

了解CAPWAP AP PMTU发现

目录

[简介](#)

[场景和范围](#)

[CAPWAP控制与数据（协商内容）](#)

[事实：最大大小的CAPWAP数据包](#)

[三阶段PMTU检查](#)

[CAPWAP PMTU发现机制](#)

[IOS AP行为](#)

[AP加入阶段](#)

[运行状态阶段](#)

[COS AP行为](#)

[AP加入阶段](#)

[运行状态阶段](#)

[结论（算法总结）](#)

[相关CDET](#)

简介

本文档介绍IOS® XE和COS上的CAPWAP接入点路径最大传输单元(PMTU)发现机制、问题和解决方案。

场景和范围

当远程站点的CAPWAP接入点(AP)通过WAN注册到无线LAN控制器(WLC)时，尤其是当路径包括VPN、GRE或MTU低于标准1500字节的任何网段时，您通常会看到PMTU问题。

我们还检查使用可扩展身份验证协议传输层安全(EAP-TLS)的身份验证。由于EAP-TLS交换大型证书，减少的路径MTU会增加分段风险。

在代码版本17.9.3上捕获所有日志。输出被截断为仅显示相关行。

CAPWAP控制与数据（协商内容）

CAPWAP控制：

控制信道处理关键管理消息，例如加入请求、配置交换和保持连接信号。这些消息使用DTLS进行保护，并且是路径MTU(PMTU)协商过程的主要焦点，以确保可靠、有效的控制平面通信。

CAPWAP数据：

此通道传输封装的客户端流量，通常在大多数部署中还受DTLS保护。当控制信道上发生PMTU协商时，生成的PMTU值间接确定数据平面封装的最大数据包大小，从而影响客户端数据传输的可靠性和分段。

Examples

- 控制数据包:加入请求和响应、配置更新以及回应/保持连接消息。
- 数据包：在接入点(AP)和无线局域网控制器(WLC)之间传输的封装客户端帧。

事实：最大大小的CAPWAP数据包

IOS AP (示例)

发送的PMTU数据包大小：1499字节=以太网+ CAPWAP PMTU

- 以太网= 14字节
- CAPWAP PMTU = 1485字节
 - 外部IP = 20字节
 - UDP = 25字节
 - DTLS = 1440字节

AP-COS (示例)

发送的PMTU数据包大小：1483字节=以太网+ CAPWAP PMTU

- 以太网= 14字节
- CAPWAP PMTU = 1469字节
 - 外部IP = 20字节
 - UDP = 25字节
 - DTLS = 1424字节

三阶段PMTU检查

两个平台都会探测三个硬编码PMTU值：信道 576、信道 1005 和信道 1485。区别在于每个平台计算以太网报头的方式：

- IOS AP在576/1005/1485值中不包括以太网报头。
 - 帧总数=以太网(14)+ PMTU(576/1005/1485)⇒ 590、1019、1499字节 (线路大小)。
- AP-COS在576/1005/1485值中包括以太网报头。
 - 帧总数= PMTU (已包含以太网)。这些数据包在线路上比IOS AP等效数据包小14字节。

CAPWAP PMTU发现机制

IOS AP行为

AP加入阶段

在CAPWAP加入期间，AP会使用DF位设置协商最大1485字节的CAPWAP PMTU。它等待5秒以等待响应。

- 如果没有响应或ICMP“需要分段”到达，则AP将回退到576字节以快速完成加入，然后在到达RUN后尝试引发PMTU。

数据包捕获 (示例)

数据包编号106您会看到一个1499字节的探测 (DF集)。 没有大小相同的响应表示数据包无法在不分段的情况下通过路径。然后您将看到ICMP“需要分段”。

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

对应的AP级别调试(“debug capwap client path-mtu”)显示，AP首先尝试了1485字节，然后等待5秒进行响应。如果没有响应，它会发送另一个长度较小的加入请求数据包，因为此数据包仍处于加入阶段，我们没有时间浪费。如调试日志中所示，它转至最小值以使AP加入WLC:

```
*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
```

如果此时运#showcapwap客户端rcb，您会看到576字节的CAPWAP AP MTU:

```
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
```

运行状态阶段

在AP成功加入无线LAN控制器之后。您会看到正在运行的PMTU发现机制，30秒后，您可以看到

AP开始通过发送另一个CAPWAP数据包（其中的DF位集为下一个最高PMTU值的大小）来协商更高的PMTU值。

在本示例中，AP尝试了1005字节值。由于IOS从PMTU字段中排除以太网，因此您会看到线路上有1019个字节。如果WLC响应，AP将PMTU更新为1005字节。否则，它将等待30秒并再次尝试。

此屏幕截图显示1005 PMTU的AP协商成功(请参阅数据包#268和#269)。 请注意，这些数据包具有不同的大小，这是因为WLC具有不同的PMTU计算算法。

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

此处，相应的AP级别Debug(debug capwap client pmtu)显示AP成功协商1005字节PMTU并更新AP PMTU值的位置。

```
*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
```

如果您在此时已#show行（配置capwap客户端rcb），您会发现1005字节的CAPWAP AP MTU，以下是show输出：

```
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
```

30秒后，AP再次尝试协商下一个更高的值1485字节，但是，当AP处于运行状态时，AP收到ICMP unreachable。ICMP unreachable有一个下一跳值，AP将使用该值来计算自己的PMTU，如调试所示。

```
*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
```

相应的AP级别捕获

注意ICMP不可达数据包编号281，然后AP尝试通过遵循数据包编号288上1300字节的ICMP下一跳值和289上的响应来协商PMTU:

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行为

AP-COS AP的发现机制存在差异。我们从AP加入开始。

AP加入阶段

在加入时，AP会发送一个具有最大值的加入请求，并等待五秒。

如果没有响应，它会再次尝试并再等待五秒。

如果仍然没有响应，它将发送另一个包含1005字节的加入请求。如果成功，它会更新PMTU并继续（例如，映像下载）。如果1005字节的DF探针仍然无法到达控制器，它将降至最低576并重试。

以下是AP级别的debug capwap client pmtu:

```
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
```

请注意，数据包大小为1483字节，即pmtu值，没有以太网报头，这与AP-COS预期一致。您可以在此处查看数据包编号1168的以下内容：

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 fragment overlay]
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 overlay]
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 overlay]
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 overlay]
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 overlay]
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 fragment overlay]
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 overlay]
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 overlay]
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 fragment overlay]
1165	09:13:43.554047	0.001493	10.201.166.187	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

运行状态阶段

在AP达到RUN状态后。它继续尝试每隔30秒改进PMTU，发送带有DF设置和下一个硬编码值的CAPWAP数据包。

这是AP级别调试(debug capwap client pmtu)

```
Ju1 11 19:08:15 kernel: [*07/11/2023 19:08:15.1341] wtpEncodePathMTUPayload: Total Packet Size:
Ju1 11 19:08:15 kernel: [*07/11/2023 19:08:15.1341] wtpEncodePathMTUPayload: Capwap Size is 137
Ju1 11 19:08:15 kernel: [*07/11/2023 19:08:15.1341] [ENC]AP_PATH_MTU_PAYLOAD: pmtu 1485, len 13
Ju1 11 19:08:15 kernel: [*07/11/2023 19:08:15.1341] capwap_build_and_send_pmtu_packet: packet 1
Ju1 11 19:08:15 kernel: [*07/11/2023 19:08:15.1343] Ap Path MTU payload sent, length 1368
Ju1 11 19:08:15 kernel: [*07/11/2023 19:08:15.1343] WTP Event Request: AP Path MTU payload sent
Ju1 11 19:08:15 kernel: [*07/11/2023 19:08:15.1351] pmtu icmp pkt(ICMP_NEED_FRAG) from click re
Ju1 11 19:08:15 kernel: [*07/11/2023 19:08:15.1351] chatter: chkcapwapicmpneedfrag :: CheckCapw
Ju1 11 19:08:15 kernel: [*07/11/2023 19:08:15.1351] PMTU data: dcb->mtu 1005, pmtu_overhead:118
Ju1 11 19:08:15 kernel: [*07/11/2023 19:08:15.1351] PMTU: Last try for next hop MTU failed
Ju1 11 19:08:17 kernel: [*07/11/2023 19:08:17.9850] wtpCleanupPMTUPacket: PMTU: Found matching I
. .
Ju1 11 19:08:43 kernel: [*07/11/2023 19:08:43.6435] wtpEncodePathMTUPayload: Total Packet Size:
Ju1 11 19:08:43 kernel: [*07/11/2023 19:08:43.6435] wtpEncodePathMTUPayload: Capwap Size is 137
Ju1 11 19:08:43 kernel: [*07/11/2023 19:08:43.6436] [ENC]AP_PATH_MTU_PAYLOAD: pmtu 1485, len 13
Ju1 11 19:08:43 kernel: [*07/11/2023 19:08:43.6436] capwap_build-and-send_pmtu_packet: packet 1
Ju1 11 19:08:43 kernel: [*07/11/2023 19:08:43.6437] Ap Path MTU payload sent, length 1368
Ju1 11 19:08:43 kernel: [*07/11/2023 19:08:43.6438] WTP Event Request: AP Path MTU payload sent
Ju1 11 19:08:43 kernel: [*07/11/2023 19:08:43.6446] pmtu icmp pkt(ICMP_NEED_FRAG) from click re
Ju1 11 19:08:43 kernel: [*07/11/2023 19:08:43.6446] chatter: chkcapwapicmpneedfrag :: CheckCapw
Ju1 11 19:08:43 kernel: [*07/11/2023 19:08:43.6446] PMTU data: dcb->mtu 1005, pmtu_overhead:118
Ju1 11 19:08:43 kernel: [*07/11/2023 19:08:43.6447] PMTU: Last try for next hop MTU failed
Ju1 11 19:08:46 kernel: [*07/11/2023 19:08:46.4945] wtpCleanupPMTUPacket: PMTU: Found matching I
```

以下是相应的AP捕获。查看数据包编号1427和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

结论（算法总结）

总之，接入点上的CAPWAP PMTUD算法的工作方式如下。

步骤1.在AP加入阶段协商初始CAPWAP PMTU。

步骤2. 30秒后，AP尝试通过发送下一个预定义的较高值（576、1005、1485字节）来改进当前CAPWAP PMTU。

第3步（选项1）。如果WLC响应，请将当前CAPWAP PMTU调整为新值并重复步骤2。

步骤3（选项2）。如果没有响应，请保留当前的CAPWAP PMTU并重复步骤2。

第3步（选项3）。如果没有响应，并且ICMP不可达（类型3，代码4）包括下一跳MTU，请将CAPWAP PMTU调整为该值，然后重复步骤2。

NOTE:请参阅修复程序，确保在提供ICMP下一跳值时使用正确的CAPWAP PMTU。

相关CDET

问题1:

Cisco Bug ID [CSCwf52815](#)

AP-COS AP在更高值探测失败时不会认可ICMP不可达下一跳值。

修复：8.10.190.0、17.3.8、17.6.6、17.9.5、17.12.2。

IOS AP会遵循下一跳值并更新PMTU。

问题二：

Cisco Bug ID [CSCwc05350](#)

当ICMP未反映最大→双向PMTU时，非对称→（WLC-AP不同于AP-WLC）会导致PMTU抖动。

修复：8.10.181.0、17.3.6、17.6.5、17.9.2、17.10.1。

解决方法：在控制WLC和AP之间的MTU（路由器、防火墙、VPN集中器）的设备上，在两个方向上配置相同的MTU。

相关AP端思科漏洞ID [CSCwc05364](#):COS-AP改进了PMTU机制，以便能够识别非对称MTU的最大定向MTU大小

相关WLC端Cisco Bug ID [CSCwc48316](#):改进PMTU计算，使AP能够具有两个不同的MTU（一个上游和另一个）（标记为

Closed by DE , 因为它们没有解决此问题的计划)

问题三 :

Cisco Bug ID [CSCwf91557](#)

AP-COS在达到最大硬编码值后停止PMTU发现。

在17.13.1中修复 ; 也可通过思科漏洞ID [CSCwf52815](#) (位于17.3.8、17.6.6、17.9.5、17.12.2中) 修复。

问题四 :

Cisco Bug ID [CSCwk70785](#)

AP-COS未更新PMTU探测的MTU值 , 导致断开连接。

已在思科漏洞ID [CSCwk90660](#) - APSP6 17.9.5]中修复 , 目标为17.9.6、17.12.5、17.15.2、17.16。

问题五 :

Cisco Bug ID [CSCvv53456](#)

9800静态CAPWAP路径MTU配置 (与AireOS奇偶校验) 。

这允许9800基于每个AP加入配置文件配置静态CAPWAP路径MTU。 进入17.17。

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。