

Gestionnaire de classification de l'EARL 8 : Un examen comportemental de LOUs, de L4Ops, et de Tableaux de Capmap

Contenu

[Introduction](#)

[Informations générales](#)

[Tableaux et registre des pertes d'unités logiques \(LOU\) de Capmap de programme](#)

[Étude de cas #1 - ACLs avec des indicateurs de TCP](#)

[Étude de cas #2 - utilisation de registre des pertes d'unités logiques \(LOU\) de 100%](#)

[Étude de cas #3 - QoS programmant avec L4Ops](#)

[Étude de cas #4 - Épuisement de la cause CAPMAP d'ACLs de Double-pile](#)

Introduction

Ce document décrit comment des unités de fonctionnement logique (LOUs) et les exécutions de la couche 4 (L4Ops) sont programmées dans la table de capmap. Il fournit des scénarios de panne, les genres d'erreurs que vous rencontrez typiquement dans ces situations, et ce que vous devriez impliquer de ces erreurs.

Le gestionnaire de classification (cm) gère la mémoire associative ternaire de classification (TCAM) et les ressources associées telles que des étiquettes, LOUs, des entrées de capmap, et d'autres. Des services cm sont utilisés par le gestionnaire de caractéristique (FM) et le gestionnaire de QoS (QM) pour programmer des entrées TCAM pour prendre en charge les caractéristiques de liste de contrôle d'accès (ACL) et de Qualité de service (QoS) de Cisco IOS®.

[Informations générales](#)

LOUs et L4Ops - LOUs signifie les unités de fonctionnement logique, qui sont des registres matériels qui sont utilisés pour enregistrer {opérateur, opérande} des tuples pour des numéros de port TCP/UDP spécifiés dans ACLs et listes de contrôle d'accès VLAN (VACLs). Ces tuples s'appellent également comme L4Ops. Par exemple, si vous appariez l'hôte X au gt 1023 de hôte Y, puis le tuple devient {gt, 1023}.

L4Ops - Exécutions de la couche 4.

Tables de Capmap - Les L4Ops décrits précédemment sont programmés dans les registre des pertes d'unités logiques (LOU) qui sont mis en référence par des entrées dans des tables de capmap. Chaque table de capmap a une limite de 10 (on est réservé pour la direction, qui réduit la limite à neuf) entrées (L4Ops). Des tables de Capmap sont indexées par l'étiquette TCAM elle-même.

Il y a de deux TCAMs, A et B ; chaque TCAM a les étiquettes 8K. Pour chaque TCAM, il y a une table de capmap des entrées 2K. Puisque chaque TCAM a les étiquettes 8K, il y a une superposition de 4:1 ici - carte de quatre étiquettes à une entrée de capmap. La superposition est


```

I M 32738 0x1FFF 0 0x00 0x000 255.255.255.255 - 0.0.0.0
- 0 0 0x0
I V 32739 4097 0 0 0 1.1.1.2 - 0.0.0.0
- 0 0 0 - ----- dscp=8; 0x00000000000000038
0
I M 32739 0x1FFF 0 0x00 0x000 255.255.255.255 - 0.0.0.0
- 0 0 0x0
I V 32740 4097 0 0 0 1.1.1.2 - 0.0.0.0
- 0 0 0 - ----- dscp=7; 0x00000000000000038
0
I M 32740 0x1FFF 0 0x00 0x000 255.255.255.255 - 0.0.0.0
- 0 0 0x0
I V 32741 4097 0 0 0 1.1.1.2 - 0.0.0.0
- 0 0 0 - ----- dscp=6; 0x00000000000000038
0
I M 32741 0x1FFF 0 0x00 0x000 255.255.255.255 - 0.0.0.0
- 0 0 0x0
I V 32745 4097 0 0 0 0.0.0.0 - 0.0.0.0
- 0 0 0 - ----- - 0x00000000400000038
0
I M 32745 0x1FFF 0 0x00 0x000 0.0.0.0 - 0.0.0.0
- 0 0 0x0

```

L'étiquette TCAM allouée pour l'interface vlan 1 est 2049 et l'étiquette TCAM allouée à l'interface vlan 2 est 4097. Ceci signifie que chacun des deux interfaces emploient la même table de capmap afin de mettre en référence les registre des pertes d'unités logiques (LOU) pour leur programmation L4Op.

Vous pouvez confirmer ceci avec cette commande (cinq as dans l'ACL a1 et quatre as dans l'ACL a2 vous implique devraient voir le capmap ajourner en tant que complètement) :

```

Sup2T#show platform hardware acl capmap tcam B label 4097
Hardware Capmap Table Entry For TCAM B. Free items are not shown

```

Index	Loc[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
1	212	10	9	8	7	6	5	4	3	2

```

Sup2T#show platform hardware acl capmap tcam B label 2049
Hardware Capmap Table Entry For TCAM B. Free items are not shown

```

Index	Loc[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
1	212	10	9	8	7	6	5	4	3	2

Tellement maintenant, à ce stade, si vous tentez d'installer un autre entrée de contrôle d'accès (ACE) L4Op-based, qui n'est pas expansible, pour l'un de ces interfaces, vous ne recevez aucune erreur disponible d'entrée libre de capmap.

```

Sup2T(config)#ip access-list extended a2
Sup2T(config-ext-nacl)#permit ip host 1.1.1.2 any dscp 10
Sup2T(config-ext-nacl)#end

```

```

*Sep 16 14:57:55.983: %EARL_CM-5-NOCAPMAP: No free capmap entry available
*Sep 16 14:57:55.991: %FMCORE-4-RACL_REDUCED: Interface Vlan2 routed traffic
will be software switched in ingress direction. L2 features may not be applied
at the interface

```

Ceci a comme conséquence le logiciel jetant un pont sur l'interface entière qui poserait potentiellement une commutation plus lente, une utilisation du CPU élevé, et d'autres problèmes

relatifs.

Note: L'ID de bogue Cisco [CSCuo02666](#) a été augmenté pour réparer ce problème. Le plus grand changement de la logique qu'il introduit est comment les étiquettes TCAM sont allouées. Maintenant Cisco alloue les étiquettes TCAM continuellement (2,3,4,5, et ainsi de suite) jusqu'à 2048 au lieu de à des lacunes de 2K. Ceci signifie que des tables de capmap ne sont plus partagées du début.

Souvenez-vous que LOUs, comme n'importe quelle autre ressource en matériel, sont limité. Il y a un total de 104 LOUs disponible pour l'usage :

```
Sup2T#show platform software acl lou
```

```
LOUs Registers (shadow copies)
```

Index	Type	A_Op	A_Val	A_Cnt	B_Op	B_Val	B_Cnt
0	PKT_QOS_GI	A is free.			NEQ	0	1
1	DST_PORT	LT	81	2	B is free.		
2	B & A are free						
3	B & A are free						
4	B & A are free						
5	B & A are free						
6	B & A are free						
7	B & A are free						
8	B & A are free						
9	B & A are free						
10	B & A are free						
11	B & A are free						
12	B & A are free						
13	B & A are free						
14	B & A are free						
15	B & A are free						

```
*snip*
```

```
95 B & A are free
96 B & A are free
97 B & A are free
98 B & A are free
99 B & A are free
100 B & A are free
101 B & A are free
102 B & A are free
103 B & A are free
```

Tableaux et registre des pertes d'unités logiques (LOU) de Capmap de programme

Des tables de Capmap sont utilisées seulement quand les exécutions L4 doivent être prises en considération. Notez qu'appariant sur des valeurs de Differentiated Services Code Point (DSCP)/Classe de service (Cos) est également considéré comme L4Op. Voici un exemple simple (ce utilise une version du code qui inclut la difficulté de l'ID de bogue Cisco [CSCuo02666](#)) que ce document établit en fonction incrémentalement :

```
Sup2T#show ip access-lists a3
```

```
Extended IP access list a3
 10 permit ip host 192.168.1.1 host 192.168.1.2
```

I have this applied to interface VLAN 1.

```
Sup2T#show run int vlan 1
Building configuration...
```

```
Current configuration : 84 bytes
!
interface Vlan1
 ip address 192.168.1.1 255.255.255.0
 ip access-group a3 in
end
```

Ceci est correctement programmé dans TCAM :

```
Sup2T#show platform hardware acl entry interface vlan 1 security in ip
mls_if_index:20000001 dir:0 feature:0 proto:0
```

pass#0 features

fno:0

tcam:B, bank:1, prot:0 Aces

```
Permit          ip host 192.168.1.1 host 192.168.1.2
L3_Deny         ip any any
```

```
Sup2t-MA1.7#show platform hardware acl entry interface vlan 1 security in ip detail
mls_if_index:20000001 dir:0 feature:0 proto:0
```

```
pass#0 features
UAPRSF: U-urg, A-ack, P-psh, R-rst, S-syn, F-fin
MLGFI: M-mpls_plus_ip_pkt, L-L4_hdr_vld, G-gpid_present, F-global_fmt_match, I-ife/ofe
's' means set; 'u' means unset; '-' means don't care
```

```
-----
-----
-----
-----
-----
```

I	INDEX	LABEL	FS	ACOS	AS	IP_SA	SRC_PORT
IP_DA		DST_PORT	F	FF	L4PROT	TCP-F:UAPRSF	MLGFI OtherL4OPs
RSLT				CNT			

```
-----
-----
-----
-----
```

fno:0

tcam:B, bank:1, prot:0 Aces

I	V	32741	2	0	0	0	192.168.1.1	-	192.168.1.2
-	0	0	0	-	-----	-			0x00000000000000038
0									
I	M	32741	0x1FFF	0	0x00	0x000	255.255.255.255	-	255.255.255.255
-	0	0	0x0						
I	V	32745	2	0	0	0	0.0.0.0	-	0.0.0.0

```

- 0 0 0 - ----- - 0x0000000040000038
0
I M 32745 0x1FFF 0 0x00 0x000 0.0.0.0 - 0.0.0.0
- 0 0 0x0

```

Des tables de Capmap sont mises en référence par l'intermédiaire de l'étiquette TCAM elle-même. Vous pouvez employer l'étiquette TCAM dans la commande de **<> d'étiquette de <> de tcam de capmap d'acl de logiciel de show platform [matériel]** afin de visualiser la table correspondante (matériel ou logiciel) pour cette étiquette TCAM.

```

Sup2T#show platform hardware acl capmap tcam B label 2
Hardware Capmap Table Entry For TCAM B. Free items are not shown

```

```

Index  Loc[9]  [8]  [7]  [6]  [5]  [4]  [3]  [2]  [1]  [0]
-----
1      212    0    0    0    0    0    0    0    0    0

```

Rien n'est alloué dans la table de capmap pour cette étiquette. L'ACL défini n'a aucun L4Ops ; il n'y a aucune condition requise d'installer une entrée dans la table de capmap.

Changez cet ACE à ceci :

```

Sup2T#show ip access-lists a3
Extended IP access list a3
 10 permit tcp host 192.168.1.1 host 192.168.1.2 eq www

```

Regardez la table de capmap de nouveau.

```

Sup2T#show platform software acl capmap tcam B label 2
Shadow Capmap Table Entry For TCAM B

```

```

-----
Output in a RST/INV/CNT format: RST - result value; INV - inverted;
                               CNT - aggregated reference account;

```

```

CBF - number of free cap bits (one per entry);
Free items are not shown

```

```

-----
Index  CBF      [9]          [8]          [7]          [6]
[5]    [4]          [3]          [2]          [1]
[0]

```

```

-----
1      9      Reserved      Free      Free      Free
Free    Free      Free      Free      Free
Free

```

Si vous égalisez directement à un numéro de port, il ne compte pas comme L4Op aussi bien.

Changez-le à ceci :

```

Sup2T#show ip access-lists a3
Extended IP access list a3
 10 permit tcp host 192.168.1.1 host 192.168.1.2 gt www

```

Examinez la table de capmap une fois de plus :

```
Sup2T#show platform software acl capmap tcam B label 2
Shadow Capmap Table Entry For TCAM B
```

```
-----
Output in a RST/INV/CNT format: RST - result value; INV - inverted;
                               CNT - aggregated reference account;
```

```
CBF - number of free cap bits (one per entry);
Free items are not shown
```

```
-----
Index   CBF      [9]          [8]          [7]          [6]
[5]     [4]     [3]          [2]          [1]
[0]
```

```
-----
-----
      2      8 212/0/1          Free          Free          Free
Free          Free          Free          Free          Free  3/1/1
```

Il y a maintenant une entrée dans la table de capmap. ACE a été traduit à un 3/1/1 dans la table de capmap. C'est du format RST/INV/CNT. Le RST ici spécifie dans lequel on a installé le registre des pertes d'unités logiques (LOU) ce L4Op, et le CNT décrit le compte agrégé pour ce LOU (plus d'informations sur ceci plus tard). Regardez cette sortie afin de comprendre comment la valeur RST est indexée :

```
Sup2T#show platform software acl capmap mapping
L4op_sel value      Reference
=====
0      -----      LOU0 B register
1      -----      LOU0 A register
2      -----      LOU1 B register
3      -----      LOU1 A register
.....
.....
206    -----      LOU103 B register
207    -----      LOU103 A register
208    -----      Global format match for global acl
209    -----      Group id present
210    -----      L4_hdr_vld
211    -----      Mpls_plus_ip_pkt
212    -----      ife/ofe for direction
(213-223)  ----      Reserved
(224-239)  ----      16 TCP flags map
(240-255)  ----      16 IPv6 ext header map
```

Vous pouvez voir que la valeur L4op_sel de 0 indique le registre d'index LOU0, la valeur de 1 indique le registre LOU0 A, la valeur de 2 points au registre d'index LOU1, la valeur de 3 points au registre d'index LOU1, et ainsi de suite. Le registre A est toujours programmé d'abord. **Le 3/1/1** sorti semble plus de raisonnable maintenant que vous voyez ceci.

Dans cette sortie, 3 signifie que le L4Op a été programmé dans le registre LOU1 A. Vous pouvez également vérifier où un L4Op est programmé si vous regardez dans le contenu des registre des pertes d'unités logiques (LOU) directement :

```
Sup2T#show platform software acl lou
LOUs Registers (shadow copies)

Index      Type  A_Op  A_Val  A_Cnt  B_Op  B_Val  B_Cnt
```



```

-----
0PKT_QOS_GI A is free.          NEQ      0      1
1  DST_PORT    LT      81      1 B is free.
2  B & A are free
3  B & A are free
4  B & A are free

```

snip

```

Sup2T#show platform hardware acl lou
Dumping h/w lou values

```

Index	lou_mux_sel	A_Opcode	A_Value	B_Opcode	B_Value
0	7	NEQ	0	NEQ	0
1	1	LT	81	NEQ	0
2	0	NEQ	0	NEQ	0
3	0	NEQ	0	NEQ	0

snip

Comme vous pouvez voir, a (le gt, X) tuple obtient programmé en tant que (LT, X+1) dans les registre des pertes d'unités logiques (LOU).

Note: L4Ops obtiennent programmé dans des registre des pertes d'unités logiques (LOU) SEULEMENT quand ils sont appliqués aux interfaces. Si ACLs sont créés avec L4Ops (sans ACL étant appliqué réellement à une interface), il ne programme pas le L4Ops applicable dans des registre des pertes d'unités logiques (LOU).

Retirez l'ACL de l'interface vlan 1 et regardez les registre des pertes d'unités logiques (LOU) de nouveau :

```

Sup2T(config)#int vlan 1
Sup2T(config-if)#no ip access-group a3 in

```

```

Sup2T#show platform software acl lou
LOUs Registers (shadow copies)

```

Index	Type	A_Op	A_Val	A_Cnt	B_Op	B_Val	B_Cnt
0	PKT_QOS_GI A is free.				NEQ	0	1
1	B & A are free						
2	B & A are free						
3	B & A are free						
4	B & A are free						

snip

```

Sup2T#show platform hardware acl lou
Dumping h/w lou values

```

Index	lou_mux_sel	A_Opcode	A_Value	B_Opcode	B_Value
0	7	NEQ	0	NEQ	0
1	1	NEQ	0	NEQ	0
2	0	NEQ	0	NEQ	0
3	0	NEQ	0	NEQ	0

snip

Étude de cas #1 - ACLs avec des indicateurs de TCP

Les indicateurs de TCP ont un ensemble spécial de registres alloués dans la marge de registre des pertes d'unités logiques (LOU). Vous pouvez visualiser cette plage par l'intermédiaire de la commande de mappage de capmap d'acl de logiciel de show platform comme affiché ici :

```
Sup2T#show platform software acl capmap mapping
L4op_sel value      Reference
=====
0      -----      LOU0 B register
1      -----      LOU0 A register
2      -----      LOU1 B register
3      -----      LOU1 A register
.....
.....
206     -----      LOU103 B register
207     -----      LOU103 A register
208     -----      Global format match for global acl
209     -----      Group id present
210     -----      L4_hdr_vld
211     -----      Mpls_plus_ip_pkt
212     -----      ife/ofe for direction
(213-223)  ----      Reserved
(224-239)  ----      16 TCP flags map
(240-255)  ----      16 IPv6 ext header map
```

Les valeurs L4op_sel 224-239 sont disponibles pour les utiliser pour des indicateurs de TCP, qui te donne un ensemble de 16 inscriptions à l'utilisation. Voici un exemple simple au demostrate ceci. Cet ACL est défini :

```
Sup2T#show platform software acl capmap mapping
L4op_sel value      Reference
=====
0      -----      LOU0 B register
1      -----      LOU0 A register
2      -----      LOU1 B register
3      -----      LOU1 A register
.....
.....
206     -----      LOU103 B register
207     -----      LOU103 A register
208     -----      Global format match for global acl
209     -----      Group id present
210     -----      L4_hdr_vld
211     -----      Mpls_plus_ip_pkt
212     -----      ife/ofe for direction
(213-223)  ----      Reserved
(224-239)  ----      16 TCP flags map
(240-255)  ----      16 IPv6 ext header map
```

Appliquez ceci d'arrivée sur l'interface vlan 13 :

```
Sup2T(config)#int vlan 13
Sup2T(config-if)#ip access-group a13 in
Sup2T(config-if)#end
```

```
Sup2T#show platform hardware acl entry interface vlan 13 security in ip detail
```



```
Sup2T(config)#ip access-list extended a29
Sup2T(config-ext-nacl)#permit tcp host 192.168.1.1 host 192.168.1.2 psh rst
```

```
Sup2T(config-if)#int vlan 29
Sup2T(config-if)#ip access-group a29 in
```

***Oct 6 13:57:47.612: %FMCORE-4-RACL_REDUCED: Interface Vlan29 routed traffic will be software switched in ingress direction. L2 features may not be applied at the interface**

```
Sup2T#show platform hardware acl entry interface vlan 29 security in ip
mls_if_index:2000001D dir:0 feature:0 proto:0
```

pass#0 features

fno:0

tcam:B, bank:1, prot:0 Aces

Bridge ip any any

Étude de cas #2 - utilisation de registre des pertes d'unités logiques (LOU) de 100%

Souvenez-vous que LOUs sont une ressource finie - vous pouvez manquer de l'espace pour ceux aussi bien. Vous pouvez surveiller l'utilisation du LOU avec cette commande :

```
Sup2T#show platform hardware capacity acl
```

Classification Mgr Tcam Resources

Key: Ttlent - Total TCAM entries, QoSent - QoS TCAM entries, LOU - LOUs,
RBLent - RBACL TCAM entries, Lbl - Labels, TCP - TCP Flags,
Dsttbl - Destinfo Table, Ethcam - Ethertype Cam Table,
ACTtbl - Accounting Table, V6ext - V6 Extn Hdr Table

Module	Ttlent	QoSent	RBLent	Lbl	LOU	TCP	Dsttbl	Ethcam	ACTtbl	V6ext
1	2%	7%	0%	1%	1%	0%	1%	0%	0%	0%
3	2%	7%	0%	1%	1%	0%	1%	0%	0%	0%
4	2%	7%	0%	1%	1%	0%	1%	0%	0%	0%
6	2%	7%	0%	1%	1%	0%	2%	0%	0%	0%

Mesurez l'ACLs afin d'utiliser plus de LOUs. Après l'installation des plusieurs ACLs (avec la commande de plage qui prend deux registre des pertes d'unités logiques (LOU), A et B), cette utilisation du LOU d'expositions 96% d'exemple :

```
Sup2T#show platform hardware capacity acl
```

Classification Mgr Tcam Resources

Key: Ttlent - Total TCAM entries, QoSent - QoS TCAM entries, LOU - LOUs,
RBLent - RBACL TCAM entries, Lbl - Labels, TCP - TCP Flags,
Dsttbl - Destinfo Table, Ethcam - Ethertype Cam Table,
ACTtbl - Accounting Table, V6ext - V6 Extn Hdr Table

Module	Ttlent	QoSent	RBLent	Lbl	LOU	TCP	Dsttbl	Ethcam	ACTtbl	V6ext
1	3%	7%	0%	1%	96%	0%	1%	0%	0%	0%
3	3%	7%	0%	1%	96%	0%	1%	0%	0%	0%
4	3%	7%	0%	1%	96%	0%	1%	0%	0%	0%

6 3% 7% 0% 1% 96% 0% 2% 0% 0% 0%

Créez un autre ACL et appliquez-vous cela à une interface qui ferait dépasser l'utilisation du LOU 100%.

```
Sup2T(config)#ip access-list extended a12
Sup2T(config-ext-nacl)#$68.14.1 host 192.168.14.2 range 1401 1410
Sup2T(config-ext-nacl)#$68.14.1 host 192.168.14.2 range 1411 1420
Sup2T(config-ext-nacl)#$68.14.1 host 192.168.14.2 range 1421 1430
Sup2T(config-ext-nacl)#$68.14.1 host 192.168.14.2 range 1431 1440
Sup2T(config-ext-nacl)#$68.14.1 host 192.168.14.2 range 1441 1450
Sup2T(config-ext-nacl)#$68.14.1 host 192.168.14.2 range 1451 1460
Sup2T(config-ext-nacl)#$68.14.1 host 192.168.14.2 range 1461 1470
Sup2T(config-ext-nacl)#$68.14.1 host 192.168.14.2 range 1471 1480
Sup2T(config-ext-nacl)#$68.14.1 host 192.168.14.2 range 1481 1490
Sup2T(config-ext-nacl)#$68.14.1 host 192.168.14.2 range 1491 1500
```

```
Sup2T(config-ext-nacl)#exit
Sup2T(config)#int vlan 12
Sup2T(config-if)#ip access-group a12 in
```

L'exemple a atteint l'utilisation du LOU de 100% ; cependant, notez qu'aucun message d'erreur n'a été reçu.

```
Sup2T#show platform hardware capacity acl
```

Classification Mgr Tcam Resources

Key: Ttlent - Total TCAM entries, QoSent - QoS TCAM entries, LOU - LOUs,
RBLent - RBACL TCAM entries, Lbl - Labels, TCP - TCP Flags,
Dsttbl - Destinfo Table, Ethcam - Ethertype Cam Table,
ACTtbl - Accounting Table, V6ext - V6 Extn Hdr Table

Module	Ttlent	QoSent	RBLent	Lbl	LOU	TCP	Dsttbl	Ethcam	ACTtbl	V6ext
1	3%	7%	0%	1%	100%	0%	1%	0%	0%	0%
3	3%	7%	0%	1%	100%	0%	1%	0%	0%	0%
4	3%	7%	0%	1%	100%	0%	1%	0%	0%	0%
6	3%	7%	0%	1%	100%	0%	2%	0%	0%	0%

Voici un autre test. Maintenant que le LOU est à 100%, prenez un L4Op très simple et l'essayez d'installer cela pour une interface. Configurez cet ACL :

```
Sup2T#show ip access-lists a13
Extended IP access list a13
 10 permit tcp host 192.168.14.1 host 192.168.14.2 range 1600 1650
```

Appliquez-vous ceci d'arrivée à l'interface vlan 13.

```
Sup2T#show run int vlan 13
Building configuration...

Current configuration : 87 bytes
!
interface Vlan13
 ip address 192.168.13.1 255.255.255.0
 ip access-group a13 in
end
```

Regardez le TCAM pour ce VLAN maintenant :

```
Sup2T#show platform hardware acl entry interface vlan 13 sec in ip
```



```

0
I M 16139 0x1FFF 0 0x00 0x000 255.255.255.255 - 255.255.255.255
0xF
I V 16140 14 0 0 0 192.168.14.1 - 192.168.14.2
- 1 0 1 - ----- - 0x00000000000000038
0

```

```

Sup2T#show platform software acl capmap tcam B label 14
Shadow Capmap Table Entry For TCAM B

```

```

-----
Output in a RST/INV/CNT format: RST - result value; INV - inverted;
                               CNT - aggregated reference account;

```

```

CBF - number of free cap bits (one per entry);
Free items are not shown

```

```

-----
Index   CBF      [9]          [8]          [7]          [6]
[5]     [4]          [3]          [2]          [1]
[0]

```

```

-----
-----
-----
14      9 212/0/1          Free          Free          Free
Free          Free          Free          Free          Free
Free

```

Voici une explication de ce qui s'est produit. Puisque les registre des pertes d'unités logiques (LOU) sont pleins, vous pouvez plus n'installer n'importe quel nouveau L4Ops là et rien ne peut être mis en référence dans la table de capmap. À ce stade, vous tentez toujours d'installer le L4Ops dans TCAM en les développant. Si les L4Ops sont non expansibles, puis vous commutateur de logiciel l'interface entière dans la direction donnée.

Queest-ce qu'une utilisation de registre des pertes d'unités logiques (LOU) de 100% implique ? Vos débuts TCAM à remplir rapidement (en raison de l'extension L4Op). Si vous tentez d'installer L4Ops non expansible, alors avec l'implémentation en cours, votre interface entière obtient le logiciel jeté un pont sur.

Sans modification maintenant, une erreur est seulement générée quand vous tentez d'installer un L4Op non expansible dans une telle situation. Cet exemple a modifié l'ACL en cours a13 qui a été appliqué à l'interface vlan 13 en plus d'un L4Op non expansible.

```

Sup2T(config)#ip access-list extended a13
Sup2T(config-ext-nacl)#permit tcp host 192.168.14.1 host 192.168.14.2 dscp 40

```

```

Oct  5 04:50:13.104: %FMCORE-4-RACL_REDUCED: Interface Vlan13 routed traffic will
be software switched in ingress direction. L2 features may not be applied at the
interface
Oct  5 04:50:13.096: %EARL_CM-DFC3-5-NOLOU: No free LOU entry available on the EARL
Oct  5 04:50:13.096: %EARL_CM-DFC1-5-NOLOU: No free LOU entry available on the EARL
Oct  5 04:50:13.096: %EARL_CM-DFC4-5-NOLOU: No free LOU entry available on the EARL

```

```

Sup2T#show platform hardware acl entry interface vlan 13 security in ip
mls_if_index:2000000D dir:0 feature:0 proto:0

```

```

pass#0 features

```


fno:0

tcam:B, bank:0, prot:0 Aces

Bridge ip any any

Étude de cas #3 - QoS programmé avec L4Ops

Les stratégies QoS pourraient également mettre en référence L4Ops ; ces L4Ops doivent être installés comme n'importe quel autre L4Op. Ceci implique que par interface, même pour vos stratégies QoS, vous êtes limité par les restrictions que le capmap ajourne et LOUs ont en soi. Voici un exemple pour illustrer ceci d'une petite manière :

```
Sup2T#show ip access-lists a1
Extended IP access list a1
 10 permit tcp host 192.168.1.10 host 192.168.2.10 dscp ef
```

```
Sup2T#show class-map a1-class
Class Map match-all a1-class (id 37)
Match access-group name a1
```

```
Sup2T#show policy-map a1-policy
Policy Map a1-policy
Class a1-class
  police cir 80000 bc 2500
    conform-action transmit
    exceed-action drop
```

Cet exemple a un policy-map associant un class-map qui appelle la liste d'accès a1 que les correspondances trafiquent de 192.168.1.10 à 192.168.2.10 qui est identifié par l'expédition expédié (E-F). Associer sur une valeur DSCP est un L4Op non expansible ; ceci est exigé pour être programmé dans un registre des pertes d'unités logiques (LOU) et pour être mis en référence par l'intermédiaire d'une entrée dans la table de capmap. Ce policy-map est maintenant d'arrivée installé à gig3/23.

```
Sup2T#show run int gig3/23
Building configuration...

Current configuration : 176 bytes
!
interface GigabitEthernet3/23
 switchport
 switchport trunk allowed vlan 1-30
 switchport mode trunk
 service-policy input a1-policy
end
```

Afin de regarder le QoS programmé pour une interface, utilisez cette commande :

```
Sup2T#show platform hardware acl entry interface gig3/23 qos in ip module 3
mls_if_index:8096000 dir:0 feature:1 proto:0
```

pass#0 features

fno:0

tcam:A, bank:0, prot:0 Aces

0x0000E0100000D00B tcp host 192.168.1.10 host 192.168.2.10 dscp eq 46
0x000000000080D00B ip any any

Détailler cette commande vous donne quelle étiquette TCAM est utilisée pour sur cette interface.

Sup2T#show platform hardware acl entry interface gig3/23 qos in ip detail module 3
mls_if_index:8096000 dir:0 feature:1 proto:0

pass#0 features
UAPRSF: U-urg, A-ack, P-psh, R-rst, S-syn, F-fin
MLGFI: M-mpls_plus_ip_pkt, L-L4_hdr_vld, G-gpid_present, F-global_fmt_match, I-ife/ofe
's' means set; 'u' means unset; '-' means don't care

```

-----
-----
-----
I      INDEX LABEL FS ACOS  AS          IP_SA          SRC_PORT
IP_DA          DST_PORT F FF L4PROT TCP-F:UAPRSF MLGFI OtherL4OPs
RSLT                                CNT
-----
-----
-----

```

fno:0

tcam:A, bank:0, prot:0 Aces

```

I V 16238      2 0 0 0 192.168.1.10 - 192.168.2.10
- 0 0 1 - ----- dscp=46; 0x0000E0100000D00B
0
I M 16238 0x1FFF 0 0x00 0x000 255.255.255.255 - 255.255.255.255
- 0 0 0xF
I V 16239      2 0 0 0 0.0.0.0 - 0.0.0.0
- 0 0 0 - ----- - 0x000000000080D00B
0
I M 16239 0x1FFF 0 0x00 0x000 0.0.0.0 - 0.0.0.0
- 0 0 0x0

```

L'étiquette TCAM qui est utilisée est 2. regardent la table de capmap pour ceci maintenant :

Sup2T#show platform software acl capmap tcam A label 2 module 3
Shadow Capmap Table Entry For TCAM A

Output in a RST/INV/CNT format: RST - result value; INV - inverted;
CNT - aggregated reference account;

CBF - number of free cap bits (one per entry);
Free items are not shown

```

-----
-----
Index  CBF      [9]          [8]          [7]          [6]
[5]    [4]    [3]          [2]          [1]
[0]
-----
-----

```

```

-----
      2      8 212/0/1
Free      Free      Free      Free      Free
Free      Free      Free      Free      Free      2/1/1

```

Note: Pour QoS TCAM, vous devez spécifier le numéro de module. Sans ceci, la sortie ne donne aucun résultat.

```
Sup2T#show platform software acl capmap mapping
```

```

L4op_sel value      Reference
=====
0      -----      LOU0 B register
1      -----      LOU0 A register
2      -----      LOU1 B register
3      -----      LOU1 A register

```

snip

Une valeur de LOU de 2 points à LOU1, registre B. Vous pouvez confirmer ceci qui programme avec cette commande :

```
Sup2T#show platform hardware acl lou
```

Dumping h/w lou values

```

Index  lou_mux_sel  A_Opcode  A_Value  B_Opcode  B_Value
-----
0      7      NEQ      0      NEQ      0
1      4      NEQ     0      NEQ     46
2      1      NEQ      0      NEQ      0

```

snip

Agrandissez la configuration.

```
Sup2T#show ip access-lists a1
```

Extended IP access list a1

```

10 permit tcp host 192.168.1.10 host 192.168.2.10 dscp ef
20 permit tcp host 192.168.2.11 host 192.168.2.11 dscp ef
30 permit tcp host 192.168.3.11 host 192.168.3.11 dscp ef
40 permit tcp host 192.168.4.11 host 192.168.4.11 dscp ef
50 permit tcp host 192.168.5.11 host 192.168.5.11 dscp ef
60 permit tcp host 192.168.6.11 host 192.168.6.11 dscp ef
70 permit tcp host 192.168.7.11 host 192.168.7.11 dscp ef
80 permit tcp host 192.168.8.11 host 192.168.8.11 dscp ef

```

```
Sup2T#show platform software acl capmap tcam A label 2 module 3
```

Shadow Capmap Table Entry For TCAM A

```

-----
Output in a RST/INV/CNT format: RST - result value; INV - inverted;
                               CNT - aggregated reference account;

```

```

CBF - number of free cap bits (one per entry);
Free items are not shown

```

```

-----
Index  CBF      [9]      [8]      [7]      [6]
[5]    [4]      [3]      [2]      [1]
[0]
-----
-----

```

```

-----
      2      8 212/0/1                Free                Free                Free
Free                Free                Free                Free                Free    2/1/8

```

Ceci n'utilise plus d'entrées ; au lieu de cela, il augmente le compte d'agrégat de référence contre la première entrée lui-même, qui semble raisonnable. D'une table de capmap et d'un point de vue de registre des pertes d'unités logiques (LOU), il n'y a aucun souci concernant la source et la destination. Ceci stocke simplement les informations L4Op. Puisqu'il s'assortit contre la même valeur DSCP sur tous les as, vous avez besoin seulement d'une entrée pour cette valeur DSCP.

Modifiez ceci de sorte que vous utilisiez neuf valeurs DSCP différentes.

```

Sup2T#show ip access-lists a1
Extended IP access list a1
 10 permit tcp host 192.168.1.10 host 192.168.2.10 dscp af11
 20 permit tcp host 192.168.2.11 host 192.168.2.11 dscp af12
 30 permit tcp host 192.168.3.11 host 192.168.3.11 dscp af13
 40 permit tcp host 192.168.4.11 host 192.168.4.11 dscp af21
 50 permit tcp host 192.168.5.11 host 192.168.5.11 dscp af22
 60 permit tcp host 192.168.6.11 host 192.168.6.11 dscp af23
 70 permit tcp host 192.168.7.11 host 192.168.7.11 dscp af31
 80 permit tcp host 192.168.8.11 host 192.168.8.11 dscp af32
 90 permit tcp host 192.168.9.11 host 192.168.9.11 dscp af33

```

Maintenant si vous regardez la table de capmap, vous voyez qu'elle est pleine :

```

Sup2T#show platform software acl capmap tcam A label 2 module 3
Shadow Capmap Table Entry For TCAM A

```

```

-----
Output in a RST/INV/CNT format: RST - result value; INV - inverted;
                               CNT - aggregated reference account;

```

```

CBF - number of free cap bits (one per entry);
Free items are not shown

```

```

-----
Index   CBF      [9]                [8]                [7]                [6]
[5]     [4]                [3]                [2]
[1]     [0]

```

```

-----
      2      0 212/0/1                10/1/1                9/1/1                8/1/1
7/1/1                6/1/1                5/1/1                4/1/1                3/1/1
2/1/1

```

Voici ce qui se produit si vous essayez et installez une autre entrée non expansible L4Op-based :

```

Sup2T(config-ext-nacl)#permit tcp host 192.168.10.11 host 192.168.10.11 dscp 2
Sup2T(config-ext-nacl)#end

```

```

%QM-4-TCAM_ENTRY: Hardware TCAM entry programming failed for slot 3 intf Gi3/23
dir IN: <CONFIG_UPDATE_REQ> TCAM Req Error: FAIL (4): Low TCAM Entries (1)
%QM-4-TCAM_ENTRY: Hardware TCAM entry programming failed for slot 3 intf Gi3/23
dir IN: <CONFIG_UPDATE_REQ> TCAM Req Error: FAIL (4): Low TCAM Entries (1)
%QM-4-TCAM_ENTRY: Hardware TCAM entry programming failed for slot 3 intf Gi3/23
dir IN: <CONFIG_UPDATE_REQ> TCAM Req Error: FAIL (4): Low TCAM Entries (1)
%QM-4-TCAM_ENTRY: Hardware TCAM entry programming failed for slot 3 intf Gi3/23

```

```
dir IN: <CONFIG_UPDATE_REQ> TCAM Req Error: FAIL (4): Low TCAM Entries (1)
%QM-4-TCAM_ENTRY: Hardware TCAM entry programming failed for slot 3 intf Gi3/23
dir IN: <CONFIG_UPDATE_REQ> TCAM Req Error: FAIL (4): Low TCAM Entries (1)
%FMCORE-6-RACL_ENABLED: Interface GigabitEthernet3/23 routed traffic is hardware
switched in ingress direction
Oct 20 17:12:54.304: %EARL_CM-DFC3-5-NOCAPMAP: No free capmap entry available
```

Regardez le TCAM pour cette interface maintenant :

```
Sup2T#show platform hardware acl entry interface gig3/23 qos in ip module 3
```

```
mls_if_index:8096000 dir:0 feature:1 proto:0
```

```
Couldnt find feature for mls_if_index 0x8096000, dir 0
```

Aucune des caractéristiques de QoS n'est installée désormais dans le TCAM pour cette interface.

Notez que le repérage ne consomme aucun L4Ops. Ainsi si vous avez un ACL simple qui ne fait pas placer L4Ops et vous une valeur DSCP sur la correspondance, aucun registre des pertes d'unités logiques (LOU) n'est utilisé pour ceci. Voici un exemple :

```
Sup2T#show policy-map a1-policy
```

```
Policy Map a1-policy
  Class a1-class
    set dscp ef
```

```
Sup2T#show class-map a1-class
```

```
Class Map match-all a1-class (id 37)
  Match access-group name a1
```

```
Sup2T#show ip access-lists a1
```

```
Extended IP access list a1
  10 permit tcp host 192.168.1.1 host 192.168.2.1
```

Ceci est appliqué pour relier gig3/23 :

```
Sup2T#show run interface gig3/23
```

```
Building configuration...
```

```
Current configuration : 176 bytes
```

```
!
interface GigabitEthernet3/23
  switchport
  switchport trunk allowed vlan 1-30
  switchport mode trunk
  service-policy input a1-policy
end
```

```
Sup2T#show platform hardware acl entry interface gig3/23 qos in ip detail module 3
```

```
mls_if_index:8096000 dir:0 feature:1 proto:0
```

```
pass#0 features
```

```
UAPRSF: U-urg, A-ack, P-psh, R-rst, S-syn, F-fin
```

```
MLGFI: M-mpls_plus_ip_pkt, L-L4_hdr_vld, G-gpid_present, F-global_fmt_match, I-ife/ofe
```

```
's' means set; 'u' means unset; '-' means don't care
```

```
-----
-----
-----
```


Free items are not shown

```
-----  
Index   CBF      [9]          [8]          [7]          [6]  
[5]     [4]      [3]          [2]          [1]  
[0]  
-----  
-----  
-----  
3       9 212/0/1          Free          Free          Free  
Free     Free          Free          Free          Free  
Free
```

Comme vu dans l'exemple précédent, l'ACL d'ipv4 a plus de neuf seuls L4Ops expansibles. Sous une interface configurée avec seulement l'ipv4 ceux-ci seront développés comme nécessaire afin de ne pas épuiser la table de capmap.

En programmant ces derniers dans le matériel TCAM dans un environnement de double-pile, les débuts de commutateur avec l'ACL d'ipv4 d'abord. Avec les entrées libres insuffisantes dans la table de capmap, le commutateur développe une partie du L4Ops expansible afin de remplir table de capmap sans la dépasser. Le résultat est que maintenant le nombre d'entrées libres dans la table est 0, qui signifie qu'il n'y a maintenant aucune entrée disponible pour programmer le L4Op non expansible qui est exigé quand vous allez programmer l'ACL d'IPv6 ensuite.

Quand vous retirez seulement l'ACL d'ipv4, le nombre d'entrées libres dans la table de capmap augmente, et l'ACL d'IPv6 est maintenant correctement programmé dans le matériel et utilise une des entrées nouvellement libérées de capmap. Quand l'ACL d'ipv4 est réappliqué à la configuration d'interface, la même extension se produit de nouveau. Seulement maintenant une entrée supplémentaire d'ipv4 est développée en raison de l'ACL d'IPv6 qui épuise une valeur libre de capmap. Puisque tous les L4Ops sont expansibles dans cet ACL que la programmation réussit.

Afin d'empêcher la suppression manuelle et l'ajout de l'ACL d'ipv4 pour permettre aux entrées pour fusionner dans le matériel, une amélioration a été créée pour changer le TCAM fusionnant l'algorithme dans de tels scénarios. Voir le pour en savoir plus de l'ID de bogue Cisco [CSCuq24924](#).

En raison de cette amélioration, « Réparer-dans » des releases de logiciel aura maintenant une option configurable en configuration globale qui change la manière que L4Ops sont programmés dans les exemples tels qu'un déploiement d'ACL de la double-pile IPv4/v6. C'est la modification de configuration qui peut être apportée

```
Switch(config)#platform hardware acl tcam-exp-logic enable
```

Note: En raison des modifications introduites en raison de cette amélioration, L4Ops expansibles sont développés à un plus grand que normal de débit et pourraient entraîner une augmentation significative dans l'utilisation TCAM en raison de l'extension.