CAPWAP PMTU-Erkennung

Inhalt

Einleitung

In diesem Dokument werden der CAPWAP Access Point Path Maximum Transmission Unit (PMTU)-Erkennungsmechanismus für IOS® XE und COS sowie Probleme und deren Behebung beschrieben.

Szenario und Umfang

PMTU-Probleme treten in der Regel dann auf, wenn sich ein CAPWAP-Access Point (AP) an einem Remote-Standort bei einem Wireless LAN Controller (WLC) über ein WAN registriert, insbesondere dann, wenn der Pfad VPN, GRE oder ein Netzwerksegment mit einer MTU unterhalb der standardmäßigen 1500 Byte umfasst.

Außerdem wird die Authentifizierung mit EAP-TLS (Extensible Authentication Protocol Transport Layer Security) geprüft. Da EAP-TLS große Zertifikate austauscht, erhöht eine reduzierte Pfad-MTU das Fragmentierungsrisiko.

Alle Protokolle wurden in der Codeversion 17.9.3 erfasst. Die Ausgaben werden gekürzt, sodass nur relevante Zeilen angezeigt werden.

CAPWAP-Kontrolle und Daten (ausgehandelt)

CAPWAP-Steuerelement:
Der Steuerungskanal verarbeitet wichtige Verwaltungsnachrichten wie Join-Anforderungen, Konfigurationsaustausch und Keepalive-Signale. Diese Nachrichten werden mittels DTLS

gesichert und bilden den Hauptfokus des PMTU-Verhandlungsprozesses (Path MTU), um eine zuverlässige und effiziente Kommunikation auf Steuerungsebene sicherzustellen.

CAPWAP-Daten:
Dieser Kanal überträgt eingekapselten Client-Datenverkehr, der in den meisten Bereitstellungen in der Regel ebenfalls durch DTLS geschützt ist. Während die PMTU-Aushandlung auf dem Steuerungskanal stattfindet, bestimmen die resultierenden PMTU-Werte indirekt die maximale Paketgröße für die Datenebenenverkapselung, was sich auf die Zuverlässigkeit und Fragmentierung der Client-Datenübertragung auswirkt.

Beispiele

- Steuerungspakete: Verknüpfen von Anfragen und Antworten, Konfigurationsaktualisierungen und Echo-/Keepalive-Nachrichten
- Datenpakete: Gekapselte Client-Frames, die zwischen dem Access Point (AP) und dem Wireless LAN Controller (WLC) übertragen werden.

Fakten: Maximale CAPWAP-Paketgröße

IOS AP (Beispiel)

Paketgröße des gesendeten PMTU: 1499 Byte = Ethernet + CAPWAP PMTU

- Ethernet = 14 Byte
- CAPWAP PMTU = 1.485 Byte
    - Äußere IP = 20 Byte
    - UDP = 25 Byte
    - DTLS = 1440 Byte

AP-COS (Beispiel)

Paketgröße des gesendeten PMTU: 1483 Byte = Ethernet + CAPWAP PMTU

- Ethernet = 14 Byte
- CAPWAP PMTU = 1.469 Byte
    - Äußere IP = 20 Byte
    - UDP = 25 Byte
    - DTLS = 1.424 Byte

Dreistufige PMTU-Prüfungen

Beide Plattformen prüfen drei hartcodierte PMTU-Werte: 576, 1005 und 1485. Der Unterschied besteht darin, wie die einzelnen Plattformen den Ethernet-Header zählen:

- IOS-APs enthalten den Ethernet-Header nicht in den Werten für 576/1005/1485.
    - Frame gesamt = Ethernet (14) + PMTU (576/1005/1485) ⇒ 590, 1019, 1499 Byte (Wire-Größe).

- AP-COS enthält den Ethernet-Header in den Werten für 576/1005/1485.
    - Frame gesamt = PMTU (enthält bereits Ethernet). Diese Pakete sind in der Leitung 14 Byte kleiner als die entsprechenden IOS-APs.

CAPWAP-PMTU-Erkennungsmechanismus

IOS-AP-Verhalten

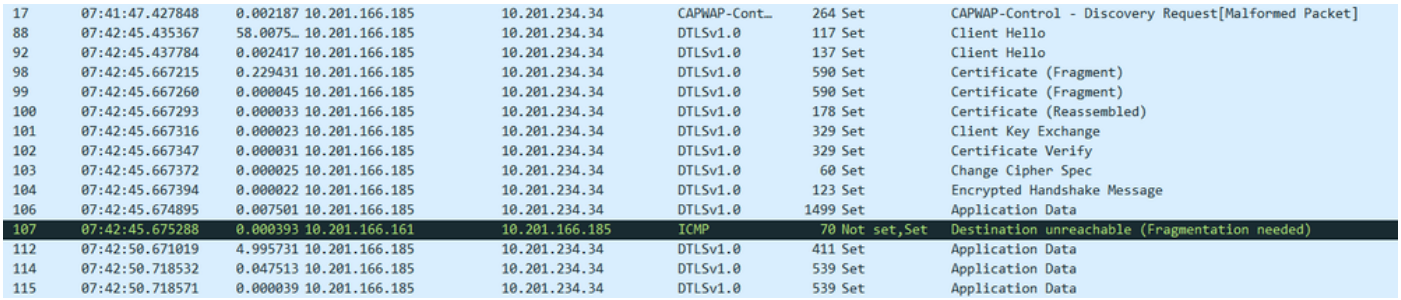
AP-Beitrittsphase

Beim CAPWAP-Beitritt handelt der WAP mit dem DF-Bit-Satz eine maximale CAPWAP-PMTU von 1485 Byte aus. Es wartet 5 Sekunden auf eine Antwort.

- Wenn keine Antwort eingeht oder eine ICMP-Fragmentierung benötigt wird, geht der Access Point auf 576 Byte zurück, um den Beitritt schnell abzuschließen. Anschließend versucht er, die PMTU nach Erreichen von RUN zu erhöhen.

Paketerfassung (Beispiel)

Packetnummer 106 Sie sehen eine 1499 Byte Probe (DF-Satz). Keine Antwort derselben Größe gibt an, dass das Paket den Pfad nicht ohne Fragmentierung durchlaufen konnte. Dann sehen Sie ICMP "Fragmentation Needed" (Fragmentierung erforderlich).



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

Das entsprechende Debug auf AP-Ebene ("debug capwap client path-mtu") zeigt, dass der Access Point zuerst einen Versuch mit 1485 Byte unternommen und 5 Sekunden auf eine Antwort gewartet hat. Wenn keine Antwort, sendet es ein weiteres Join-Request-Paket mit einer kleineren Länge, da dieses noch in der Join-Phase ist und wir keine Zeit verlieren. Es entspricht dem Mindestwert, den AP dazu zu bringen, dem WLC beizutreten, wie im Debug-Protokoll angegeben:

```
*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
```

Und wenn Sie #show capwap client rcb in diesem Moment ausführen, sehen Sie, dass die CAPWAP AP MTU bei 576 Byte:

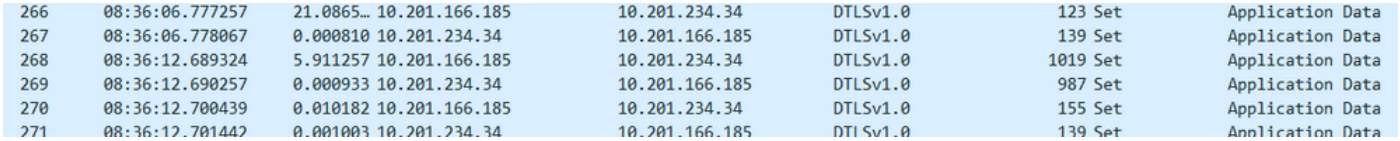
```
3702-AP#show capwap client rcb
AdminState : ADMIN_ENABLED
Primary SwVer : 17.9.3.50
..
MwarName : c9800-CL
MwarApMgrIp : 10.201.234.34
OperationState : JOIN
CAPWAP Path MTU : 576
```

RUN-Zustandsphase

Nachdem der Access Point erfolgreich am Wireless LAN-Controller angeschlossen wurde. Sie sehen den PMTU Discovery Mechanism im Spiel, wo nach 30 Sekunden der AP beginnt, einen höheren PMTU-Wert auszuhandeln, indem ein anderes CAPWAP-Paket mit einem DF-Bit-Satz von dieser Größe des nächsthöheren PMTU-Werts gesendet wird.

In diesem Beispiel hat der Access Point einen Wert von 1005 Byte verwendet. Da das IOS das Ethernet aus dem PMTU-Feld ausschließt, sehen Sie auf dem Kabel 1019 Byte. Wenn der WLC antwortet, aktualisiert der WAP die PMTU auf 1005 Byte. Falls nicht, wartet er 30 Sekunden und versucht es erneut.

Dieser Screenshot zeigt eine erfolgreiche AP-Verhandlung von 1.005 PMTU (siehe Pakete #268 und #269). Beachten Sie, dass diese Pakete unterschiedliche Größen haben, was darauf zurückzuführen ist, dass der WLC einen anderen Algorithmus für die PMTU-Berechnung verwendet.



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

Hier zeigt das entsprechende Debug auf AP-Ebene (debug capwap client pmtu) an, wo der WAP die 1005 Byte PMTU erfolgreich ausgehandelt und den WAP PMTU-Wert aktualisiert hat.

```
*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
```

Und wenn Sie (#show capwap client rcb) in diesem Moment finden Sie, dass die CAPWAP AP MTU bei 1005 Bytes, Hier ist die show output:

```
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
```

Nach 30 Sekunden versucht der WAP erneut, den nächsthöheren Wert von 1485 Byte auszuhandeln, doch der WAP empfing ICMP unreachable, während sich der WAP-Status im RUN-Zustand befindet. Der ICMP Unreachable hat einen Next-Hop-Wert, den der Access Point berücksichtigt und zur Berechnung seiner eigenen PMTU verwendet, wie in den Debugs zu sehen ist.

```
*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
```

Die entsprechende AP-Ebene erfasst

Beachten Sie die ICMP Unreachable Packet Number 281 (nicht erreichbare Paketnummer 281). Anschließend versucht der AP, eine PMTU auszuhandeln, bei der der nächste ICMP-Hop-Wert auf 1300 Byte bei Paket Number 288 und die Antwort auf 289 berücksichtigt wird:

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

COS-AP-Verhalten

Es gibt Unterschiede beim Erkennungsmechanismus für AP-COS-APs. Wir beginnen mit AP Join.

AP-Beitrittsphase

Beim Join sendet der AP eine Join-Anforderung mit dem Maximalwert und wartet fünf Sekunden.

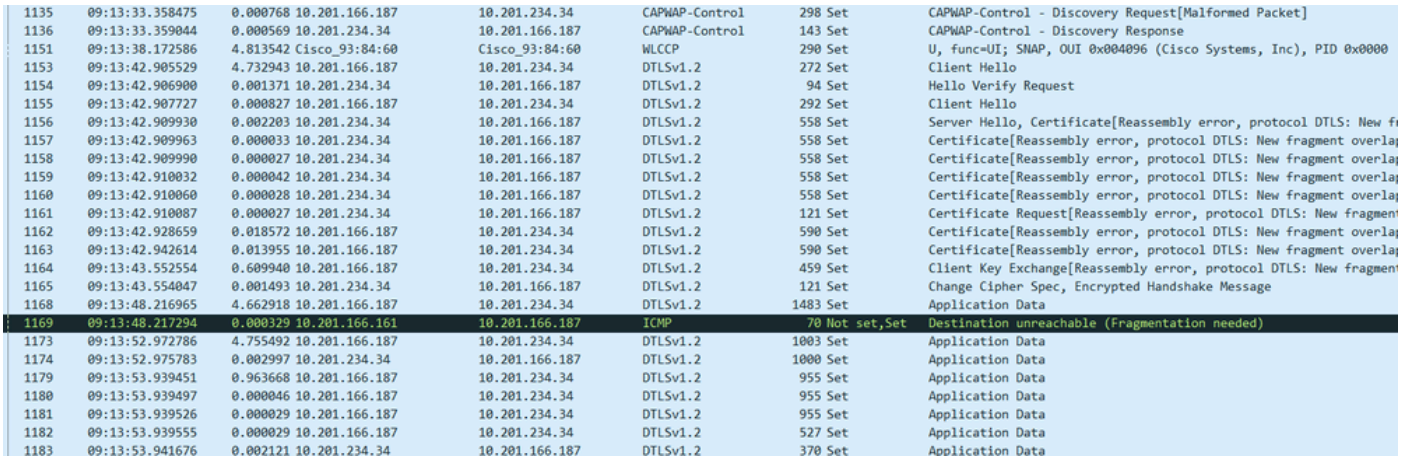
Wenn keine Antwort erfolgt, wird erneut versucht, und es werden weitere fünf Sekunden gewartet.

Wenn immer noch keine Antwort erfolgt, wird eine weitere Join-Anforderung mit 1005 Byte gesendet. Wenn dies erfolgreich ist, wird die PMTU aktualisiert und fortgesetzt (z. B. Image-Download). Wenn die 1005-Byte-DF-Sonde den Controller immer noch nicht erreichen kann, fällt sie auf das Minimum von 576 und versucht es erneut.

Hier ist die Debug-Capwap-Client-PMTU auf AP-Ebene:

```
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
```

Beachten Sie, dass die Paketgröße 1483 Byte beträgt. Dies ist der pmtu-Wert ohne Ethernet-Header, wie er für AP-COS erwartet wird. Sie sehen dies auf Paket Nr. 1168 hier:



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

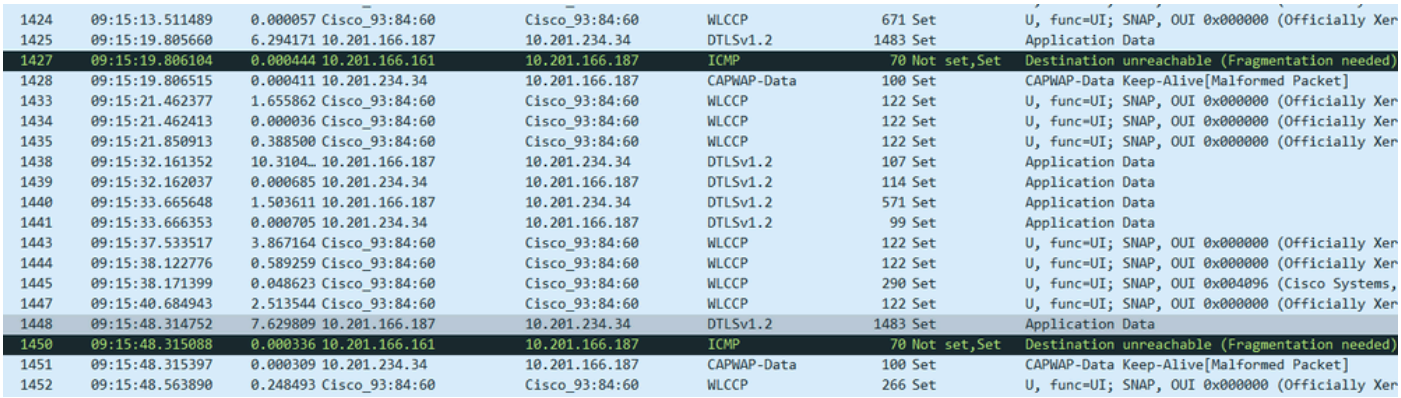
RUN-Zustandsphase

Nachdem der AP den Status "RUN" erreicht hat. Es wird weiterhin versucht, die PMTU alle 30 Sekunden zu verbessern und CAPWAP-Pakete mit festgelegtem DF und dem nächsten hartcodierten Wert zu senden.

Hier ist die AP-Ebene debug (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 :: CheckCapw
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 :: CheckCapw
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
```

Hier sind die entsprechenden AP-Aufnahmen. Schauen Sie sich die Paketnummern 1427 und 1448 an:



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

Zusammenfassung (Algorithm Summary)

Zusammenfassend lässt sich sagen, dass der CAPWAP-PMTUD-Algorithmus für Access Points in dieser Weise funktioniert.

Schritt 1: Die CAPWAP-PMTU wird während der AP-Join-Phase ausgehandelt.

Schritt 2: 30 Sekunden später versucht der WAP, die aktuelle CAPWAP PMTU zu verbessern, indem er den nächst vordefinierten höheren Wert sendet (576, 1005, 1485 Byte).

Schritt 3 (Option 1). Wenn der WLC antwortet, passen Sie die aktuelle CAPWAP-PMTU auf den neuen Wert an, und

wiederholen Sie Schritt 2.

Schritt 3 (Option 2): Wenn keine Antwort erfolgt, belassen Sie die aktuelle CAPWAP-PMTU, und wiederholen Sie Schritt 2.

Schritt 3 (Option 3). Wenn keine Antwort erfolgt und ein ICMP Unreachable (Typ 3, Code 4) eine Next-Hop-MTU enthält, stellen Sie die CAPWAP-PMTU auf diesen Wert ein, und wiederholen Sie Schritt 2.

HINWEIS: Informieren Sie sich über die Korrekturen, um sicherzustellen, dass die richtige CAPWAP-PMTU verwendet wird, wenn ein ICMP-Next-Hop-Wert bereitgestellt wird.

Zugehörige CDETs

Ausgabe 1:

Cisco Bug-ID CSCwf52815

AP-COS-APs akzeptieren den ICMP Unreachable Next-Hop-Wert nicht, wenn höherwertige Tests fehlschlagen.
Fehlerbehebung: 8.10.190,0, 17.3.8, 17.6.6, 17.9.5, 17.12.2.
IOS-APs berücksichtigen den Next-Hop-Wert und aktualisieren die PMTU.

Ausgabe 2:

Cisco Bug-ID CSCwc05350

Asymmetrische MTU (WLC→AP unterscheidet sich von AP→WLC) führte zu PMTU-Flapping, wenn ICMP nicht die maximale bidirektionale PMTU widerspiegelte.
Fehlerbehebung: 8.10.181.0, 17.3.6, 17.6.5, 17.9.2, 17.10.1.
Problemumgehung: Konfigurieren derselben MTU in beide Richtungen auf Geräten, die die MTU (Router, Firewall, VPN-Konzentrator) zwischen WLC und AP steuern.

Zugehörige AP-seitige Cisco Bug-ID CSCwc05364: COS-AP optimiert PMTU-Mechanismus zur Identifizierung der maximalen gerichteten MTU-Größe für asymmetrische MTUs

Zugehörige WLC-seitige Cisco Bug-ID CSCwc48316: Verbesserung der PMTU-Berechnungen für AP, damit diese zwei verschiedene MTUs haben können (von DE als geschlossen markiert, da sie keine Pläne haben, dies zu berücksichtigen)

Ausgabe 3:

Cisco Bug-ID CSCwf91557

AP-COS beendet die PMTU-Erkennung, nachdem der maximal hardcodierte Wert erreicht wurde.

Behoben in 17.13.1; auch über Fixed über Cisco Bug-ID CSCwf52815 in 17.3.8, 17.6.6, 17.9.5, 17.12.2.

Ausgabe 4:

Cisco Bug-ID CSCwk70785

AP-COS aktualisiert den MTU-Wert für die PMTU-Überprüfung nicht, wodurch Verbindungen getrennt werden.

behoben in Cisco Bug-ID CSCwk90660 - APSP6 17.9.5] Ziel 17.9.6, 17.12.5, 17.15.2, 17.16.

Ausgabe 5:

Cisco Bug-ID CSCvv53456

9800 Static CAPWAP Path MTU configuration (Parität mit AireOS)

Auf diese Weise kann für den 9800 eine MTU für den statischen CAPWAP-Pfad pro AP-Join-Profil konfiguriert werden. Wir gehen in den 17.17.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.