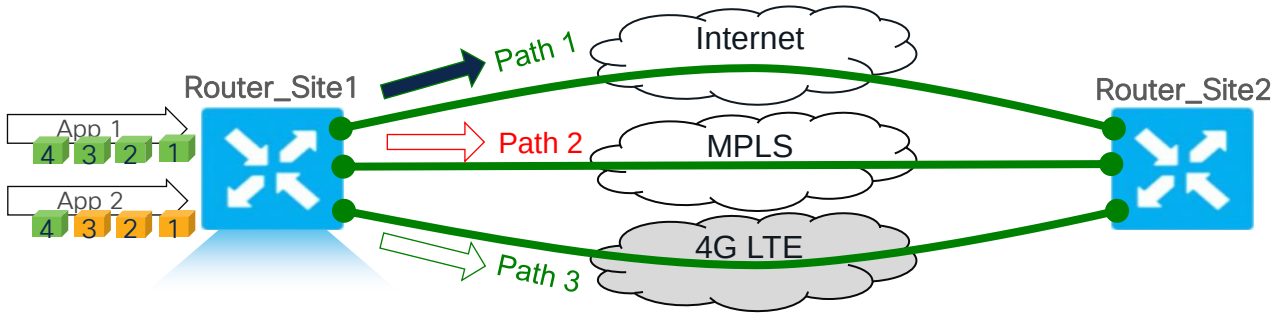
FPM dependant use cases:
- **Transport selection (Forwarding)**
- Application Firewall

App 1 and App 2 path must have latency <150ms and loss <2%

All traffic in the flow will follow the same path as the first packet

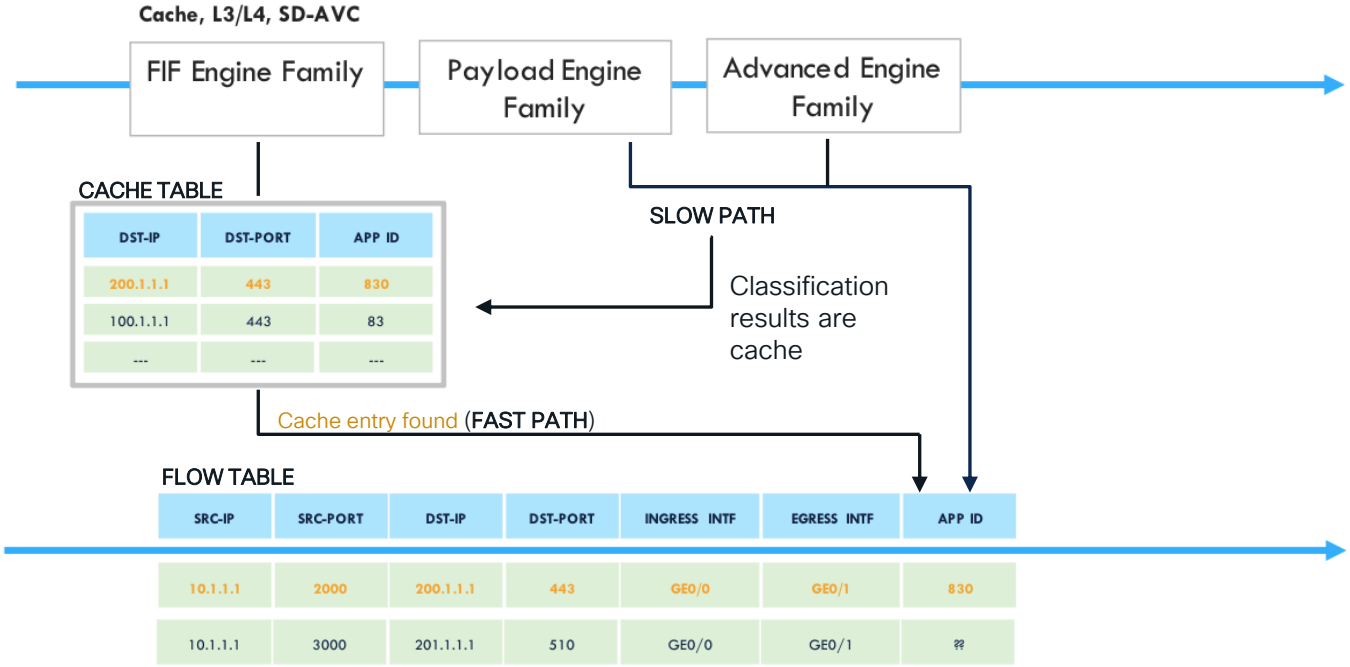
Flow Stickiness (default)

Flow Stickiness Disabled

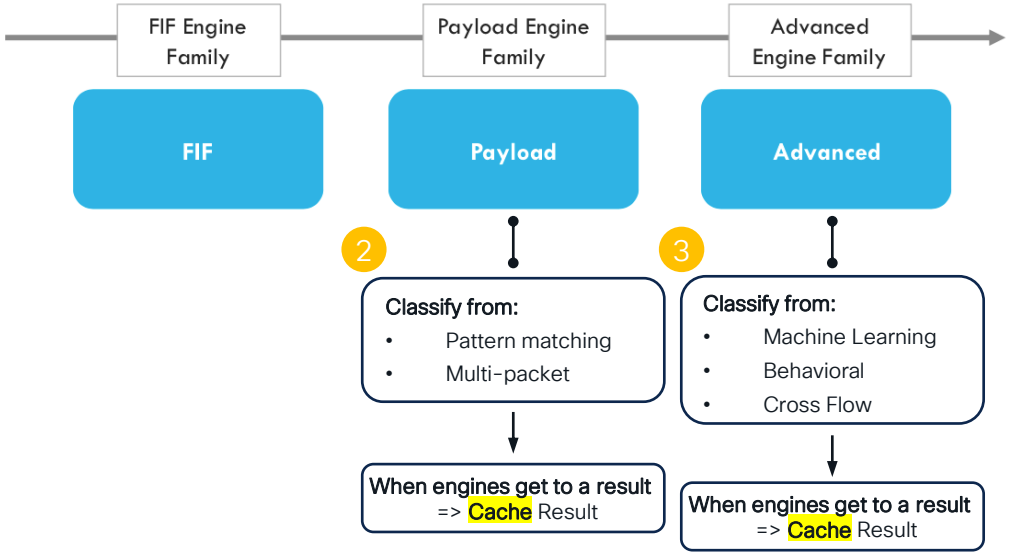


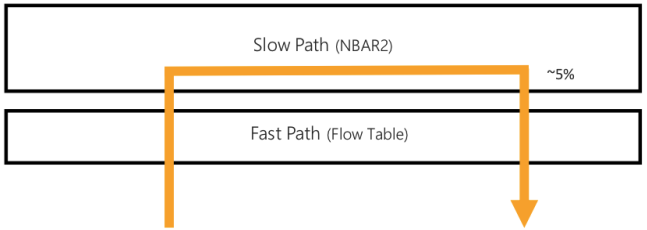
Path1: 10ms, 0% loss
Path2: 200ms, 3% loss
Path3: 140ms, 1% loss

NBAR2 Flow Processing – High Level View



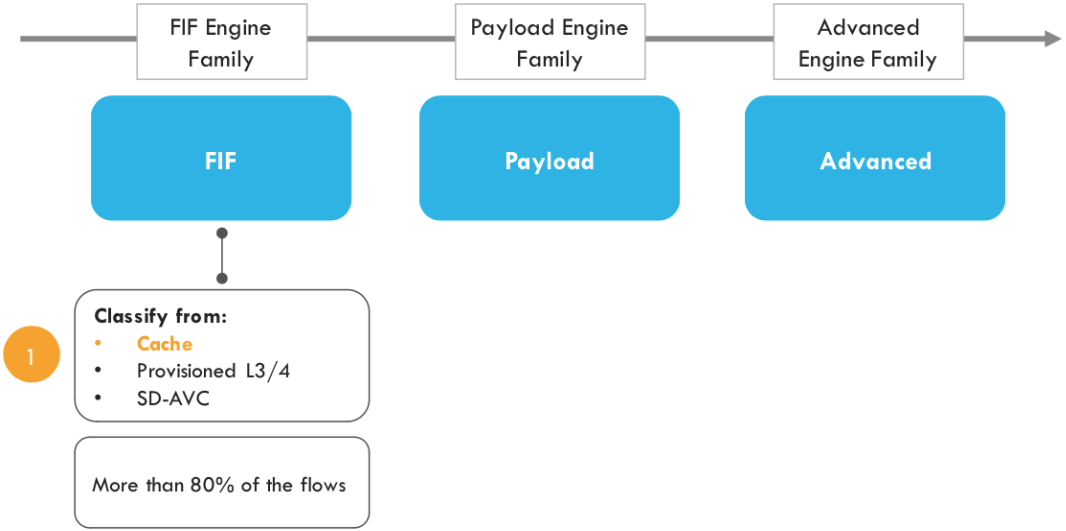
NBAR2 Classification – Slow Path

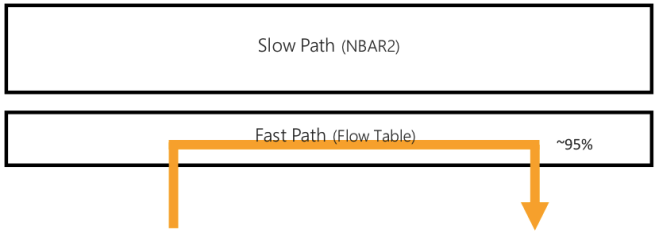




- Classifies the flow based on NBAR2 packet processing – Heavy duty work
- Potential classification on the first packet (FIF). If not, process more packets.
- Programs the Fast Path with classification results

NBAR2 Classification – Fast Path

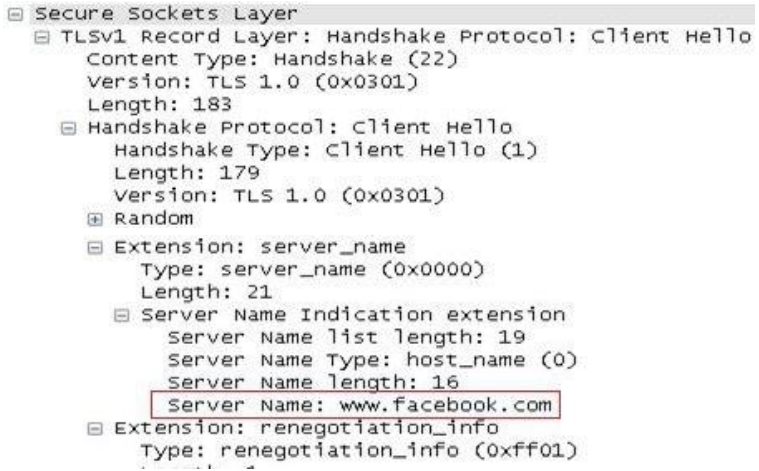




- Optimized forwarding path in router
- Programmed from the slow path
- Completely bypasses NBAR2 processing

NBAR2 Encrypted Traffic Classification

SSL handshake analysis – certificate, Server Name Indicator (SNI, RFC 6066)



DNS traffic analysis

To classify flows on the first packet, NBAR learns the information from dns requests that come before the actual flow.

For this to happen few basic conditions should be met:

1. Dns packets must be "seen" by router.
2. Domain-name must match NBAR relevant application.

NBAR will cache the IP address + matching classification in dns-learning-table so when the actual flow is opened it will be classified on the first packet.

Machine learning/Statistical classification and Heuristics based classification.

FIF classification and NBAR caches

DNS Cache (Learnt from DNS request and response)

show ip nbar classification dns learning cache <# of entries>

ip	vrf	application name	visibility	time to ttl expiry (sec)	ttl (sec)	entry aging (sec)	entry type	hit count	domain
31.13.66.35	1	facebook	facebook	0	23	60	PL DNS	7	www.facebook.com
157.240.229.35	1	facebook	facebook	0	60	60	PL DNS	10	www.facebook.com
151.101.131.52	1	time-news	time-news	0	979	60	PL DNS	0	time.com
151.101.0.81	1	bbc	bbc	0	29	60	PL DNS	0	www.bbc.com
172.253.62.94	1	google-services	google-services	0	211	60	PL DNS	4	www.gstatic.com
18.160.41.63	1	pocket	pocket	0	60	60	PL DNS	0	getpocket.com

Socket Cache (1-7 packets approx to finalize)

show ip nbar classification socket-cache <# of entries>

```
CSRSD-WAN-1#show ip nbar classification socket-cache 10
```

server ip	vrf	port	proto	application name	is valid	is black list	is learn phase	max time to expiry (sec)	entry type	hit count
172.253.115.95	1	443	TCP	google-services	No	No	Yes	1452	Infra	1
34.120.5.221	1	443	TCP	ssl	No	No	Yes	902	Infra	1
104.244.42.194	1	443	TCP	ssl	Yes	No	No	1480	Infra	13
185.125.190.58	1	123	UDP	ntp	No	No	Yes	338	Infra	1

FIF classification and DNS-Cache

We leverage **Packet-trace (Fia-trace)** captures to sniff into the packet's data-path and confirm **FIF and FPM functionality**:

If device does not see the DNS query and response,

```
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet3
    Output      : <unknown>
    Source      : 172.16.8.2
    Destination : 208.80.154.224
    Protocol    : 6 (TCP)
      SrcPort   : 54464
      DstPort   : 443
    ...
```

```
TCP  << flags
    Source Port         : 54464
    Destination Port    : 443
    Sequence Number     : 0x3a43a3f9
    ACK Number          : 0x00000000
    TCP flags           : 0xa002   (SYN packet)
    ...
Feature: NBAR
    Packet number in flow: 1
    Classification state: Not final
    Classification name: unknown
    Classification ID: 1 [CANA-L7:1]
    Candidate classification sources:
      N/A
```

```
TCP
    Source Port         : 54464
    Destination Port    : 443
    Sequence Number     : 0x3a43a3fa
    ACK Number          : 0x9b5930ac
    TCP flags           : 0x8018
    ...
Feature: NBAR
    Packet number in flow: 4
    Classification state: Final
    Classification name: wikipedia
    Classification ID: 1547 [CANA-L7:608]
    Candidate classification sources:
      N/A
    ...
```

FIF classification and DNS-Cache (Part 2)

No DNS cache entry is created

```
CSRSD-WAN-1#show ip nbar classification dns learning cache 100
|ip                                  |vrf  |application        |visibility |time to    |ttl       |entry|entry    |hit     |domain
|                                    |     |name               |           |ttl expiry|(sec)     |aging|type     |count   |
|                                    |     |                   |           |(sec)     |          |(sec)|         |        |
|------------------------------------|-----|-------------------|-----------|----------|----------|-----|---------|--------|--------
```

Socket cache entry is created after final classification

```
CSRSD-WAN-1#show ip nbar classification socket-cache 100
|server ip                           |vrf       |port |proto|application            |is    |is    |is    |max time |entry|hit    |
|                                    |          |     |     |name                   |valid|black|learn|to expiry|type |count  |
|                                    |          |     |     |                       |     |list |phase|(sec)    |     |       |
|------------------------------------|----------|-----|-----|-----------------------|-----|-----|-----|---------|-----|-------|
|34.120.208.123                      |         1|  443|TCP  |ssl                    |No   |No   |Yes  |1720     |Infra|1      |
|208.80.154.224                      |         1|  443|TCP  |wikipedia              |No   |No   |Yes  |1724     |Infra|1      |
```

FIF classification and DNS-Cache (Part 3)

What if device sees the DNS query and response?

ip	vrf	application name	visibility	time to ttl expiry (sec)	ttl (sec)	entry aging (sec)	entry type	hit count	domain
...									
208.80.154.224	1	wikipedia	wikipedia	278	300	60	PL DNS	1	www.wikipedia.org
...									

```
Path Trace
  Feature: IPV4(Input)
    Input       : GigabitEthernet3
    Output      : <unknown>
    Source      : 172.16.8.2
    Destination : 208.80.154.224
    Protocol    : 6 (TCP)
      SrcPort   : 54464
      DstPort   : 443
    ...
```

```
Feature: NBAR
   Packet number in flow: 1
   Classification state: Not final
   Classification name: wikipedia
   Classification ID: 1547 [CANA-L7:608]
   Candidate classification sources:
     L3-Cache: wikipedia [1547]
...
   TCP flags           : 0xa002 >>> (SYN packet)
```

Clearing NBAR caches

Socket cache and DNS cache can be cleared using the commands below.

Clear ip nbar classification socket-cache

Clear ip nbar classification dns learning cache

Flow Stickiness

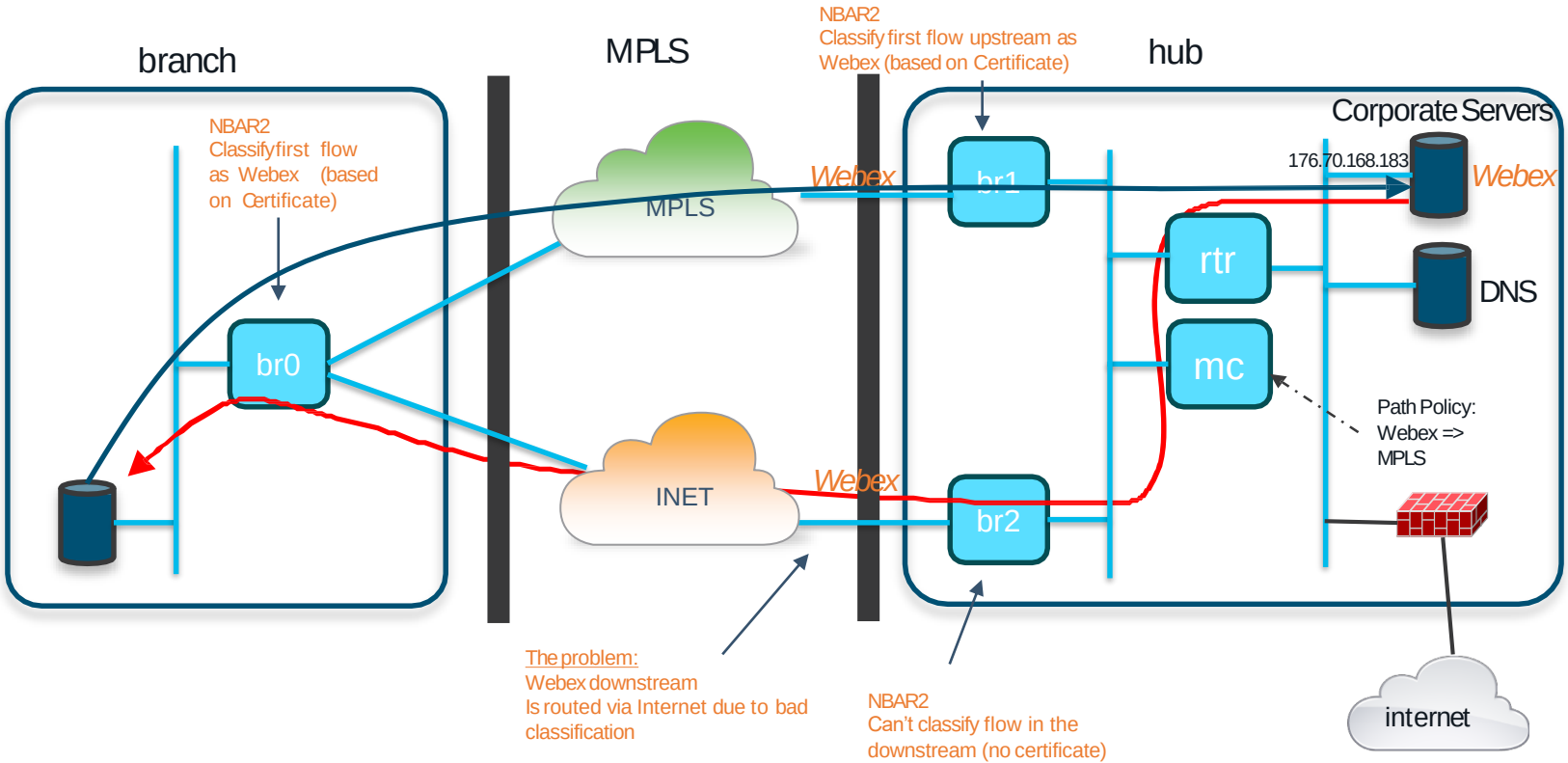
By default, flow stickiness is enabled on the SDWAN devices. This means that failing to classify the flow on the first packet will result on FPM-fail and flows, expected to match a policy sequence based on application list, may ended hitting a different sequence and pinned to and undesired path. Flow stickiness can be disabled with command below.

Device(config)# policy

Device(config-policy)# flow-stickiness disable

SD-AVC Integration and Order of precedence

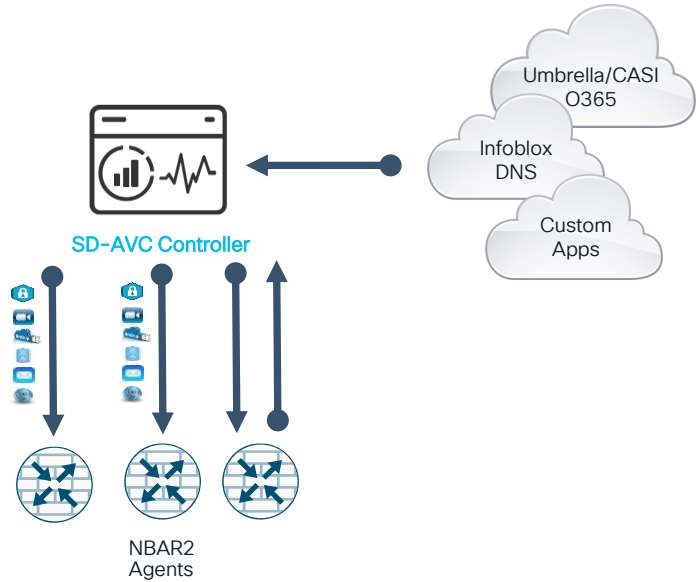
NBAR Cannot Handle Asymmetric Flows



Classification Benefits of SD-AVC

A centralised approach to the application recognition capabilities. Previously isolated routers running **NBAR2** will be now acting sensors and consumers of application classification data in our sdwan overlay.

- Network-level application recognition consistent across the network
- **Improved application recognition in symmetric and asymmetric routing environments**
- **Improved FPM (First packet match)**
- **External SaaS application feeds from vendors if Cloud connector is enabled (Microsoft, Webex, etc.)**
- **Allow to define Custom Applications**



SD-AVC elements

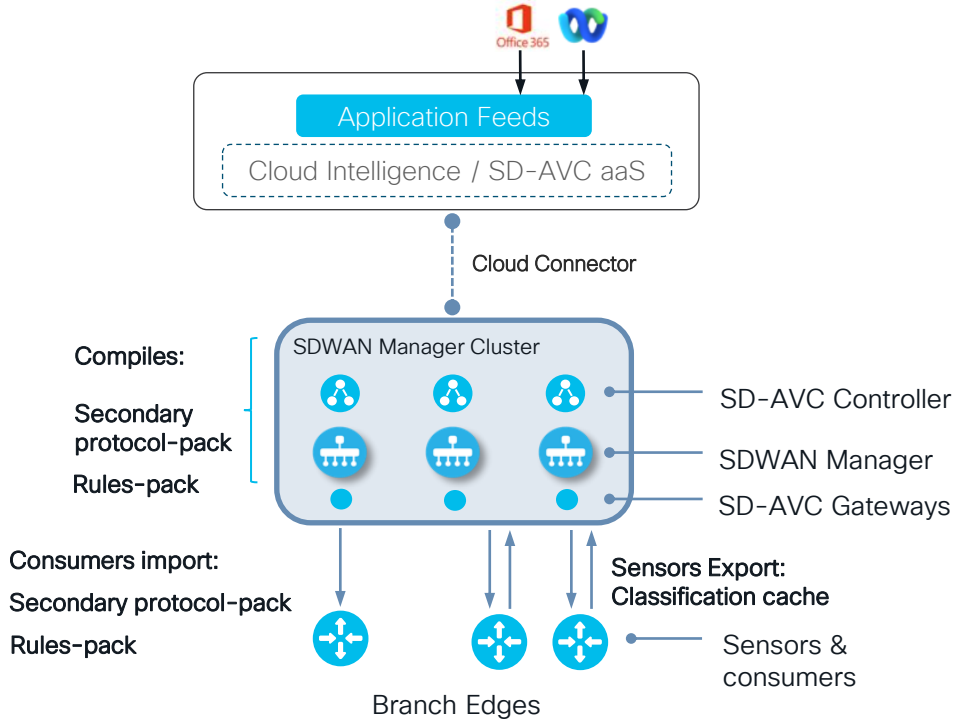
SD-AVC controller runs as a container on the SD-WAN Manager starting in version 18.4 with a few architectural changes through recent versions.

SD-AVC defines Sensors and Consumers in the network data plane

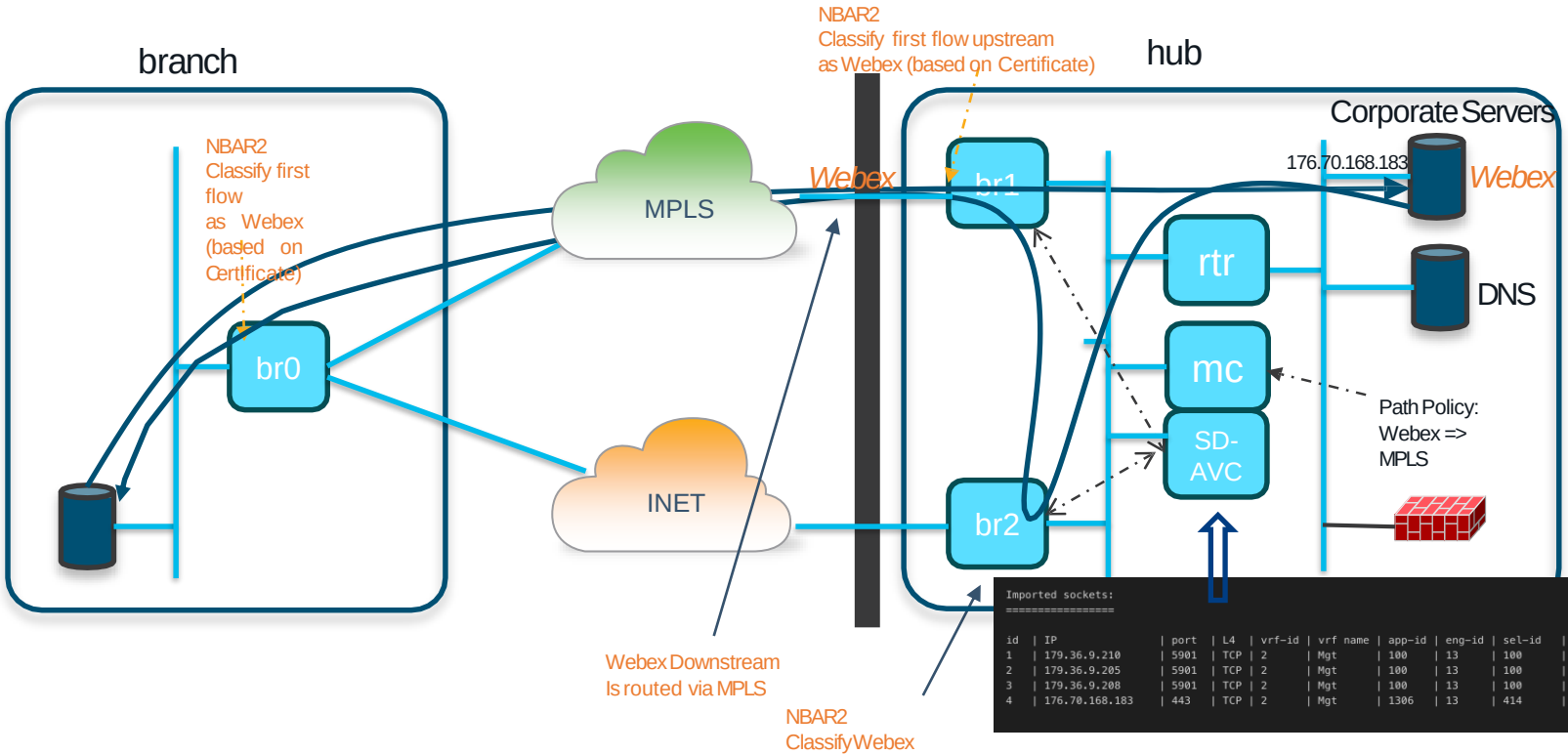
- Sensors are network devices (with NBAR2) that produce classification information and export it to the SD-AVC network service.
- Consumers are network devices that consume classification information from the SD-AVC network service

A network device can be a sensor, a consumer or both.

Cloud Intelligence or SD-AVC as a Service



Asymmetric Fixed Webex example - with SD-AVC



How does SD-AVC work?

- Sensors with NBAR2, classify traffic & cache results in the form of **Application Rules**.
- **Application Rule** is defined as a L3 or Socket cache mapping to App-ID

Application Rule Example:

```
id      | IP              | port   | L4   | vrf-id | vrf name | app-id | eng-id | sel-id | app-name   | #hits    | black | weight| rating
=========================================================================================================================
0       | 64.103.117.145 | 5902   | TCP | 0      | global    | 100    | 13     | 100    | vnc        | 1        | no    | 69    | 1
```

both cache types are exported to SDWAN manager SD-AVC container for behavioural learning and circulated across all devices in overlay

sh avc sd-service info import[export] dump[l3-cache|socket-cache]

(config-t)# service internal

sh ip nbar classification cache sync import[export] last[format table]

Socket-cache Exported Application Rules

Prerequisites:

- ✓ Payload Engine Family or Advance Engine Family reaching final classification
- ✓ Cache entry most by ”valid” and not being blacklisted or in learning phase.

```
CSRSD-WAN-1#show ip nbar classification socket-cache 100
|server ip                                |vrf       |port |proto|application           |is   |is   |is   |max time |entry|hit   |
|                                         |          |     |     |name                  |valid|black|learn|to expiry|type |count |
|                                         |          |     |     |                      |     |list |phase|(sec)    |     |      |
|-----------------------------------------|----------|-----|-----|----------------------|-----|-----|-----|---------|-----|------|
|224.0.0.251                              |         1| 5353|UDP  |mdns                  |No   |No   |Yes  |50       |Infra|2     |
|192.168.1.2                              |         1| 5201|TCP  |Luis2                 |Yes  |No   |No   |1195     |Infra|245   |
```

```
CSRSD-WAN-1#show avc sd-service info export l3-cache
{
    "ip": "2620:0:861:ED1A::1",
    "vrf": "10",
    "appName": "wikipedia",
    "hits": 0,
    "rating": 0,
    "weight": 60,
    "timeToTtlExpire": 0,
    "type": "dns",
    "fqdn": "www.wikipedia.org",
    "fqdnHits": 0,
    "fqdnCommonLabels": 3,
    "fqdnCommonLabelsHits": 0
}
```

Example of L3 cache exported entry

```
CSRSD-WAN-1#show avc sd-service info export socket-cache
"socketCache": {
"elements": [
{
    "ip": "192.168.150.14",
    "port": 22,
    "vrf": "Mgmt-intf",
    "l4Proto": "TCP",
    "appName": "ssh",
    "hits": 3,
    "black": false,
    "weight": 56,
    "rating": 0
}]
```

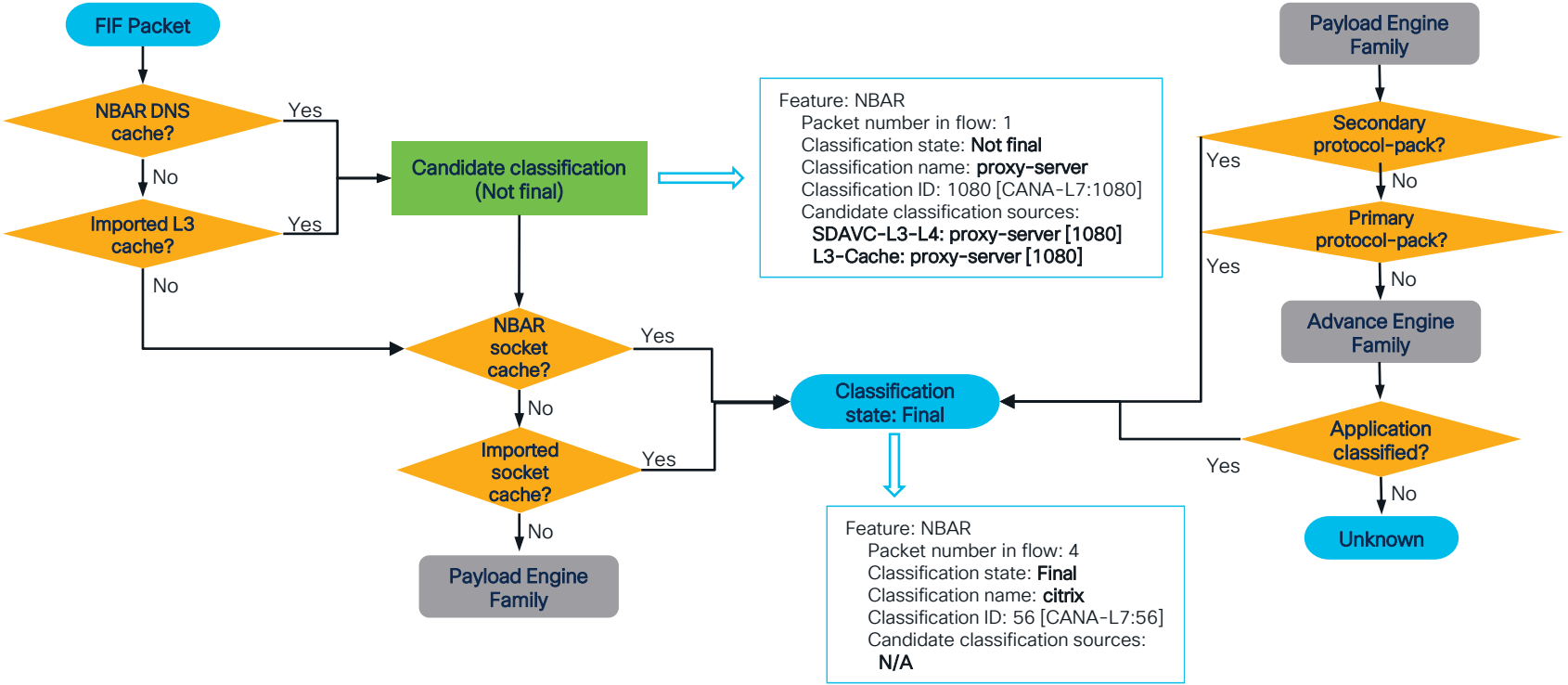
Example of socket-cache exported entry

If “rating” = 0
Cache entry is
not exported

L3 and L4 Imported Application Rules

```
CSRSD-WAN-200#show avc sd-service info import dump
Imported sockets
================

         |                                            |             |     |         |          |         |         |          |                         |       |                     |  keep after |
 id      | IP                                         | port        | L4  | vrf-id  | vrf name | app-id  | eng-id  | sel-id   | app-name                | black |  category:value     |  rollback   |
==========================================================================================================================================================================================================
=============
 0       | 157.240.229.35/32                          | 443         | TCP | 1       | 10       | 1454    | 13      | 518      | facebook                | no    | N/A                 | no          |
 1       | 172.253.115.95/32                          | 443         | TCP | 1       | 10       | 1456    | 13      | 520      | google-services         | no    | N/A                 | no          |
 2       | 208.80.154.224/32                          | 443         | TCP | 1       | 10       | 1547    | 13      | 608      | wikipedia               | no    | N/A                 | no          |
 3       | 192.229.211.108/32                         | 80          | TCP | 1       | 10       | 1750    | 13      | 819      | ocsp                    | no    | N/A                 | no          |
 4       | 192.168.150.14/32                          | 22          | TCP | 4       | Mgmt-int | 40      | 3       | 22       | ssh                     | no    | N/A                 | no          |
 5       | 192.168.150.11/32                          | 22          | TCP | 4       | Mgmt-int | 40      | 3       | 22       | ssh                     | no    | N/A                 | no          |

Imported L3 elements
====================

         |                                            |        |          |        |        |          |                          |                    |  keep after |
 id      | IP                                         | vrf-id | vrf name | app-id | eng-id | sel-id   | app-name                 |  category:value    |  rollback   |
==========================================================================================================================================================================================================
       0 | 142.251.111.84/32                          | 1      | 10       | 1456   | 13     | 520      | google-services          | N/A                | no          |
       1 | 172.253.115.95/32                          | 1      | 10       | 1456   | 13     | 520      | google-services          | N/A                | no          |
       2 | 31.13.71.36/32                             | 1      | 10       | 1454   | 13     | 518      | facebook                 | N/A                | no          |
       3 | 142.251.167.94/32                          | 1      | 10       | 1456   | 13     | 520      | google-services          | N/A                | no          |

CSRSD-WAN-200#
```

Order of Precedence Flowchart



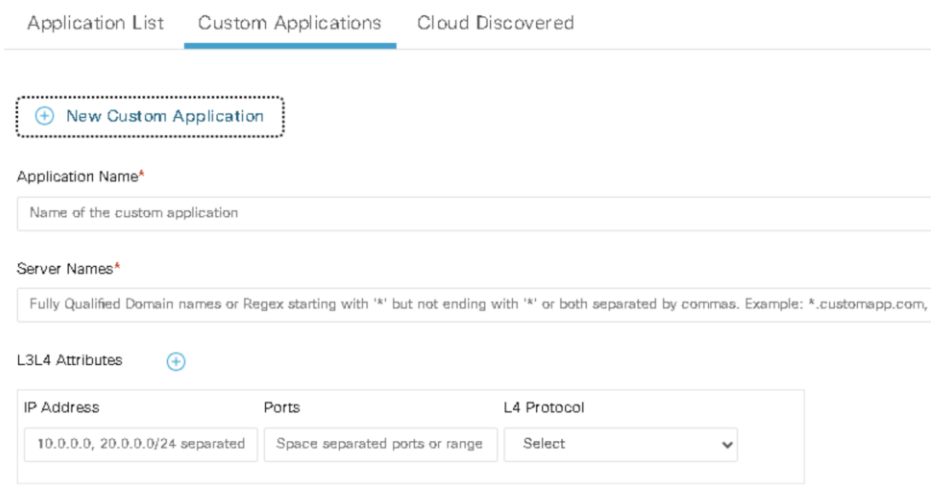
Custom Applications

Custom applications defined in SDWAN Manager are compiled by SD-AVC on the Secondary Protocol Pack.

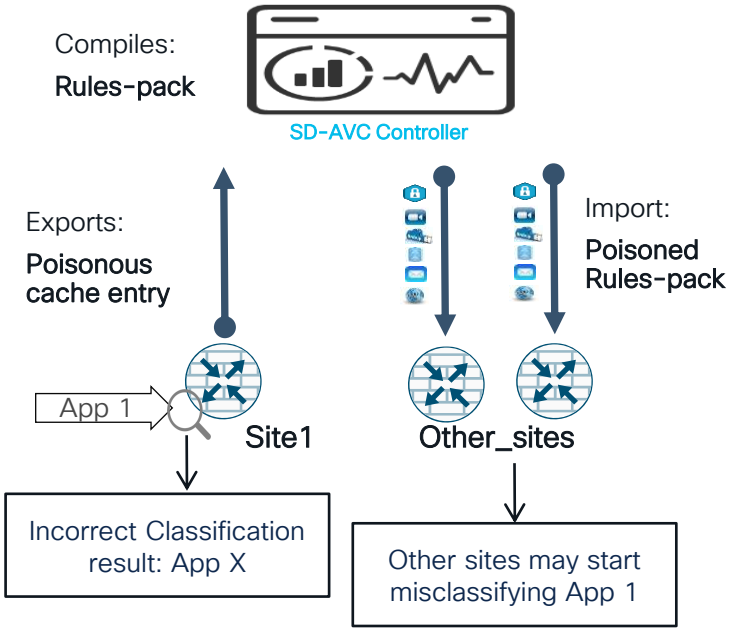
Match logic

- Between all L3/L4 attributes >> logical AND.
- Between L3/L4 and FQDN >> logical OR

~~FQDN defined Custom App and Server hosting multiple services.~~



What is the tricky part?



Cache poisoning can causa erratic classification across the SDWAN overlay!

Flows can be classified differently in different network nodes due to,

- Protocol-pack or Rules-Pack version mismatch
- DNS cache entries pointing to different application ID
- IOS version (older releases may not support custom applications)
- others

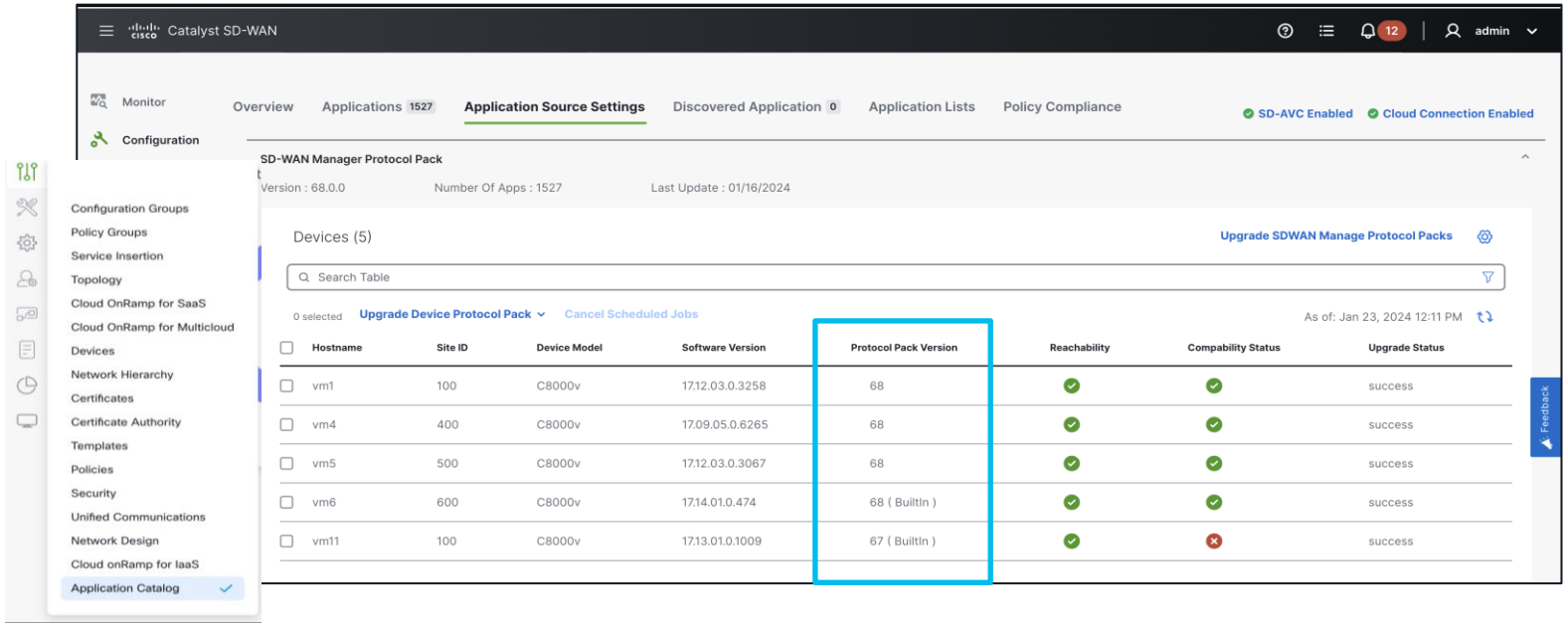
What to do if cache is poisoned?

1. Determine which device is poisoning the cache
   - Check local nbar cache on potential suspects (only local nbar cache entries are exported)
   - Check loaded Primary Protocol-pack version
   - Check SD-AVC connectivity and active Secondary protocol-pack and Rules-pack version
   - Maintain unified IOS versioning
   - Enable vertical debugging to track the source of the cache entry (requires TAC intervention)
2. Disconnect SD-AVC (no app-visibility config) or address the misclassification on culprit router
3. Wait for entry on SD-AVC cache to expired and disappear (may take up to 4 days) or flush SD-AVC cache (requires TAC intervention)

Enhancement: ability to clear cache of all devices centrally from vmanage.

Checking Primary Protocol-pack

From 20.14/17.14, Application Catalog is enhanced to simplify tasks related to NBAR Protocol Pack upgrade



Checking SDAVC connectivity

```
CSRSD-WAN-1#show avc sd-service info summary
Status: CONNECTED

Active controller:
   Type  : Primary
   IP    : 1.1.1.2
   Status: Connected
   Version        : 4.4.0
   Last connection: *12:34:48.000 PDT Tue Sep 24 2024

Active SDAVC import files:
    Protocol pack:           Not loaded
    Secondary protocol pack: PPDK_Viptela-POC-Tool-_abab33f10862c3ac6517296865fadb.pack
    Rules pack:              pp_update_Viptela-POC-Tool-19827_a_v2_df6de6458b5a.pack

CSRSD-WAN-1#
```

Check SDAVC Export/Import

```
CSRSD-WAN-1#show avc sd-service info export | sec summary
Export summary:
  Export requests:                          4
  Successful export requests:               4
  Failed export requests:                   0
  Total exported elements:                  468
  Last export time:                         *12:47:27.000 PDT Tue Sep 24 2024
  State sequence:                           0
CSRSD-WAN-1#
```

```
CSRSD-WAN-1#show avc sd-service info import | sec summary
Import summary:
  Number of import requests:                        9
  Number of import dp to cp events:                 10
  Number of import load successfully:               9
       Number of import successful sent to dataplane:  9
       Number of import successful load in dataplane:  9
  Number of import load failed:                     0
       Number of import load failed in CP:          0
       Number of import load failed in DP:          0
  Total imported elements:                  466
  Last import attempt time:                 *12:35:48.000 PDT Tue Sep 24 2024
  Last import load success time:            *12:35:48.000 PDT Tue Sep 24 2024
  Last import attempt file_name:            pp_update_Viptela-POC-Tool-19827_a_v2_1eddcae741bf.pack
  Active import file_name:                  pp_update_Viptela-POC-Tool-19827_a_v2_1eddcae741bf.pack
```

Misclassification due to SD-AVC imported cache could also be caused by outdated Rules-pack

Useful SD-AVC API calls

Initially we need generate an authentication token by login using SD-AVC credentials (TAC intervention required)

vmanage_20_9_4:~# curl --insecure -X POST -d **'username=sdavc'** -d **'password=password'** https://**localhost**:10502/avc-sd-service/external-api/login

vManage_20_12:~# curl -k -X POST -H "Content-Type: application/json" https://localhost:10502/avc-sd-service/gw/api/login -u sdavc:password

Response will be in the format **{"token":"Bearer <token>”}**. Then we can use the response token to issue API call below to clear SDAVC controller cache by disabling and reenabling behavioural learning.

curl -k -X POST -H "Content-Type: application/json" -H "Authorization:Bearer <token>" "https://localhost:10502/avc-sd-service/external-api/devices-config?segment=ALL-SEGMENT" -d '{"segment": "ALL-SEGMENT" , "devices": [{"deviceName" :null, "deivceIp" :null, "isBehavioralLearningEnabled": **false**}]}’

curl -k -X POST -H "Content-Type: application/json" -H "Authorization:Bearer <token>" "https://localhost:10502/avc-sd-service/external-api/devices-config?segment=ALL-SEGMENT" -d '{"segment": "ALL-SEGMENT" , "devices": [{"deviceName" :null, "deivceIp" :null, "isBehavioralLearningEnabled": **true**}]}’

Useful SD-AVC API calls (part 2)

To enable vertical debugging, we can use API below specifying the filter that will be use for the debugging. The Date will be in “epoch” format

```
vmanage:~# curl -k -X POST https://localhost:10502/avc-sd-service/api/verticalDebug \
                -d '{"name": "findzoom", "appname": "zoom-meetings", "date": 1724896345446, "enabled": true, "ip":"", "port":"",
"mac":"", "topics": ["socketcache", "13_rules", "asymmetric_sockets", "rules_in_db"], "segment": "", "vrf":"", "total" :0}' \
                -H "Content-Type: application/json" \
                -H "Authorization: Bearer <token>" {"success":true,"message":"Vertical debug was added successfully"}
```

To find and the delete the rule number that was created

```
vmanage:~# curl -k -X GET https://localhost:10502/avc-sd-service/api/verticalDebug/status?action=detail -H "Content-Type:
application/json" -H "Authorization: Bearer <token>"
```

```
vmanage:~#curl -v -k -X DELETE https://localhost:10502/avc-sd-service/api/verticalDebug/7 -H "Content-Type: application/json" -H
"Authorization: Bearer <token>"
```

The debugs can be found in vmanage admin-tech under directory **“/var/log/nms/containers/sdavc/vertical_debug.log”**

SDWAN Manager SD-AVC Logs

The logs contained in the directories below are mostly useful to understand everyday actions of SD-AVC controller. If an admin-tech is created, the logs in this directory will appear in it.

“/var/log/sdavc-proxy/”
“/var/log/nms/containers/sdavc”

Files **“sdavc_application.log”** and **“vertical_debug.log”** can be leveraged to see SD-AVC controller communication with dispatcher, http/udp interaction to trigger cache import/export or the applications added to the manager behavioral/Dynamic learning cache table.

```
vManage-1:/var/log/nms/containers/sdavc/avc/sdavc_application.log

2024-08-03 04:49:14 INFO  PPackRuleConnector:45 - Recycled on segment:stile-keep-alive started, total rules in DB:11550
2024-08-03 04:49:14 INFO  PPackRuleConnector:74 - Recycled of segment:stile-keep-alive,l3 :5760,sockets:5760,removed:30,skip_on_reduction:0
2024-08-03 04:49:14 INFO  PipeLineWorkerCloudResolver:41 - skip cloud resolve
...
2024-08-03 04:49:14 INFO  PPackRuleConnector:47 - creating rule pack file. segment:stile-keep-alive, json size:1471528, version:VERSION_2
2024-08-03 04:49:14 INFO  PPackRuleConnector:101 - Create rule-pack from json /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack for segment
stile-keep-alive
...
2024-08-03 04:49:14 INFO  PPackRuleConnector:132 - Going to copy /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack to /scratch/
pack_validation_dir/tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack/rule_pack_for_validation.pack
2024-08-03 04:49:15 INFO  PPackRuleConnector:195 - The validation of rule PASSED {
"cprCacheSyncTotalExmemUsed": 192992,
"cppCacheSyncDbStateReady": true,
"cppCacheSyncDbElementsNum": 2
}
...
2024-08-03 04:49:15 INFO  PPackRuleConnector:88 - Rule-pack: /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack validation success
2024-08-03 04:49:15 INFO  RulePackPublisher:421 - Going to publish 8fbd3dcad9f4 to segment stile-keep-alive
```

```
vManage-1:/var/log/nms/containers/sdavc/avc/vertical_debug.log.log

2024-09-26 13:39:07 INFO  VerticalDebugLogger:23 - {"version":1,"segment":"Cisco Sy1 - 19968","deviceId":"BR1-CEDGE1","deviceVersion":"1.0","stateS
equence":1,"msgTypes":["socketCache"],"socketCache":{"elements":[{"ip":"185.15.59.240","port":443,"vrf":"40","l4Proto":"TCP","appName":"wikipedia",
"hits":6,"black":false,"weight":67,"rating":6}]}}
2024-09-26 13:39:09 INFO  VerticalDebugLogger:23 - {"version":1,"segment":"Cisco Sy1 - 19968","deviceId":"BR1-CEDGE1","deviceVersion":"1.0","stateS
equence":1,"msgTypes":["l3Cache"],"l3Cache":{"elements":[{"ip":"185.15.59.224","vrf":"40","appName":"wikipedia","hits":9,"rating":5,"weight":60,"ti
meToTtlExpire":153,"type":"dns","fqdn":"intake-analytics.wikimedia.org","fqdnHits":1,"fqdnCommonLabels":1,"fqdnCommonLabelsHits":5}]}}
2024-09-26 13:39:11 INFO  VerticalDebugLogger:23 - {"version":1,"segment":"Cisco Sy1 - 19968","deviceId":"BR1-CEDGE1","deviceVersion":"1.0","stateS
equence":1,"msgTypes":["l3Cache"],"l3Cache":{"elements":[{"ip":"185.15.59.240","vrf":"40","appName":"wikipedia","hits":17,"rating":11,"weight":60,"
timeToTtlExpire":157,"type":"dns","fqdn":"upload.wikimedia.org","fqdnHits":11,"fqdnCommonLabels":3,"fqdnCommonLabelsHits":11}]}}
```

Note: for vertical debug logs to be printed vertical debugging must be enabled via API calls.

SDWAN Manager SD-AVC Logs

```
vManage-1:/var/log/nms/containers/sdavc/avc/sdavc_application.log

2024-08-03 04:49:14 INFO  PPackRuleConnector:45 - Recycled on segment:stile-keep-alive started, total rules in DB:11550
2024-08-03 04:49:14 INFO  PPackRuleConnector:74 - Recycled of segment:stile-keep-alive,l3 :5760,sockets:5760,removed:30,skip_on_reduction:0
2024-08-03 04:49:14 INFO  PipeLineWorkerCloudResolver:41 - skip cloud resolve
...
2024-08-03 04:49:14 INFO  PPackRuleConnector:47 - creating rule pack file. segment:stile-keep-alive, json size:1471528, version:VERSION_2
2024-08-03 04:49:14 INFO  PPackRuleConnector:101 - Create rule-pack from json /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack for segment
stile-keep-alive
...
2024-08-03 04:49:14 INFO  PPackRuleConnector:132 - Going to copy /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack to /scratch/
pack_validation_dir/tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack/rule_pack_for_validation.pack
2024-08-03 04:49:15 INFO  PPackRuleConnector:195 - The validation of rule PASSED {
"cprCacheSyncTotalExmemUsed": 192992,
"cppCacheSyncDbStateReady": true,
"cppCacheSyncDbElementsNum": 2
}
...
2024-08-03 04:49:15 INFO  PPackRuleConnector:88 - Rule-pack: /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack validation success
2024-08-03 04:49:15 INFO  RulePackPublisher:421 - Going to publish 8fbd3dcad9f4 to segment stile-keep-alive
```

SDWAN Manager SD-AVC Logs

```
vManage-1:/var/log/nms/containers/sdavc/avc/vertical_debug.log.log

2024-09-26 13:39:07 INFO  VerticalDebugLogger:23 - {"version":1,"segment":"Cisco Sy1 - 19968","deviceId":"BR1-CEDGE1","deviceVersion":"1.0","stateS
equence":1,"msgTypes":["socketCache"],"socketCache":{"elements":[{"ip":"185.15.59.240","port":443,"vrf":"40","l4Proto":"TCP","appName":"wikipedia",
"hits":6,"black":false,"weight":67,"rating":6}]}}
2024-09-26 13:39:09 INFO  VerticalDebugLogger:23 - {"version":1,"segment":"Cisco Sy1 - 19968","deviceId":"BR1-CEDGE1","deviceVersion":"1.0","stateS
equence":1,"msgTypes":["l3Cache"],"l3Cache":{"elements":[{"ip":"185.15.59.224","vrf":"40","appName":"wikipedia","hits":9,"rating":5,"weight":60,"ti
meToTtlExpire":153,"type":"dns","fqdn":"intake-analytics.wikimedia.org","fqdnHits":1,"fqdnCommonLabels":1,"fqdnCommonLabelsHits":5}]}}
2024-09-26 13:39:11 INFO  VerticalDebugLogger:23 - {"version":1,"segment":"Cisco Sy1 - 19968","deviceId":"BR1-CEDGE1","deviceVersion":"1.0","stateS
equence":1,"msgTypes":["l3Cache"],"l3Cache":{"elements":[{"ip":"185.15.59.240","vrf":"40","appName":"wikipedia","hits":17,"rating":11,"weight":60,"
timeToTtlExpire":157,"type":"dns","fqdn":"upload.wikimedia.org","fqdnHits":11,"fqdnCommonLabels":3,"fqdnCommonLabelsHits":11}]}}
```

```
vManage-1:/var/log/nms/containers/sdavc/avc/sdavc_application.log

2024-08-03 04:49:14 INFO  PPackRuleConnector:45 - Recycled on segment:stile-keep-alive started, total rules in DB:11550
2024-08-03 04:49:14 INFO  PPackRuleConnector:74 - Recycled of segment:stile-keep-alive,l3 :5760,sockets:5760,removed:30,skip_on_reduction:0
2024-08-03 04:49:14 INFO  PipeLineWorkerCloudResolver:41 - skip cloud resolve
...
2024-08-03 04:49:14 INFO  PPackRuleConnector:47 - creating rule pack file. segment:stile-keep-alive, json size:1471528, version:VERSION_2
2024-08-03 04:49:14 INFO  PPackRuleConnector:101 - Create rule-pack from json /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack for segment
stile-keep-alive
...
2024-08-03 04:49:14 INFO  PPackRuleConnector:132 - Going to copy /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack to /scratch/
pack_validation_dir/tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack/rule_pack_for_validation.pack
2024-08-03 04:49:15 INFO  PPackRuleConnector:195 - The validation of rule PASSED {
"cprCacheSyncTotalExmemUsed": 192992,
"cppCacheSyncDbStateReady": true,
"cppCacheSyncDbElementsNum": 2
}
...
2024-08-03 04:49:15 INFO  PPackRuleConnector:88 - Rule-pack: /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack validation success
2024-08-03 04:49:15 INFO  RulePackPublisher:421 - Going to publish 8fbd3dcad9f4 to segment stile-keep-alive
```

SDWAN Manager SD-AVC Logs

The logs contained in the directories below are mostly useful to understand everyday actions of SD-AVC controller. If an admin-tech is created, the logs in this directory will appear in it.

“/var/log/sdavc-proxy/”
“/var/log/nms/containers/sdavc”

Files **“sdavc_application.log”** and **“vertical_debug.log”** can be leveraged to see SD-AVC controller communication with dispatcher, http/udp interaction to trigger cache import/export or the applications added to the manager behavioral/Dynamic learning cache table.

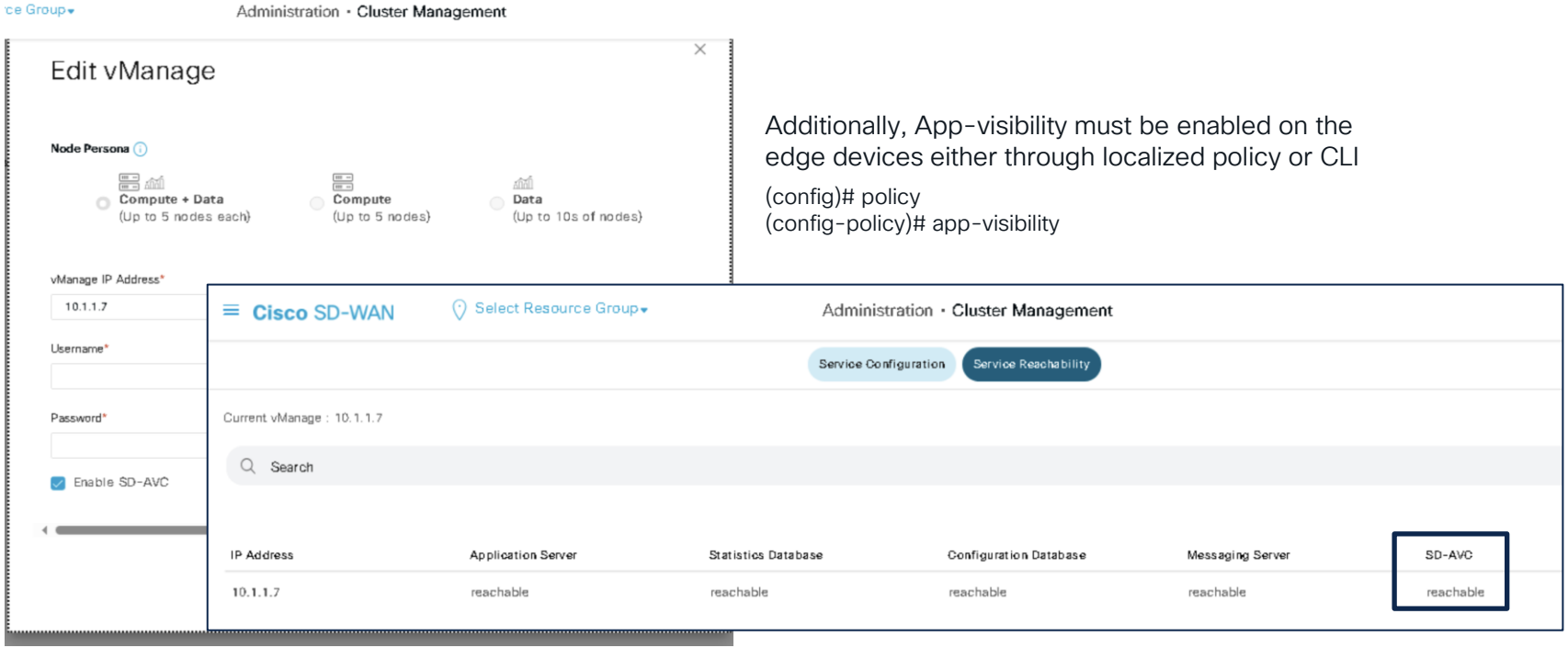
```
vManage-1:/var/log/nms/containers/sdavc/avc/sdavc_application.log

2024-08-03 04:49:14 INFO  PPackRuleConnector:45 - Recycled on segment:stile-keep-alive started, total rules in DB:11550
2024-08-03 04:49:14 INFO  PPackRuleConnector:74 - Recycled of segment:stile-keep-alive,l3 :5760,sockets:5760,removed:30,skip_on_reduction:0
2024-08-03 04:49:14 INFO  PipeLineWorkerCloudResolver:41 - skip cloud resolve
...
2024-08-03 04:49:14 INFO  PPackRuleConnector:47 - creating rule pack file. segment:stile-keep-alive, json size:1471528, version:VERSION_2
2024-08-03 04:49:14 INFO  PPackRuleConnector:101 - Create rule-pack from json /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack for segment
stile-keep-alive
...
2024-08-03 04:49:14 INFO  PPackRuleConnector:132 - Going to copy /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack to /scratch/
pack_validation_dir/tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack/rule_pack_for_validation.pack
2024-08-03 04:49:15 INFO  PPackRuleConnector:195 - The validation of rule PASSED {
"cprCacheSyncTotalExmemUsed": 192992,
"cppCacheSyncDbStateReady": true,
"cppCacheSyncDbElementsNum": 2
}
...
2024-08-03 04:49:15 INFO  PPackRuleConnector:88 - Rule-pack: /tmp/pp_update_stile-keep-alive_a_v2_8fbd3dcad9f4.pack validation success
2024-08-03 04:49:15 INFO  RulePackPublisher:421 - Going to publish 8fbd3dcad9f4 to segment stile-keep-alive
```

```
vManage-1:/var/log/nms/containers/sdavc/avc/vertical_debug.log.log

2024-09-26 13:39:07 INFO  VerticalDebugLogger:23 - {"version":1,"segment":"Cisco Sy1 - 19968","deviceId":"BR1-CEDGE1","deviceVersion":"1.0","stateS
equence":1,"msgTypes":["socketCache"],"socketCache":{"elements":[{"ip":"185.15.59.240","port":443,"vrf":"40","l4Proto":"TCP","appName":"wikipedia",
"hits":6,"black":false,"weight":67,"rating":6}]}}
2024-09-26 13:39:09 INFO  VerticalDebugLogger:23 - {"version":1,"segment":"Cisco Sy1 - 19968","deviceId":"BR1-CEDGE1","deviceVersion":"1.0","stateS
equence":1,"msgTypes":["l3Cache"],"l3Cache":{"elements":[{"ip":"185.15.59.224","vrf":"40","appName":"wikipedia","hits":9,"rating":5,"weight":60,"ti
meToTtlExpire":153,"type":"dns","fqdn":"intake-analytics.wikimedia.org","fqdnHits":1,"fqdnCommonLabels":1,"fqdnCommonLabelsHits":5}]}}
2024-09-26 13:39:11 INFO  VerticalDebugLogger:23 - {"version":1,"segment":"Cisco Sy1 - 19968","deviceId":"BR1-CEDGE1","deviceVersion":"1.0","stateS
equence":1,"msgTypes":["l3Cache"],"l3Cache":{"elements":[{"ip":"185.15.59.240","vrf":"40","appName":"wikipedia","hits":17,"rating":11,"weight":60,"
timeToTtlExpire":157,"type":"dns","fqdn":"upload.wikimedia.org","fqdnHits":11,"fqdnCommonLabels":3,"fqdnCommonLabelsHits":11}]}}
```

Note: for vertical debug logs to be printed vertical debugging must be enabled via API calls.

Cisco SDWAN Manager role

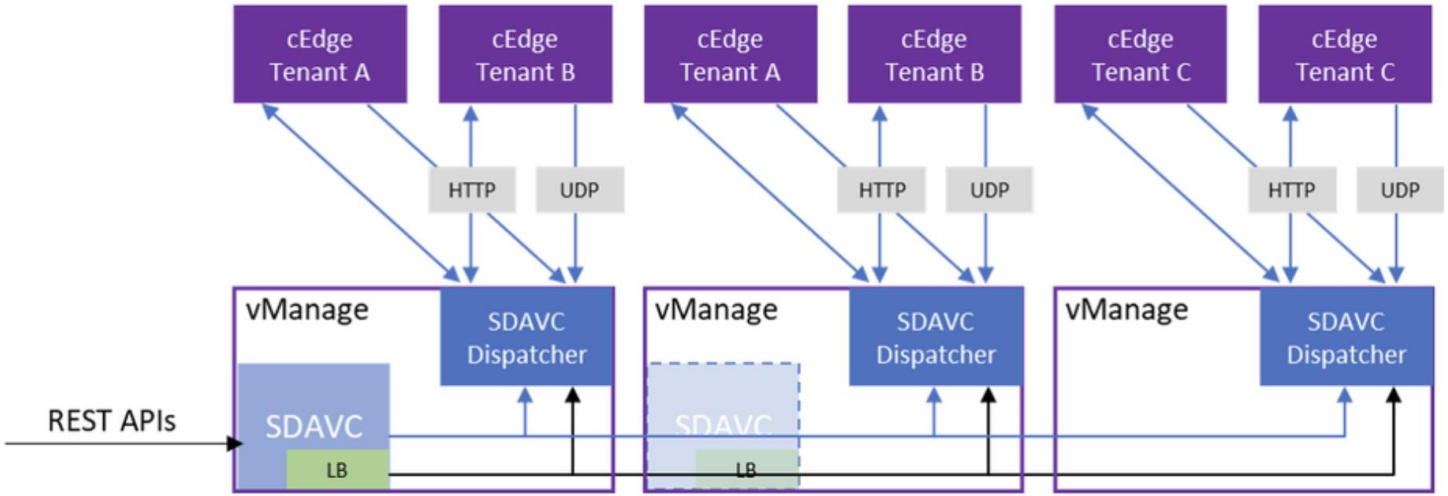
Enabling SDAVC in SDWAN Manager



Additionally, App-visibility must be enabled on the edge devices either through localized policy or CLI

(config)# policy
(config-policy)# app-visibility

SDAVC SDWAN Manager Architecture (20.9 and before)



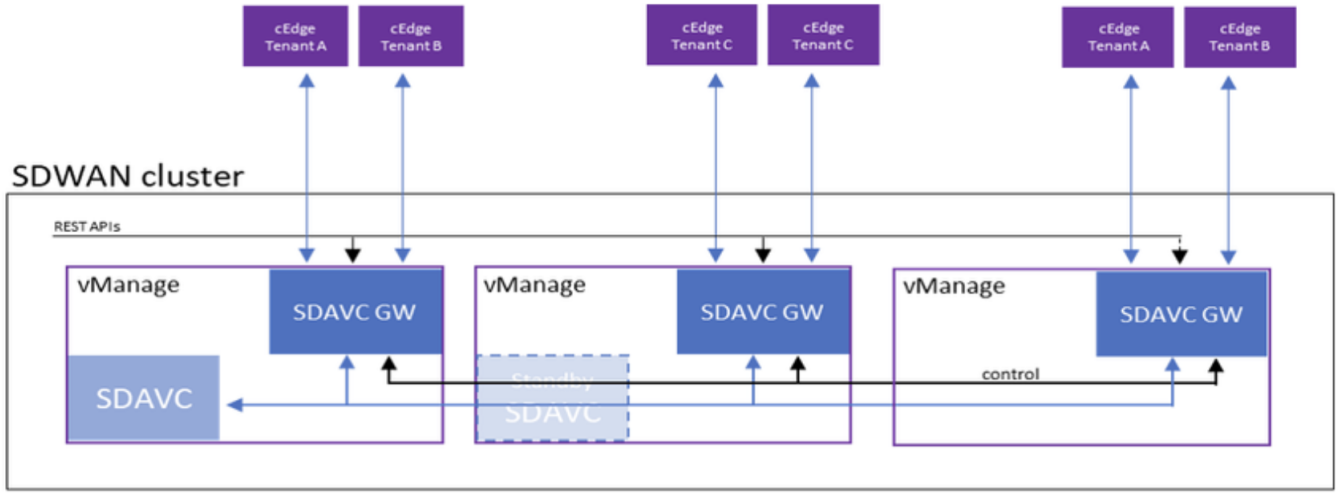
- SD-AVC service running on one or multiple SDWAN Manager nodes
- Dispatcher is the proxy service running on every Manager node

Verifying SDAVC status (20.9 and before)

```
vManage-1# request nms all status
...
NMS SDAVC server
        Enabled: true
        Status: running PID:20412 for 16657s
NMS SDAVC proxy
        Enabled: true
        Status:  running PID:12228 for 4883021s
vManage-1#
```

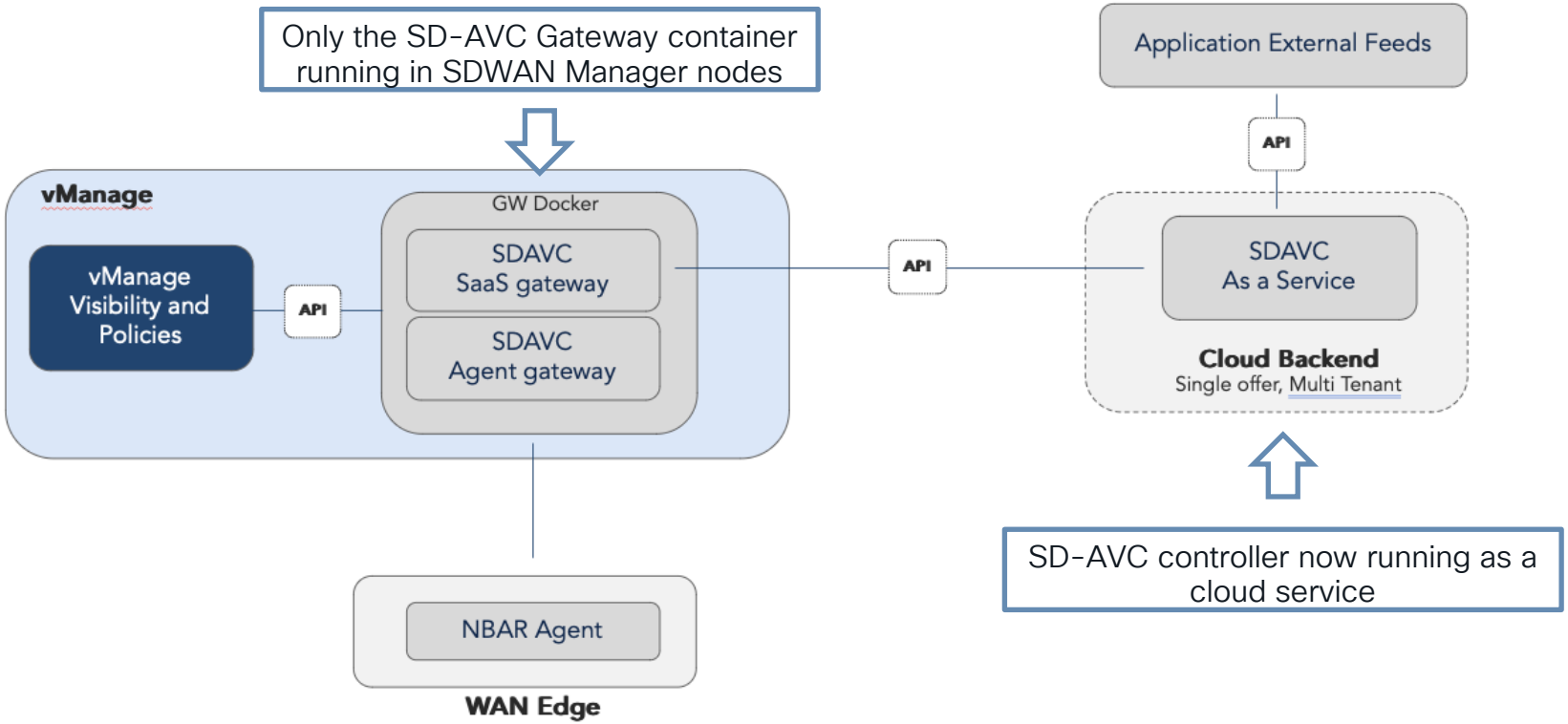
```
vManage-1# request nms sdavc diagnostics
NMS SDAVC server
Checking cluster connectivity...
Pinging server  on 10.1.1.7:10502...
Starting Nping 0.7.80 ( https://nmap.org/nping ) at 2024-09-24 15:20 PDT
SENT (0.0020s) Starting TCP Handshake > 10.1.1.7:10502
RCVD (0.0020s) Handshake with 10.1.1.7:10502 completed
SENT (1.0031s) Starting TCP Handshake > 10.1.1.7:10502
RCVD (1.0031s) Handshake with 10.1.1.7:10502 completed
SENT (2.0041s) Starting TCP Handshake > 10.1.1.7:10502
RCVD (2.0041s) Handshake with 10.1.1.7:10502 completed

Max rtt: 0.147ms | Min rtt: 0.014ms | Avg rtt: 0.071ms
TCP connection attempts: 3 | Successful connections: 3 | Failed: 0 (0.00%)
Nping done: 1 IP address pinged in 2.00 seconds

Checking server 10.1.1.7...
Server network connections
--------------------------
tcp        0      0 127.0.0.1:10502         0.0.0.0:*               LISTEN      20340/docker-proxy
tcp        0      0 10.1.1.7:34617          10.1.1.7:10502          TIME_WAIT   -
tcp        0      0 10.1.1.7:44379          10.1.1.7:10502          TIME_WAIT   -
tcp        0      0 10.1.1.7:35259          10.1.1.7:10502          TIME_WAIT   -
tcp6       0      0 10.1.1.7:57653          10.1.1.7:10502          TIME_WAIT   -
vManage-1#
```

SDAVC SDWAN Manager Architecture (starting in 20.10)



- SD-AVC service running on a single SDWAN manager
- Dispatcher is replaced with SD-AVC Gateway running as a container in every Manager node
- Gateway communicates with the local SD-AVC service instace

20.10/17.10 SD-AVC SaaS



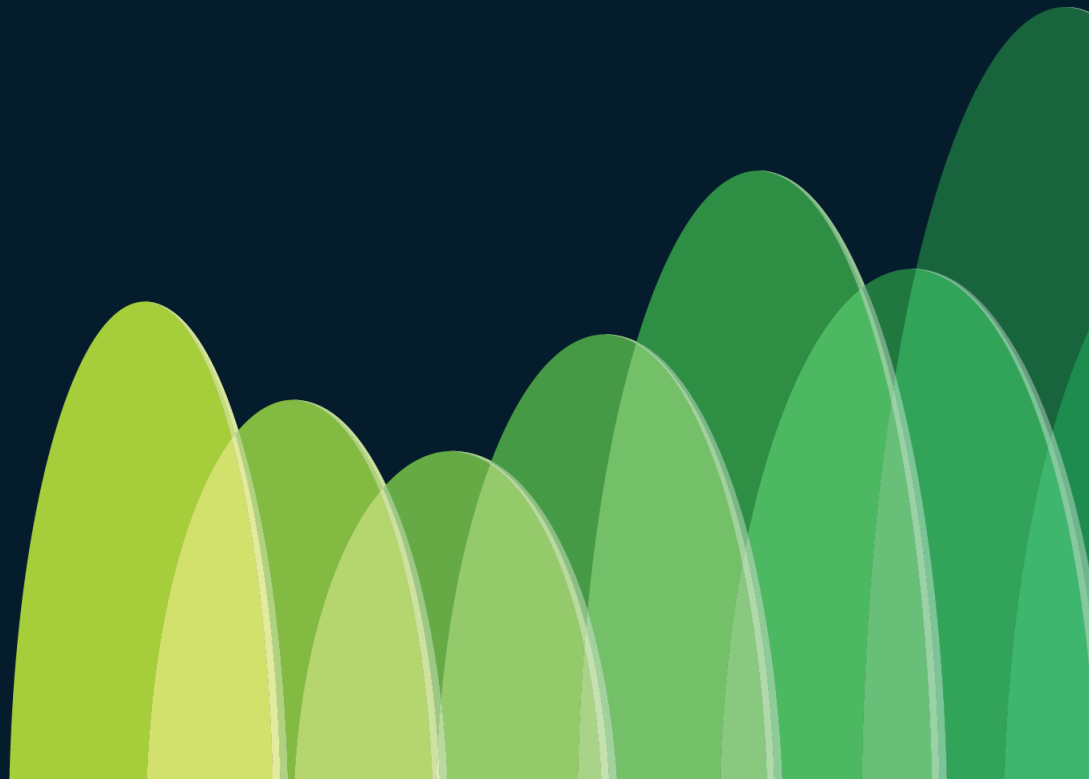
Verifying SDAVC status (starting in 20.10)

```
vManage01# request nms all status
NMS SDAVC server
        Enabled: true
        Status: running PID:6669 for 1385s
NMS SDAVC gateway
        Enabled: true
        Status: running PID:3499 for 1506s
```


```
vManage01# request nms sdavc diagnostics
NMS SDAVC server
Checking cluster connectivity...
Pinging server  on 172.16.0.1:10504...
Starting Nping 0.7.80 ( https://nmap.org/nping ) at 2024-09-24 23:29 UTC
SENT (0.0017s) Starting TCP Handshake > 172.16.0.1:10504
RCVD (0.0018s) Handshake with 172.16.0.1:10504 completed
SENT (1.0031s) Starting TCP Handshake > 172.16.0.1:10504
RCVD (1.0031s) Handshake with 172.16.0.1:10504 completed
SENT (2.0043s) Starting TCP Handshake > 172.16.0.1:10504
RCVD (2.0043s) Handshake with 172.16.0.1:10504 completed

Max rtt: 0.089ms | Min rtt: 0.016ms | Avg rtt: 0.043ms
TCP connection attempts: 3 | Successful connections: 3 | Failed: 0 (0.00%)
Nping done: 1 IP address pinged in 2.00 seconds
WARNING: Reverse DNS lookup on 172.16.0.1 timed out after 2 seconds
```

```
vManage01# request nms sdavc-gw diagnostics
NMS SDAVC gateway
Checking cluster connectivity...
Pinging server  on 172.16.0.1:10502...
Starting Nping 0.7.80 ( https://nmap.org/nping ) at 2024-09-24 23:29 UTC
SENT (0.0015s) Starting TCP Handshake > 172.16.0.1:10502
RCVD (0.0015s) Handshake with 172.16.0.1:10502 completed
SENT (1.0028s) Starting TCP Handshake > 172.16.0.1:10502
RCVD (1.0028s) Handshake with 172.16.0.1:10502 completed
SENT (2.0041s) Starting TCP Handshake > 172.16.0.1:10502
RCVD (2.0042s) Handshake with 172.16.0.1:10502 completed

Max rtt: 0.074ms | Min rtt: 0.015ms | Avg rtt: 0.054ms
TCP connection attempts: 3 | Successful connections: 3 | Failed: 0 (0.00%)
```

Key Takeaways



Key Takeaways

- Applications signatures are compiled into Protocol-packs and leveraged by NBAR2 for flows classification.
- Protocol Packs updates comes with each IOS-XE SD-WAN image, but latest updated versions can be loaded to NBAR2 any time.
- NBAR2 classification result can change as long as classification result is still “candidate” and “not final” classification has been accomplished.
- DNS enhances FIF classification, but NBAR2 agent must see the DNS request and response and the subsequent application flow to generate a DNS learning cache entry.
- Enabling SD-AVC is key to get better classification outcomes.
- When SD-AVC is enabled with cloud connectivity FPM (based in IP) is enhanced significantly due to the application vendor automatic feeds.
- Some features using application classification are FPM dependent.
- Watchout for cache poisoning
- Starting on 20.14 it is possible to clear SDAVC controller cache in SDWAN Manager through CLI (commands present in the Annex slides).

Troubleshooting Tips

- For Flow table outputs such as:

```
# show sdwan app-fwd cflowd flows format table
# show sdwan app-fwd dpi flows table
```

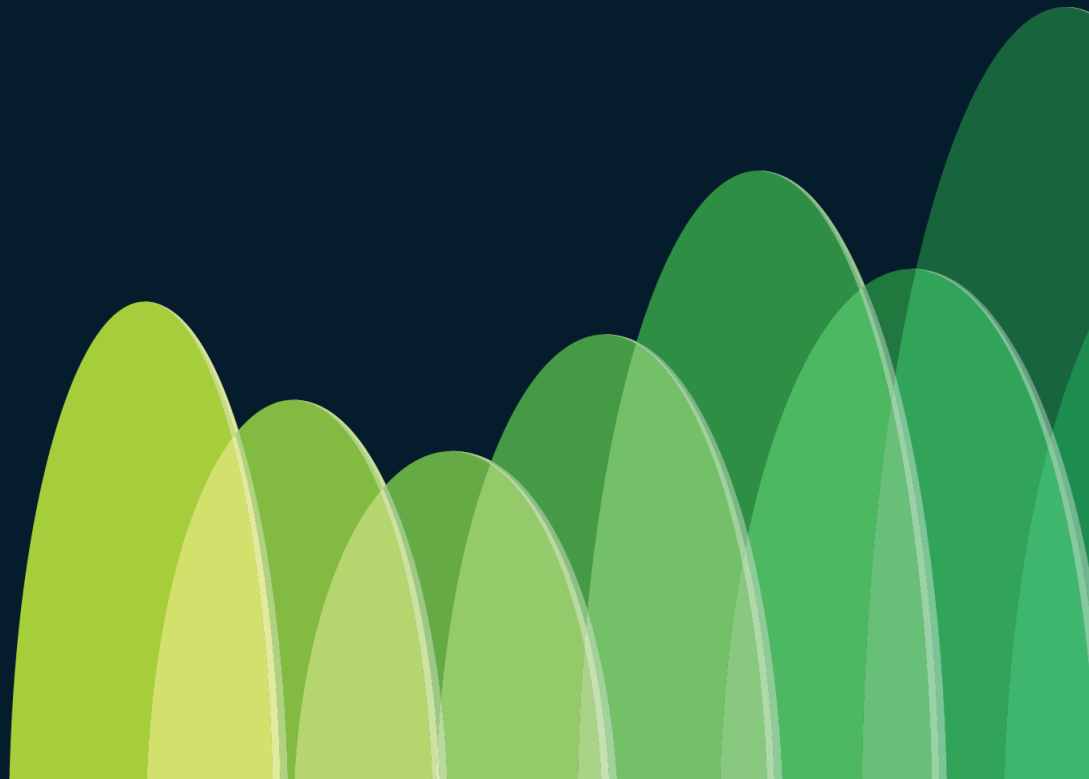
  application name field value can change over time based on Classification state Final/No-final.

- For FPM dependent features we need to focus on FIF packet classification result (Final classification is not a correct point of reference in these scenarios).
- To understand the flow behavior, we can inspect the datapath processing for FIF and subsequent packets using packet-trace feature (example below).

```
# debug platform condition ipv4 access-list <access-list-name> both
# debug platform packet-trace copy packet both l2
# debug platform packet-trace packet 1024 fia-trace
# debug platform condition (start|stop)
# show platform packet-trace (summary|statistics|packet #(all) decode)
```

- Always keep present the order of precedence between NBAR and SD-AVC application rules (cache) and protocol packs.

Annex



SDWAN Manager SDAVC Containers

```
vManage01# request nms container-manager diagnostics
NMS container manager
Checking container-manager status

Listing all images
--------------------------
REPOSITORY                    TAG               IMAGE ID       CREATED             SIZE
cloudagent-v2                 6d85a63e959       f8b22e12c03e   8 months ago        528MB
sdwan/service-proxy           1.27.2            1885de1b5c49   9 months ago        163MB
sdwan/cluster-oracle          1.0.1             7c116c2b6545   9 months ago        363MB
sdwan/data-collection-agent   1.0.1             6f472c45cbe4   11 months ago       210MB
sdwan/device-data-collector   1.0.0             2b15d5726af5   12 months ago       53.2MB
sdwan/ratelimit               master            719f624e9268   15 months ago       45.7MB
sdwan/configuration-db        4.4.15            11c2d134ee1e   15 months ago       714MB
sdavc                         4.4.0             acd4e8527e04   15 months ago       635MB
sdwan/host-agent              1.0.1             2b1f9b284760   16 months ago       243MB
sdwan/application-server      19.1.0            6e71762f7ace   16 months ago       853MB
sdwan/statistics-db           7.17.6            522bca66ca25   16 months ago       591MB
sdwan/messaging-server        0.20.0            14cf6bada947   17 months ago       106MB
sdwan/coordination-server     3.6.2             26d26d8374dd   17 months ago       251MB
sdwan/olap-db                 22.8.4.7          b7647c722e79   18 months ago       505MB
sdwan/upgrade-coordinator     1.0.0             ce2c9829562e   18 months ago       153MB
sdavc-gw                      4.4.0             4e3ebb844c67   19 months ago       208MB
sdwan/reporting               latest            a30afc5aed06   21 months ago       509MB
sdwan/support-tools           latest            0c3a995f455c   3 years ago         16.9MB

Listing all containers
--------------------------
CONTAINER ID   IMAGE                                COMMAND                  CREATED             STATUS                    PORTS
fab42c1a39a7   sdavc:4.4.0                          "/usr/local/bin/sdav…"   33 minutes ago      Up 33 minutes (healthy)   127.0.0.1:10503->8080/tcp, 127.0.0.1:10504->8443/tcp
3ec7fda95703   sdwan/statistics-db:7.17.6           "/usr/bin/docker-ini…"   34 minutes ago      Up 34 minutes (healthy)   127.0.0.1:9200->9200/tcp, 127.0.0.1:9300->9300/tcp
61c64ec1cf68   sdwan/olap-db:22.8.4.7               "/usr/bin/docker-ini…"   34 minutes ago      Up 34 minutes (healthy)   127.0.0.1:9010->9010/tcp, 8123/tcp, 127.0.0.1:9440->9440/tcp, 127.0.0.1:8143->8443/tc
c3fb06e48313   sdwan/reporting:latest               "python3 /reporting/…"   34 minutes ago      Up 34 minutes             127.0.0.1:58080->80/tcp
90004980eb0f   sdavc-gw:4.4.0                       "/bin/sh -c '/usr/lo…"   34 minutes ago      Up 34 minutes (healthy)   100.0.0.1:8444->8444/tcp, 127.0.0.1:8444->8444/tcp, 100.0.0.1:10501->8080/tcp, 127.0.
6fe45cf318da   sdwan/coordination-server:3.6.2      "/docker-entrypoint.…"   34 minutes ago      Up 34 minutes (healthy)   127.0.0.1:2181->2181/tcp, 127.0.0.1:2888->2888/tcp, 127.0.0.1:3888->3888/tcp
4663b011f0a6   sdwan/messaging-server:0.20.0        "/entrypoint.sh"         34 minutes ago      Up 34 minutes (healthy)   127.0.0.1:4222->4222/tcp, 127.0.0.1:6222->6222/tcp, 127.0.0.1:8222->8222/tcp
9effa393cc8a   cloudagent-v2:6d85a63e959            "python3 ./main.py"      34 minutes ago      Up 34 minutes             127.0.0.1:50051->50051/tcp
f7e05a65cbf7   sdwan/ratelimit:master               "/usr/local/bin/rate…"   34 minutes ago      Up 34 minutes (healthy)   6379/tcp, 127.0.0.1:8460-8462->8460-8462/tcp
960d13da4dc8   sdwan/data-collection-agent:1.0.1    "/usr/bin/docker-ini…"   34 minutes ago      Up 34 minutes (healthy)
4f61e215bad0   sdwan/host-agent:1.0.1               "/entrypoint.sh pyth…"   38 minutes ago      Up 37 minutes (healthy)   127.0.0.1:9099->9099/tcp, 172.16.0.1:9099->9099/tcp
be2c3c65eb36   sdwan/application-server:19.1.0      "/sbin/tini -g -- /e…"   40 minutes ago      Up 40 minutes (healthy)
484bb6286901   sdwan/service-proxy:1.27.2           "/entrypoint.sh"         41 minutes ago      Up 41 minutes (healthy)
940019da2d7e   sdwan/configuration-db:4.4.15        "/usr/bin/docker-ini…"   41 minutes ago      Up 41 minutes (healthy)   127.0.0.1:5000->5000/tcp, 127.0.0.1:6000->6000/tcp, 127.0.0.1:6362->6362/tcp, 127.0.0
498f0059614d   sdwan/device-data-collector:1.0.0    "/entrypoint.sh /vMD…"   5 weeks ago         Up 5 weeks (healthy)      127.0.0.1:8129->8129/tcp
9547214fd421   sdwan/cluster-oracle:1.0.1           "/entrypoint.sh java…"   5 weeks ago         Up 5 weeks (healthy)      127.0.0.1:9090->9090/tcp
```

Valid for version 20.12 for sdavc on-prem
For 20.12 with cloud connector enabled, only container sdavc-gw is enabled.
For 20.9 and below only container is SDAVC

SDWAN Manager SD-AVC Logs

The SDAVC container is first spun up by the **container-manager** service which is a wrapper service for docker process.

```
2021-07-19 17:10:11,403 [INFO] ::: Activating container(sdavc_container), enable(True)
2021-07-19 17:10:16,524 [INFO] ::: loading docker image /opt/data/extra-
packages/20.5.1/containers/sdavc_container/sdavc_docker_img_4.1.0_20201119_220858.tar
2021-07-19 17:10:16,569 [INFO] ::: Post status to
https://10.0.2.26:8443/dataservice/sdavc/task/sdavc_activate_10.0.2.26_ba2efe6f-fffd-49ae-8eea-b1d876e4309f, response(b'')
2021-07-19 17:10:38,279 [INFO] ::: Container Activation sdavc_container Successful
```

The first request post start of the SDAVC service is the loadbalance request which is initiated by SDAVC dispatcher service to SDAVC main service to check if there are any other members on the cluster running the SDAVC, if yes, how to load balance between the SDAVC members and which is the elected leader out of all members. The request is covered in the **vmanage-server.log**.

```
vManage:/var/log/nms$ cat vmanage-server.log | grep "SdavcManager\|loadToBalance"
```

SDWAN Manager SDAVC Logs (Cont.)

Once validated, the new rule pack will be stored in directory below.

```
vManage-1:/opt/data/containers/sdavc-proxy/ftp$ ls -al
total 16
drwxrwxrwx 2 root    root    4096 Sep 29 02:30 .
drwxr-xr-x 5 root    root    4096 Jul 23 09:40 ..
-rw-rw-r-- 1 vmanage vmanage  647 Sep 29 02:04 pp_update_Viptela-POC-Tool-19827_a_v2_07634d6523d2.pack
-rw-rw-r-- 1 vmanage vmanage  647 Sep 29 02:19 pp_update_Viptela-POC-Tool-19827_a_v2_90d52faaaee4.pack
vManage-1:/opt/data/containers/sdavc-proxy/ftp$
```

Log file recording SDAVC Agent interaction with Proxy/Dispatcher service in manager SDAVC controller, HTTP requests from cEdges, will be **/opt/data/containers/sdavc-proxy/log/sdavc-proxy.log**.

We can also confirm in which SDWAN Manager the SD-AVC is installed on, and which sensor/consumers are being load balanced to each instance on **/opt/data/containers/sdavc-proxy/config/load_balance.json**.

NBAR Agent

SDAVC configuration, metadata, Protocol Pack, Rules Pack among other necessary files are stored in the NBAR Agent' Bootflash.

```
CSRSD-WAN-1#more bootflash:sdavc/?
bootflash:sdavc/PPDK_Viptela-POC-Tool-_abab33f10862c3ac6517296865fadb.pack
bootflash:sdavc/container_application
bootflash:sdavc/import_file_meta.json
bootflash:sdavc/pp_update_Viptela-POC-Tool-19827_a_v2_b3c1e325328a.pack
bootflash:sdavc/pp_update_pp_minor_taxonomy_b72edc9e6ed2e42.json
bootflash:sdavc/sdavc_config.json
```


```
{
  "importVer3": {
    "protocolPackName": null,
    "secondaryProtocolPackName": "PPDK_Viptela-POC-Tool-_abab33f10862c3ac6517296865fadb.pack",
    "secondaryProtocolPackDpMemSize": 10646,
    "rulePackName": "pp_update_Viptela-POC-Tool-19827_a_v2_b3c1e325328a.pack",
    "rulePackDpMemSize": 830,
    "tcDescriptionFile": "pp_update_pp_minor_taxonomy_b72edc9e6ed2e42.json",
    "didRollbackOccur": false
  }
}
```


sdavc_config.json


```
{
  "version": 1,
  "sdavcVersion": "4.4.0",
  "import": {
    "isEnable": true,
    "intervalSec": 30,
    "isRollbackOnTimeout": false,
    "ppMonitorCycleIntervalSec": 60,
...
  },
  "export": {
    "isEnable": true,
    "socketCacheIntervalSec": 900,
    "l3CacheIntervalSec": 900,
    "protocolDiscoveryIntervalSec": 150,
...
    "socketCacheMaxEntriesToExport": 5000,
    "l3CacheMaxEntriesToExport": 5000,
    "l3CacheExportUnclassifiedDomains": false,
    "l3CacheExportUnclassifiedPrivateDomains": false,
...
    "socketCacheProtocolsToFilter": [
      "ssl",
      "http",
      "snmp",
      "ntp",
      "https",
```


import_file_meta.json

Verifying Custom Applications

If custom-app is defined, secondary protocol pack is installed in all the devices.

```
CSRSD-WAN-1#show ip nbar protocol-id | i PPDK_LOCAL
CUSTOM1                                                  3723          PPDK LOCAL
sdavc-connector-rest-CONTROL                             3879          PPDK LOCAL
sdavc-connector-test-CONTROL                             2941          PPDK LOCAL
sdavc-test                                               2931          PPDK LOCAL
```

When new protocol pack with custom application is received, a trigger is issued to resolve custom application references in policy and program application list again.

If a custom application is not used in a policy but it is present in the Protocol Pack, it is used for visibility only and not for traffic classification. Conversely, when a custom application is used in a policy, an activation message is sent to NBAR which enables the custom app for visibility and traffic classification.

```
CSRSD-WAN-1#show platform hardware qfp active feature nbar function sui_dump_graph_tunables_db p 1 4096 | inc CUSTOM1|ID|sdavc-test
ID, Name, Is active, Extraction, Aging, Priority, Visiblity only, Low id
2931, sdavc-test, TRUE, FALSE, 0, 11, No, 0
3723, CUSTOM1, TRUE, FALSE, 0, 11, Yes, 0
CSRSD-WAN-1#
```

Verifying Custom Applications

Verifying QFP programming of SDAVC imported cache entries

```
CSRSD-WAN-2#show platform hardware qfp active feature nbar function sui_dp_cache_sync_dump_db p 100000 | inc Luis|IP,
Type,IP,VRF,TCP/UDP,Port,Protocol id,Protocol name,Black listed,Has no sockets,Alternative priority,Visibility protocol id,Visibility protocol name,o365/category,o365/category priority,o365/service-area,o365/service-area priority, sdavc/endpoint-ip-location,sdavc/endpoint-ip-location priority,category id: value id
CIDR_IPV4_SOCKET,192.168.1.2/32,1,TCP,5201,3115,Luis2,FALSE,FALSE,n/a,0,n/a,n/a:n/a
CIDR_IPV4_SOCKET_IGNORE_VRF,192.168.1.2/32,n/a,TCP,5201,3115,Luis2,FALSE,FALSE,n/a,0,n/a,n/a:n/a

CSRSD-WAN-2#show platform hardware qfp active feature nbar function sui_dump_graph_tunables_db p 1 4096 | i Luis|ID,
ID, Name, Is active, Extraction, Aging, Priority, Visiblity only, Low id
3115, Luis2, TRUE, FALSE, 0, 11, No, 0
```

```
CSRSD-WAN-2#show avc sd-service info import dump
Imported sockets
================

     |                                                  |               |     |        |          |         |         |          |                              |       |                        |  keep after |
 id  | IP                                               | port          | L4  | vrf-id | vrf name | app-id  | eng-id  | sel-id   | app-name                     | black |  category:value        |  rollback   |
==================================================================================================================================================================================================================
======
 0   | 192.168.1.2/32                                   | 5201          | TCP | 65535  | N/A      | 3115    | 21      | 3115     | Luis2                        | no    | N/A                    | no          |
 ...
 8   | 192.168.1.2/32                                   | 5201          | TCP | 1      | 10       | 3115    | 21      | 3115     | Luis2                        | no    | N/A                    | no          |
```

CLIs summary

Edge devices

show ip nbar protocol-id | i <app> << To show if an application is present in HW
show platform hardware qfp active feature nbar function sui_dump_graph_tunables_db p 1 4096 << To verify if application is only for visibility
show ip nbar protocol-pack active || # show avc sd-service info summary << To show active protocol pack
test platform hardware qfp active infrastructure cft datapath function cft-debug-kill-all-flows << To clear all the flows on the device
show ip nbar classification dns learning cache <number of cache entries>
show ip nbar classification socket-cache <number of cache entries>
show ip nbar classification cache statistics
sh avc sd-service info import[export] dump[l3-cache|socket-cache]
(config-t)# service internal
show ip nbar classification cache sync export format table
show ip nbar classification cache sync import last
show platform software common-classification f0 object all << To verify application programming in software (f0)
show logging process sdavc_proxy internal start last 180 minutes >> For device's communication with dispatched/GW

CLIs summary

SDWAN Manager

```
# request nms-container sdavc[sdavc-gw] start/stop/status
# request nms sdavc[sdavc-gw] diagnostics
# request nms container-manager restart/start/status/stop
# request nms container-manager diagnostics
```

Serviceability – Clear Cache (Bad IP removal) was introduced as part of 20.14 version serviceability enhancements to address cache poisoning leading to incorrect classification of traffic.

To request SDAVC to clear cache for given IP or Subnet on all devices on SD-WAN Manager

```
# clear support sdavc cache <IP ADDRESS> [port <PORT>] [protocol <PROTOCOL>] [vrf <VRF ID>]
```

Show status of all Clear cache done on the network

```
# show support sdavc status cache [clear-index <clear-index>]
```

Show detailed status of all Clear cache done on the network

```
# show support sdavc status cache details  [clear-index <clear-index>]
```

Acronyms and Terminology

- SAIE - SD-WAN Application Intelligence Engine
- DPI – Deep Packet inspection
- NBAR – Network based application recognition
- NBAR Agent
- Protocol Pack
- Application Rules Pack
- FIF – First in Flow
- FPM – First Packet Match
- Flow stickiness
- SD-AVC – Software Defined Application Visibility and Control
- Behavioral Based Classification

Classification Terminology

Term	Description
Application Name	Application name in a human reading form
Application ID	Unique Application Identifier mapped to Application Name and used in exports
Flow	A session. Identified by 5 tuple (src IP, src Port, dst IP, dst port, vrf)
Socket	Identified by 3 tuple (dst IP, dst Port, vrf). Usually a server
L3-cache	Maps dst IP to a fqdn
FIF	First packet in the Flow
Bypass	No processing, just quick forwarding
DPI Flow Table	Table used to store flow information (incl. classification result)
Nbar cache	Used to classify subsequent flows

Fill Out Your Session Surveys

Participants who fill out a minimum of 4 session surveys and the overall event survey will get a unique Cisco Live t-shirt.

(from 11:30 on Thursday, while supplies last)

All surveys can be taken in the Cisco Events mobile app or by logging in to the Session Catalog and clicking the ‘Participant Dashboard’

Content Catalog

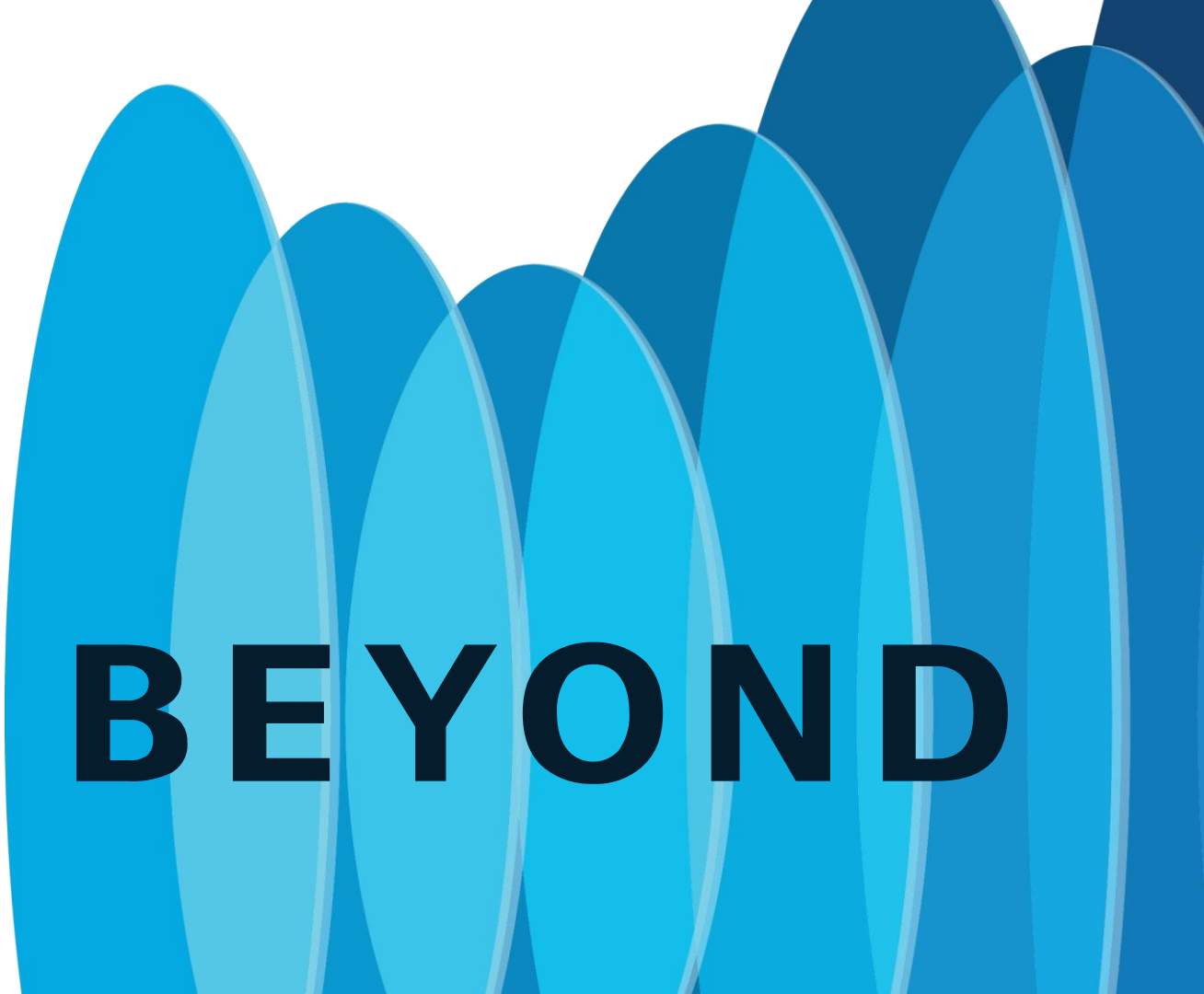
Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at ciscolive.com/on-demand. Sessions from this event will be available from March 3.

Contact me at: **luisdmar@cisco.com**


cisco
Thank you
cisco Live!

cisco Live!



GO BEYOND