Comprendre CAPWAP AP AP PMTU Discovery

Table des matières

Introduction

Scénario et portée

- Contrôle CAPWAP et données (éléments négociés)
- Faits : Paquet CAPWAP de taille maximale
- Contrôles PMTU en trois étapes

Mécanisme de détection CAPWAP PMTU

- Comportement IOS AP
  - Phase de jonction AP
  - Phase d'exécution
- Comportement COS AP
  - Phase de jonction AP
  - Phase d'exécution

Conclusion (Résumé de l'algorithme)

CDET apparentés

Introduction

Ce document décrit le mécanisme de détection de l'unité de transmission maximale (PMTU) du chemin du point d'accès CAPWAP sur IOS® XE et COS, les problèmes et leur résolution.

Scénario et portée

Les problèmes PMTU apparaissent généralement lorsqu'un point d'accès (AP) CAPWAP sur un site distant s'enregistre auprès d'un contrôleur LAN sans fil (WLC) sur un WAN, en particulier lorsque le chemin inclut un VPN, un GRE ou tout segment de réseau avec un MTU inférieur aux 1 500 octets standard.

Nous examinons également l'authentification avec EAP-TLS (Extensible Authentication Protocol Transport Layer Security). EAP-TLS échangeant de gros certificats, un MTU de chemin réduit augmente le risque de fragmentation.

Tous les journaux ont été capturés sur la version de code 17.9.3. Les sorties sont tronquées pour afficher uniquement les lignes pertinentes.

Contrôle CAPWAP et données (éléments négociés)

Contrôle CAPWAP :
Le canal de contrôle gère les messages de gestion critiques tels que les demandes de jointure, les échanges de configuration et les signaux de test d'activité. Ces messages sont sécurisés à l'aide du protocole DTLS et constituent le point central du processus de négociation de la MTU de

chemin (PMTU) afin d'assurer une communication fiable et efficace du plan de contrôle.

Données CAPWAP :
Ce canal transporte le trafic client encapsulé, généralement également protégé par DTLS dans la plupart des déploiements. Pendant la négociation PMTU sur le canal de contrôle, les valeurs PMTU résultantes déterminent indirectement la taille de paquet maximale pour l'encapsulation du plan de données, ce qui a un impact sur la fiabilité et la fragmentation de la transmission des données du client.

Exemples

- Paquets de contrôle : Demandes et réponses de jointure, mises à jour de configuration et messages d'écho/de test d'activité.
- Paquets de données : Trames client encapsulées transmises entre le point d'accès (AP) et le contrôleur LAN sans fil (WLC).

Faits : Paquet CAPWAP de taille maximale

IOS AP (exemple)

Taille de paquet PMTU envoyé : 1 499 octets = Ethernet + PMTU CAPWAP

- Ethernet = 14 octets
- CAPWAP PMTU = 1485 octets
    - IP externe = 20 octets
    - UDP = 25 octets
    - DTLS = 1 440 octets

AP-COS (exemple)

Taille de paquet PMTU envoyé : 1 483 octets = Ethernet + PMTU CAPWAP

- Ethernet = 14 octets
- CAPWAP PMTU = 1469 octets
    - IP externe = 20 octets
    - UDP = 25 octets
    - DTLS = 1 424 octets

Contrôles PMTU en trois étapes

Les deux plates-formes analysent trois valeurs PMTU codées en dur : 576, 1005 et 1485. La différence réside dans la manière dont chaque plate-forme compte l'en-tête Ethernet :

- Les AP IOS n'incluent pas l'en-tête Ethernet dans les valeurs 576/1005/1485.
    - Trame totale = Ethernet (14) + PMTU (576/1005/1485) ⇒ 590, 1019, 1499 octets (taille du câble).

- AP-COS inclut l'en-tête Ethernet dans les valeurs 576/1005/1485.
    - Trame totale = PMTU (inclut déjà Ethernet). Ces paquets sont 14 octets plus petits sur le câble que les équivalents IOS AP.

Mécanisme de détection CAPWAP PMTU

Comportement IOS AP

Phase de jonction AP

Pendant la jonction CAPWAP, le point d'accès négocie une PMTU CAPWAP maximale de 1485 octets avec le bit DF défini. Il attend une réponse dans 5 secondes.

- Si aucune réponse ou une « fragmentation requise » ICMP n'arrive, le point d'accès revient à 576 octets pour terminer la jointure rapidement, puis tente d'augmenter la PMTU après qu'il ait atteint RUN.

Capture de paquets (exemple)

Paquet numéro 106 Une sonde de 1 499 octets (ensemble DF) s’affiche. Aucune réponse de même taille n'indique que le paquet n'a pas pu traverser le chemin sans fragmentation. Vous voyez ensuite ICMP « Fragmentation Needed » (Fragmentation requise).

17	07:41:47.427848	0.002187	10.201.166.185	10.201.234.34	CAPWAP-Cont…	264	Set	CAPWAP-Control - Discovery Request[Malformed Packet]
88	07:42:45.435367	58.0075…	10.201.166.185	10.201.234.34	DTLSv1.0	117	Set	Client Hello
92	07:42:45.437784	0.002417	10.201.166.185	10.201.234.34	DTLSv1.0	137	Set	Client Hello
98	07:42:45.667215	0.229431	10.201.166.185	10.201.234.34	DTLSv1.0	590	Set	Certificate (Fragment)
99	07:42:45.667260	0.000045	10.201.166.185	10.201.234.34	DTLSv1.0	590	Set	Certificate (Fragment)
100	07:42:45.667293	0.000033	10.201.166.185	10.201.234.34	DTLSv1.0	178	Set	Certificate (Reassembled)
101	07:42:45.667316	0.000023	10.201.166.185	10.201.234.34	DTLSv1.0	329	Set	Client Key Exchange
102	07:42:45.667347	0.000031	10.201.166.185	10.201.234.34	DTLSv1.0	329	Set	Certificate Verify
103	07:42:45.667372	0.000025	10.201.166.185	10.201.234.34	DTLSv1.0	60	Set	Change Cipher Spec
104	07:42:45.667394	0.000022	10.201.166.185	10.201.234.34	DTLSv1.0	123	Set	Encrypted Handshake Message
106	07:42:45.674895	0.007501	10.201.166.185	10.201.234.34	DTLSv1.0	1499	Set	Application Data
107	07:42:45.675288	0.000393	10.201.166.161	10.201.166.185	ICMP	70	Not set,Set	Destination unreachable (Fragmentation needed)
112	07:42:50.671019	4.995731	10.201.166.185	10.201.234.34	DTLSv1.0	411	Set	Application Data
114	07:42:50.718532	0.047513	10.201.166.185	10.201.234.34	DTLSv1.0	539	Set	Application Data
115	07:42:50.718571	0.000039	10.201.166.185	10.201.234.34	DTLSv1.0	539	Set	Application Data

Le niveau AP Debug correspondant ("debug capwap client path-mtu") montre que l'AP a essayé en premier avec 1485 octets et a attendu une réponse pendant 5 secondes. En l'absence de réponse, il envoie un autre paquet de demande de jointure d'une longueur inférieure, car il est toujours en phase de jointure et nous n'avons pas de temps à perdre. Il va à la valeur minimale pour obtenir l'AP pour rejoindre le WLC, comme indiqué dans le journal de débogage :

```
*Jul 11 18:27:15.000: CAPWAP_PATHMTU: CAPWAP_DTLS_SETUP: MTU = 1485
*Jul 11 18:27:15.000: CAPWAP_PATHMTU: Setting default MTU: MTU discovery can start with 576
*Jul 11 18:27:15.235: %CAPWAP-5-DTLSREQSUCC: DTLS connection created sucessfully peer_ip: 10.201.234.34
*Jul 11 18:27:15.235: CAPWAP_PATHMTU: Sending Join Request Path MTU payload, Length 1376, MTU 576
*Jul 11 18:27:15.235: %CAPWAP-5-SENDJOIN: sending Join Request to 10.201.234.34
...
*Jul 11 18:27:20.235: %CAPWAP-5-SENDJOIN: sending Join Request to 10.201.234.34
*Jul 11 18:27:21.479: %CAPWAP-5-JOINEDCONTROLLER: AP has joined controller c9800-CL
```

Et si vous exécutez #show capwap client rcb à ce moment, vous voyez que le MTU de l'AP CAPWAP à 576 octets :

```
3702-AP#show capwap client rcb
AdminState : ADMIN_ENABLED
Primary SwVer : 17.9.3.50
..
MwarName : c9800-CL
MwarApMgrIp : 10.201.234.34
OperationState : JOIN
CAPWAP Path MTU : 576
```

Phase d'exécution

Une fois que le point d'accès a rejoint le contrôleur LAN sans fil. Vous voyez le mécanisme de découverte PMTU en jeu, où après 30 secondes, vous pouvez voir le point d'accès commencer à négocier une valeur PMTU plus élevée en envoyant un autre paquet CAPWAP avec un jeu de bits DF de cette taille de la valeur PMTU suivante la plus élevée.

Dans cet exemple, le point d'accès a essayé une valeur de 1005 octets. Comme IOS exclut Ethernet du champ PMTU, vous voyez 1 019 octets sur le câble. Si le WLC répond, l'AP met à jour PMTU à 1005 octets. Si ce n'est pas le cas, il attend 30 secondes et réessaie.

Cette capture d'écran affiche une négociation AP réussie de 1005 PMTU (voir paquets #268 et #269). Notez que ces paquets ont des tailles différentes, ce qui est dû au fait que le WLC a un algorithme différent pour le calcul de PMTU.

266	08:36:06.777257	21.0865…	10.201.166.185	10.201.234.34	DTLSv1.0	123 Set	Application Data
267	08:36:06.778067	0.000810	10.201.234.34	10.201.166.185	DTLSv1.0	139 Set	Application Data
268	08:36:12.689324	5.911257	10.201.166.185	10.201.234.34	DTLSv1.0	1019 Set	Application Data
269	08:36:12.690257	0.000933	10.201.234.34	10.201.166.185	DTLSv1.0	987 Set	Application Data
270	08:36:12.700439	0.010182	10.201.166.185	10.201.234.34	DTLSv1.0	155 Set	Application Data
271	08:36:12.701442	0.001003	10.201.234.34	10.201.166.185	DTLSv1.0	139 Set	Application Data

Ici, le niveau AP Debug correspondant (debug capwap client pmtu) montre où l'AP a négocié avec succès le PMTU de 1005 octets et mis à jour la valeur du PMTU de l'AP.

```
*Jul 11 18:28:39.911: CAPWAP_PATHMTU: PMTU Timer Expired: Trying to send higher MTU packet 576
*Jul 11 18:28:39.911: CAPWAP_PATHMTU: PMTU Timer:Sending Path MTU packet of size 1005
*Jul 11 18:28:39.911: CAPWAP_PATHMTU: MTU = 1005 for current MTU path discovery
*Jul 11 18:28:39.911: CAPWAP_PATHMTU: Ap Path MTU payload with MTU 1005 sent 888
*Jul 11 18:28:39.911: CAPWAP_PATHMTU: Stopping the message timeout timer
*Jul 11 18:28:39.911: CAPWAP_PATHMTU: Setting MTU to : 1005, it was 576
*Jul 11 18:28:39.911: CAPWAP_PATHMTU: Updating MTU to DPAA
*Jul 11 18:28:39.915: CAPWAP_PATHMTU: Sending MTU update to WLC
*Jul 11 18:28:39.915: CAPWAP_PATHMTU: MTU = 1005 for current MTU path discovery
*Jul 11 18:28:39.915: CAPWAP_PATHMTU: Ap Path MTU payload with MTU 1005 sent 21
```

Et si vous le faites (#show capwap client rcb) à ce moment, vous constatez que le MTU de l'AP

CAPWAP à 1005 octets, Voici la sortie de show :

```
3702-AP#show capwap client rcb
AdminState : ADMIN_ENABLED
Primary SwVer : 17.9.3.50
Name : 3702-AP
MwarName : c9800-CL
MwarApMgrIp : 10.201.234.34
OperationState : UP
CAPWAP Path MTU : 1005
```

Après 30 secondes, le point d'accès tente à nouveau de négocier la valeur supérieure suivante de 1485 octets, mais le point d'accès a reçu le message ICMP inaccessible alors que l'état du point d'accès est en cours d'exécution. L'ICMP inaccessible a une valeur de saut suivant, et l'AP honore cette valeur et l'utilise pour calculer sa propre PMTU comme nous pouvons le voir dans les débogages.

```
*Jul 11 18:29:45.911: CAPWAP_PATHMTU: PMTU Timer:Sending Path MTU packet of size 1485
*Jul 11 18:29:45.911: CAPWAP_PATHMTU: MTU = 1485 for current MTU path discovery
*Jul 11 18:29:45.911: CAPWAP_PATHMTU: Ap Path MTU payload with MTU 1485 sent 1368
*Jul 11 18:29:45.911: CAPWAP_PATHMTU: Received ICMP Dst unreachable
*Jul 11 18:29:45.911: CAPWAP_PATHMTU: Src port:5246 Dst Port:60542, SrcAddr:10.201.166.185 Dst Addr:10.
*Jul 11 18:29:45.911: CAPWAP_PATHMTU: Calculated MTU 1293, last_icmp_mtu 1300
*Jul 11 18:29:48.911: CAPWAP_PATHMTU: Path MTU message could not reach WLC, Removing it from the Reliab
```

Captures du niveau AP correspondant

Notez le numéro de paquet ICMP inaccessible 281, puis le point d'accès tente de négocier une PMTU en honorant la valeur de tronçon suivant ICMP sur 1300 octets sur le numéro de paquet 288 et la réponse sur 289 :

280	08:36:42.691876	23.9733…	10.201.166.185	10.201.234.34	DTLSv1.0	1499 Set	Application Data
281	08:36:42.692200	0.000324	10.201.166.161	10.201.166.185	ICMP	70 Not set,Set	Destination unreachable (Fragmentation needed)
282	08:36:45.695098	3.002898	10.201.166.185	10.201.234.34	CAPWAP-Data	92 Set	CAPWAP-Data Keep-Alive[Malformed Packet]
283	08:36:45.695533	0.000435	10.201.166.185	10.201.234.34	DTLSv1.0	139 Set	Application Data
284	08:36:45.695785	0.000252	10.201.234.34	10.201.166.185	CAPWAP-Data	92 Set	CAPWAP-Data Keep-Alive[Malformed Packet]
285	08:36:45.695931	0.000146	10.201.234.34	10.201.166.185	DTLSv1.0	123 Set	Application Data
286	08:36:45.696416	0.000485	10.201.166.185	10.201.234.34	DTLSv1.0	155 Set	Application Data
287	08:36:45.696981	0.000565	10.201.234.34	10.201.166.185	DTLSv1.0	139 Set	Application Data
288	08:36:48.695568	2.998587	10.201.166.185	10.201.234.34	DTLSv1.0	1307 Set	Application Data
289	08:36:48.696456	0.000888	10.201.234.34	10.201.166.185	DTLSv1.0	1275 Set	Application Data
290	08:36:48.706641	0.010185	10.201.166.185	10.201.234.34	DTLSv1.0	155 Set	Application Data
291	08:36:48.707636	0.000995	10.201.234.34	10.201.166.185	DTLSv1.0	139 Set	Application Data

Comportement COS AP

Il existe des différences dans le mécanisme de découverte pour AP-COS APs. Nous commençons par la jonction AP.

Phase de jonction AP

Au moment de la jonction, l'AP envoie une demande de jonction avec la valeur maximale et attend

cinq secondes.

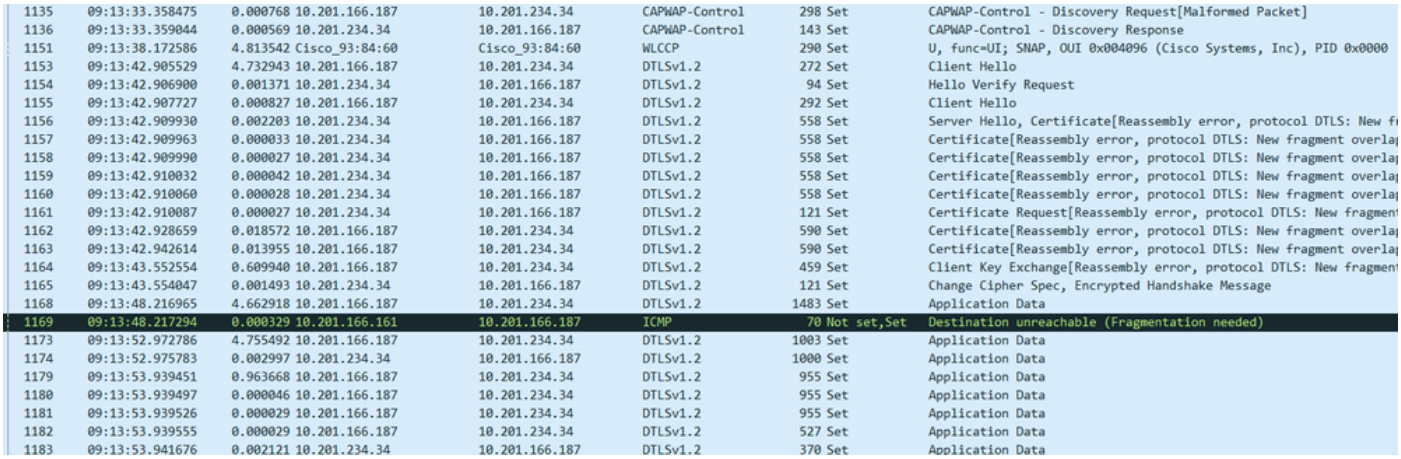
En l'absence de réponse, il réessaie et attend cinq secondes supplémentaires.

S'il n'y a toujours pas de réponse, il envoie une autre requête de jointure de 1 005 octets. Si cela réussit, il met à jour PMTU et continue (par exemple, le téléchargement d'image). Si la sonde DF de 1 005 octets ne parvient toujours pas à atteindre le contrôleur, elle passe au minimum à 576 et recommence.

Voici la commande debug capwap client pmtu au niveau du point d'accès :

```
Jul 11 19:06:10 kernel: [*07/11/2023 19:06:10.7065] AP_PATH_MTU_PAYLOAD_msg_enc_cb: request pmtu 1485, 
Jul 11 19:06:10 kernel: [*07/11/2023 19:06:10.7066] Sending Join request to 10.201.234.34 through port 
Jul 11 19:06:10 kernel: [*07/11/2023 19:06:10.7066] Sending Join Request Path MTU payload, Length 1376
..
Jul 11 19:06:15 kernel: [*07/11/2023 19:06:15.3235] AP_PATH_MTU_PAYLOAD_msg_enc_cb: request pmtu 1485, 
Jul 11 19:06:15 kernel: [*07/11/2023 19:06:15.3235] Sending Join request to 10.201.234.34 through port 
Jul 11 19:06:15 kernel: [*07/11/2023 19:06:15.3235] Sending Join Request Path MTU payload, Length 1376
Jul 11 19:06:15 kernel: [*07/11/2023 19:06:15.3245] chatter: chkcapwapicmpneedfrag :: CheckCapwapICMPNe
..
Jul 11 19:06:20 kernel: [*07/11/2023 19:06:20.0794] AP_PATH_MTU_PAYLOAD_msg_enc_cb: request pmtu 1005, 
Jul 11 19:06:20 kernel: [*07/11/2023 19:06:20.0794] Sending Join request to 10.201.234.34 through port 
Jul 11 19:06:20 kernel: [*07/11/2023 19:06:20.0794] Sending Join Request Path MTU payload, Length 896
Jul 11 19:06:20 kernel: [*07/11/2023 19:06:20.0831] Join Response from 10.201.234.34, packet size 917
Jul 11 19:06:20 kernel: [*07/11/2023 19:06:20.0832] AC accepted previous sent request with result code:
Jul 11 19:06:20 kernel: [*07/11/2023 19:06:20.0832] Received wlcType 0, timer 30
Jul 11 19:06:20 kernel: [*07/11/2023 19:06:20.5280] WLC confirms PMTU 1005, updating MTU now.
Jul 11 19:06:20 kernel: [*07/11/2023 19:06:20.5702] PMTU: Set capwap_init_mtu to TRUE and dcb's mtu to 
Jul 11 19:06:20 kernel: [*07/11/2023 19:06:20.5816] CAPWAP State: Image Data
Jul 11 19:06:20 kernel: [*07/11/2023 19:06:20.5822] AP image version 17.9.3.50 backup 17.6.5.22, Contro
```

Notez que la taille du paquet est de 1483 octets, ce qui correspond à la valeur pmtu sans l'en-tête Ethernet comme prévu pour AP-COS. Vous voyez ceci sur le paquet numéro 1168 ici :



1135	09:13:33.358475	0.000768	10.201.166.187	10.201.234.34	CAPWAP-Control	298	Set	CAPWAP-Control - Discovery Request[Malformed Packet]
1136	09:13:33.359044	0.000569	10.201.234.34	10.201.166.187	CAPWAP-Control	143	Set	CAPWAP-Control - Discovery Response
1151	09:13:38.172586	4.813542	Cisco_93:84:60	Cisco_93:84:60	WLCCP	290	Set	U, func=UI; SNAP, OUI 0x004096 (Cisco Systems, Inc), PID 0x0000
1153	09:13:42.905529	4.732943	10.201.166.187	10.201.234.34	DTLSv1.2	272	Set	Client Hello
1154	09:13:42.906900	0.001371	10.201.234.34	10.201.166.187	DTLSv1.2	94	Set	Hello Verify Request
1155	09:13:42.907727	0.000827	10.201.166.187	10.201.234.34	DTLSv1.2	292	Set	Client Hello
1156	09:13:42.909930	0.002203	10.201.234.34	10.201.166.187	DTLSv1.2	558	Set	Server Hello, Certificate[Reassembly error, protocol DTLS: New f
1157	09:13:42.909963	0.000033	10.201.234.34	10.201.166.187	DTLSv1.2	558	Set	Certificate[Reassembly error, protocol DTLS: New fragment overla
1158	09:13:42.909990	0.000027	10.201.234.34	10.201.166.187	DTLSv1.2	558	Set	Certificate[Reassembly error, protocol DTLS: New fragment overla
1159	09:13:42.910032	0.000042	10.201.234.34	10.201.166.187	DTLSv1.2	558	Set	Certificate[Reassembly error, protocol DTLS: New fragment overla
1160	09:13:42.910060	0.000028	10.201.234.34	10.201.166.187	DTLSv1.2	558	Set	Certificate[Reassembly error, protocol DTLS: New fragment overla
1161	09:13:42.910087	0.000027	10.201.234.34	10.201.166.187	DTLSv1.2	121	Set	Certificate Request[Reassembly error, protocol DTLS: New fragmen
1162	09:13:42.928659	0.018572	10.201.166.187	10.201.234.34	DTLSv1.2	590	Set	Certificate[Reassembly error, protocol DTLS: New fragment overla
1163	09:13:42.942614	0.013955	10.201.166.187	10.201.234.34	DTLSv1.2	590	Set	Certificate[Reassembly error, protocol DTLS: New fragment overla
1164	09:13:43.552554	0.609940	10.201.166.187	10.201.234.34	DTLSv1.2	459	Set	Client Key Exchange[Reassembly error, protocol DTLS: New fragmen
1165	09:13:43.554047	0.001493	10.201.234.34	10.201.166.187	DTLSv1.2	121	Set	Change Cipher Spec, Encrypted Handshake Message
1168	09:13:48.216965	4.662918	10.201.166.187	10.201.234.34	DTLSv1.2	1483	Set	Application Data
1169	09:13:48.217294	0.000329	10.201.166.161	10.201.166.187	ICMP	70	Not set,Set	Destination unreachable (Fragmentation needed)
1173	09:13:52.972786	4.755492	10.201.166.187	10.201.234.34	DTLSv1.2	1003	Set	Application Data
1174	09:13:52.975783	0.002997	10.201.234.34	10.201.166.187	DTLSv1.2	1000	Set	Application Data
1179	09:13:53.939451	0.963668	10.201.166.187	10.201.234.34	DTLSv1.2	955	Set	Application Data
1180	09:13:53.939497	0.000046	10.201.166.187	10.201.234.34	DTLSv1.2	955	Set	Application Data
1181	09:13:53.939526	0.000029	10.201.166.187	10.201.234.34	DTLSv1.2	955	Set	Application Data
1182	09:13:53.939555	0.000029	10.201.166.187	10.201.234.34	DTLSv1.2	527	Set	Application Data
1183	09:13:53.941676	0.002121	10.201.234.34	10.201.166.187	DTLSv1.2	370	Set	Application Data

Phase d'exécution

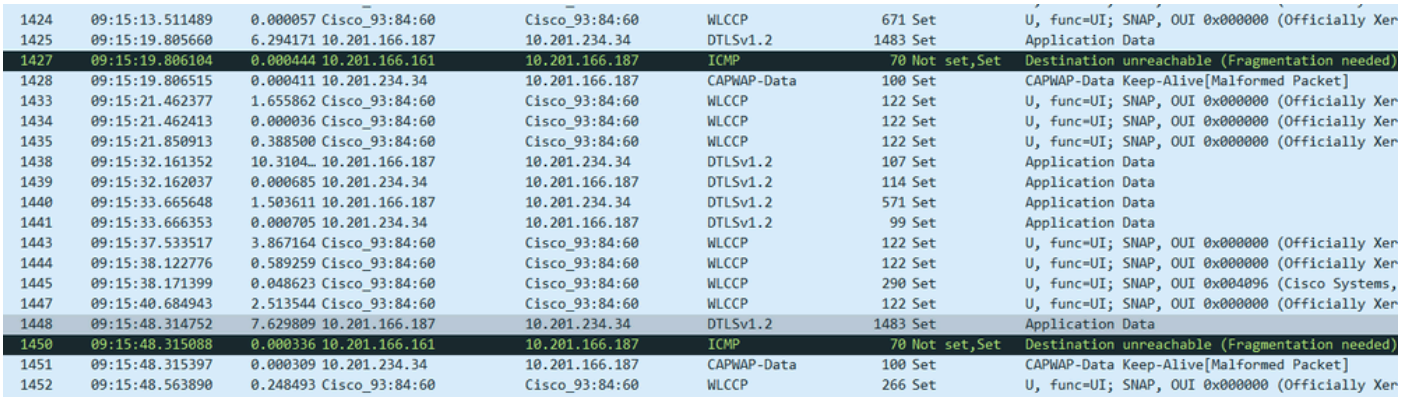
Une fois que le point d'accès atteint l'état RUN. il continue à essayer d'améliorer la PMTU toutes les 30 secondes, en

envoyant des paquets CAPWAP avec DF défini et la prochaine valeur codée en dur.

Voici le débogage au niveau AP (debug capwap client pmtu)

```
Jul 11 19:08:15 kernel: [*07/11/2023 19:08:15.1341] wtpEncodePathMTUPayload: Total Packet Size:
Jul 11 19:08:15 kernel: [*07/11/2023 19:08:15.1341] wtpEncodePathMTUPayload: Capwap Size is 137
Jul 11 19:08:15 kernel: [*07/11/2023 19:08:15.1341] [ENC]AP_PATH_MTU_PAYLOAD: pmtu 1485, len 13
Jul 11 19:08:15 kernel: [*07/11/2023 19:08:15.1341] capwap_build_and_send_pmtu_packet: packet l
Jul 11 19:08:15 kernel: [*07/11/2023 19:08:15.1343] Ap Path MTU payload sent, length 1368
Jul 11 19:08:15 kernel: [*07/11/2023 19:08:15.1343] WTP Event Request: AP Path MTU payload sent
Jul 11 19:08:15 kernel: [*07/11/2023 19:08:15.1351] pmtu icmp pkt(ICMP_NEED_FRAG) from click re
Jul 11 19:08:15 kernel: [*07/11/2023 19:08:15.1351] chatter: chkcapwapicmpneedfrag :: CheckCapwa
Jul 11 19:08:15 kernel: [*07/11/2023 19:08:15.1351] PMTU data: dcb->mtu 1005, pmtu_overhead:118
Jul 11 19:08:15 kernel: [*07/11/2023 19:08:15.1351] PMTU: Last try for next hop MTU failed
Jul 11 19:08:17 kernel: [*07/11/2023 19:08:17.9850] wtpCleanupPMTUPacket: PMTU: Found matching
..
Jul 11 19:08:43 kernel: [*07/11/2023 19:08:43.6435] wtpEncodePathMTUPayload: Total Packet Size:
Jul 11 19:08:43 kernel: [*07/11/2023 19:08:43.6435] wtpEncodePathMTUPayload: Capwap Size is 137
Jul 11 19:08:43 kernel: [*07/11/2023 19:08:43.6436] [ENC]AP_PATH_MTU_PAYLOAD: pmtu 1485, len 13
Jul 11 19:08:43 kernel: [*07/11/2023 19:08:43.6436] capwap_build-and-send_pmtu_packet: packet l
Jul 11 19:08:43 kernel: [*07/11/2023 19:08:43.6437] Ap Path MTU payload sent, length 1368
Jul 11 19:08:43 kernel: [*07/11/2023 19:08:43.6438] WTP Event Request: AP Path MTU payload sent
Jul 11 19:08:43 kernel: [*07/11/2023 19:08:43.6446] pmtu icmp pkt(ICMP_NEED_FRAG) from click re
Jul 11 19:08:43 kernel: [*07/11/2023 19:08:43.6446] chatter: chkcapwapicmpneedfrag :: CheckCapwa
Jul 11 19:08:43 kernel: [*07/11/2023 19:08:43.6446] PMTU data: dcb->mtu 1005, pmtu_overhead:118
Jul 11 19:08:43 kernel: [*07/11/2023 19:08:43.6447] PMTU: Last try for next hop MTU failed
Jul 11 19:08:46 kernel: [*07/11/2023 19:08:46.4945] wtpCleanupPMTUPacket: PMTU: Found matching
```

Voici les captures d'AP correspondantes. observez les paquets numéros 1427 et 1448 :



1424	09:15:13.511489	0.000057	Cisco_93:84:60	Cisco_93:84:60	WLCCP	671 Set	U, func=UI; SNAP, OUI 0x000000 (Officially Xer
1425	09:15:19.805660	6.294171	10.201.166.187	10.201.234.34	DTLSv1.2	1483 Set	Application Data
1427	09:15:19.806104	0.000444	10.201.166.161	10.201.166.187	ICMP	70 Not set,Set	Destination unreachable (Fragmentation needed)
1428	09:15:19.806515	0.000411	10.201.234.34	10.201.166.187	CAPWAP-Data	100 Set	CAPWAP-Data Keep-Alive[Malformed Packet]
1433	09:15:21.462377	1.655862	Cisco_93:84:60	Cisco_93:84:60	WLCCP	122 Set	U, func=UI; SNAP, OUI 0x000000 (Officially Xer
1434	09:15:21.462413	0.000036	Cisco_93:84:60	Cisco_93:84:60	WLCCP	122 Set	U, func=UI; SNAP, OUI 0x000000 (Officially Xer
1435	09:15:21.850913	0.388500	Cisco_93:84:60	Cisco_93:84:60	WLCCP	122 Set	U, func=UI; SNAP, OUI 0x000000 (Officially Xer
1438	09:15:32.161352	10.3104…	10.201.166.187	10.201.234.34	DTLSv1.2	107 Set	Application Data
1439	09:15:32.162037	0.000685	10.201.234.34	10.201.166.187	DTLSv1.2	114 Set	Application Data
1440	09:15:33.665648	1.503611	10.201.166.187	10.201.234.34	DTLSv1.2	571 Set	Application Data
1441	09:15:33.666353	0.000705	10.201.234.34	10.201.166.187	DTLSv1.2	99 Set	Application Data
1443	09:15:37.533517	3.867164	Cisco_93:84:60	Cisco_93:84:60	WLCCP	122 Set	U, func=UI; SNAP, OUI 0x000000 (Officially Xer
1444	09:15:38.122776	0.589259	Cisco_93:84:60	Cisco_93:84:60	WLCCP	122 Set	U, func=UI; SNAP, OUI 0x000000 (Officially Xer
1445	09:15:38.171399	0.048623	Cisco_93:84:60	Cisco_93:84:60	WLCCP	290 Set	U, func=UI; SNAP, OUI 0x004096 (Cisco Systems,
1447	09:15:40.684943	2.513544	Cisco_93:84:60	Cisco_93:84:60	WLCCP	122 Set	U, func=UI; SNAP, OUI 0x000000 (Officially Xer
1448	09:15:48.314752	7.629809	10.201.166.187	10.201.234.34	DTLSv1.2	1483 Set	Application Data
1450	09:15:48.315088	0.000336	10.201.166.161	10.201.166.187	ICMP	70 Not set,Set	Destination unreachable (Fragmentation needed)
1451	09:15:48.315397	0.000309	10.201.234.34	10.201.166.187	CAPWAP-Data	100 Set	CAPWAP-Data Keep-Alive[Malformed Packet]
1452	09:15:48.563890	0.248493	Cisco_93:84:60	Cisco_93:84:60	WLCCP	266 Set	U, func=UI; SNAP, OUI 0x000000 (Officially Xer

Conclusion (Résumé de l'algorithme)

En résumé, l'algorithme CAPWAP PMTUD sur les points d'accès fonctionne comme ceci.

Étape 1. La PMTU CAPWAP initiale est négociée pendant la phase de jonction AP.

Étape 2. 30 secondes plus tard, le point d'accès tente d'améliorer la PMTU CAPWAP actuelle en envoyant la valeur supérieure prédéfinie suivante (576, 1005, 1485 octets).

Étape 3 (option 1). Si le WLC répond, ajustez la PMTU CAPWAP actuelle à la nouvelle valeur et répétez l'étape 2.

Étape 3 (option 2). En l'absence de réponse, conservez la PMTU CAPWAP actuelle et répétez l'étape 2.

Étape 3 (option 3). S'il n'y a pas de réponse et qu'un ICMP Inaccessible (Type 3, Code 4) inclut un MTU de tronçon suivant, ajustez le PMTU CAPWAP à cette valeur et répétez l'étape 2.

NOTE: Reportez-vous aux correctifs pour vous assurer que la PMTU CAPWAP appropriée est utilisée lorsqu'une valeur de tronçon suivant ICMP est fournie.

CDET apparentés

Numéro de problème 1 :

ID de bogue Cisco CSCwf52815

AP-COS AP ne respectant pas la valeur de tronçon suivant ICMP Unreachable lorsque les sondes de valeur supérieure échouent.
Correctifs : 8.10.190.0, 17.3.8, 17.6.6, 17.9.5, 17.12.2.
Les AP IOS respectent la valeur de tronçon suivant et mettent à jour la PMTU.

Problème numéro 2 :

ID de bogue Cisco CSCwc05350

Le MTU asymétrique (WLC→AP diffère de AP→WLC) a conduit à un battement de PMTU lorsque le protocole ICMP ne reflétait pas le PMTU bidirectionnel maximum.
Correctifs : 8.10.181.0, 17.3.6, 17.6.5, 17.9.2, 17.10.1.
Solution de contournement: configurer le même MTU dans les deux directions sur les périphériques contrôlant le MTU (routeur, pare-feu, concentrateur VPN) entre le WLC et le point d'accès.

ID de bogue Cisco associé côté point d'accès CSCwc05364 : COS-AP améliore le mécanisme PMTU pour pouvoir identifier la taille MTU directionnelle maximale pour les MTU asymétriques

ID de bogue Cisco relatif au WLC CSCwc48316 : Améliorer les calculs PMTU pour qu'AP puisse avoir deux MTU différents l'un en amont et l'autre (marqué Fermé par DE car ils n'ont pas de plans pour répondre à cela)

Problème numéro 3 :

ID de bogue Cisco CSCwf91557

AP-COS arrête la détection PMTU après avoir atteint la valeur codée en dur maximale.

Fixé dans 17.13.1 ; également via Fixé via l'ID de bogue Cisco CSCwf52815 dans 17.3.8, 17.6.6, 17.9.5, 17.12.2.

Problème numéro 4 :

ID de bogue Cisco CSCwk70785

AP-COS ne met pas à jour la valeur MTU pour la sonde PMTU, ce qui entraîne des déconnexions.

corrigé dans le bogue Cisco ayant l'ID CSCwk90660 - APSP6 [17.9.5] Cible 17.9.6, 17.12.5, 17.15.2, 17.16.

Problème numéro 5 :

ID de bogue Cisco CSCvv53456

Configuration MTU du chemin CAPWAP statique 9800 (parité avec AireOS).

Cela permet au 9800 d'avoir un MTU de chemin CAPWAP statique configuré sur une base de profil de jointure par point d'accès. Passage à 17.17.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.