Entender a descoberta de PMTU do AP CAPWAP

Contents

Introdução

Este documento descreve o mecanismo de descoberta da PMTU (Unidade máxima de transmissão) do caminho do ponto de acesso do CAPWAP no IOS® XE e COS, problemas e resolução.

Cenário e escopo

Normalmente, você vê problemas de PMTU quando um ponto de acesso (AP) CAPWAP em um local remoto se registra em um controlador de LAN sem fio (WLC) através de uma WAN, especialmente quando o caminho inclui VPN, GRE ou qualquer segmento de rede com um MTU inferior aos 1500 bytes padrão.

Também examinamos a autenticação com Extensible Authentication Protocol Transport Layer Security (EAP-TLS). Como o EAP-TLS troca certificados grandes, um MTU de caminho reduzido aumenta o risco de fragmentação.

Todos os registros foram capturados na versão de código 17.9.3. As saídas são truncadas para mostrar apenas as linhas relevantes.

Controle do CAPWAP versus dados (o que é negociado)

Controle CAPWAP:

O canal de controle lida com mensagens críticas de gerenciamento, como solicitações de junção, trocas de configuração e sinais de keepalive. Essas mensagens são protegidas usando DTLS e são o foco principal do processo de negociação de MTU de Caminho (PMTU) para garantir uma comunicação de plano de controle confiável e eficiente.

Dados CAPWAP:
Esse canal transporta tráfego de cliente encapsulado, normalmente também protegido por DTLS na maioria das implantações. Enquanto a negociação de PMTU ocorre no canal de controle, os valores de PMTU resultantes determinam indiretamente o tamanho máximo do pacote para o encapsulamento do plano de dados, impactando a confiabilidade e a fragmentação da transmissão de dados do cliente.

Examples

- Pacotes de controle: Solicitações e respostas de junção, atualizações de configuração e mensagens de eco/keepalive.
- Pacotes de dados: Quadros de cliente encapsulados transmitidos entre o Ponto de Acesso (AP) e a Controladora Wireless LAN (WLC).

Fatos: Pacote CAPWAP de tamanho máximo

IOS AP (exemplo)

Tamanho do pacote PMTU enviado: 1.499 bytes = Ethernet + CAPWAP PMTU

- Ethernet = 14 bytes
- CAPWAP PMTU = 1485 bytes
    - IP externo = 20 bytes
    - UDP = 25 bytes
    - DTLS = 1440 bytes

AP-COS (Exemplo)

Tamanho do pacote PMTU enviado: 1.483 bytes = Ethernet + CAPWAP PMTU

- Ethernet = 14 bytes
- CAPWAP PMTU = 1469 bytes
    - IP externo = 20 bytes
    - UDP = 25 bytes
    - DTLS = 1424 bytes

Verificações de PMTU em Três Estágios

Ambas as plataformas testam três valores PMTU codificados: 576, 1005 e 1485. A diferença é como cada plataforma conta o cabeçalho Ethernet:

- Os APs IOS não incluem o cabeçalho Ethernet nos valores 576/1005/1485.

- Quadro total = Ethernet (14) + PMTU (576/1005/1485) ⇒ 590, 1019, 1499 bytes (tamanho do fio).

- O AP-COS inclui o cabeçalho Ethernet nos valores 576/1005/1485.
  - Quadro total = PMTU (já inclui Ethernet). Esses pacotes são 14 bytes menores no fio do que os equivalentes do AP IOS.

Mecanismo de descoberta CAPWAP PMTU

Comportamento do AP IOS

Fase de ingresso do AP

Durante a junção CAPWAP, o AP negocia um CAPWAP PMTU máximo de 1485 bytes com o bit DF definido. Ele espera 5 segundos por uma resposta.

- Se nenhuma resposta ou uma "Fragmentação necessária" de ICMP chegar, o AP recuará para 576 bytes para completar a junção rapidamente e, em seguida, tentará elevar o PMTU após alcançar RUN.

Captura de pacotes (exemplo)

Número do pacote 106 Você vê uma sonda de 1499 bytes (DF definido). Nenhuma resposta de mesmo tamanho indica que o pacote não pode atravessar o caminho sem fragmentação. Você verá ICMP "Fragmentation Needed" (A fragmentação é necessária).

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

O nível de AP correspondente Debug ("debug capwap client path-mtu") mostra que o AP tentou primeiro com 1485 bytes e esperou 5 segundos por uma resposta. Se não houver resposta, ele enviará outro pacote de solicitação de junção com um comprimento menor, pois ele ainda está na fase de junção e não temos tempo a perder. Ele vai até o valor mínimo para fazer com que o AP se una à WLC, como indicado no log de depuração:

```
*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
```

E se você executar #show capwap client rcb neste momento, verá que o CAPWAP AP AP MTU a 576 bytes:

```
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
```

Fase de estado de EXECUÇÃO

Depois que o AP se unir com êxito à controladora Wireless LAN. Você vê o Mecanismo de descoberta de PMTU em jogo, onde após 30 segundos você pode ver o AP começar a negociar um valor de PMTU mais alto enviando outro Pacote CAPWAP com o conjunto de bits DF daquele tamanho do próximo valor de PMTU mais alto.

Neste exemplo, o AP tentou um valor de 1005 bytes. Como o IOS exclui a Ethernet do campo PMTU, você vê 1019 bytes no fio. Se a WLC responder, o AP atualizará o PMTU para 1005 bytes. Caso contrário, ele espera 30 segundos e tenta novamente.

Esta captura de tela exibe uma negociação de AP bem-sucedida de 1005 PMTU (consulte os pacotes #268 e #269). Observe que esses pacotes têm tamanhos diferentes, o que se deve ao fato de a WLC ter um algoritmo diferente para o cálculo 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

Aqui, a depuração de nível de AP correspondente (debug capwap client pmtu) mostra onde o AP negociou com êxito o PMTU de 1005 bytes e atualizou o valor do PMTU do 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
```

E se você fizer (#show capwap client rcb) neste momento, você verá que o CAPWAP AP AP MTU em 1005 bytes, Aqui está a saída 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
```

Após 30 segundos, o AP tenta novamente negociar o próximo valor mais alto de 1485 bytes, ainda assim, o AP recebeu o ICMP inalcançável enquanto o status do AP está no estado RUN. O ICMP inalcançável tem um valor de próximo salto, e o AP honra esse valor e o usa para calcular seu próprio PMTU, como podemos ver nas depurações.

```
*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
```

As capturas de nível de AP correspondentes

Observe o número de pacote ICMP inalcançável 281 e, em seguida, o AP tenta negociar um PMTU honrando o valor do próximo salto ICMP em 1300 bytes nos pacotes número 288 e resposta em 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

Comportamento de COS AP

Há diferenças no mecanismo de descoberta para APs AP-COS. Começamos na junção do AP.

Fase de ingresso do AP

Na junção, o AP envia uma solicitação de junção com o valor máximo e espera cinco segundos.

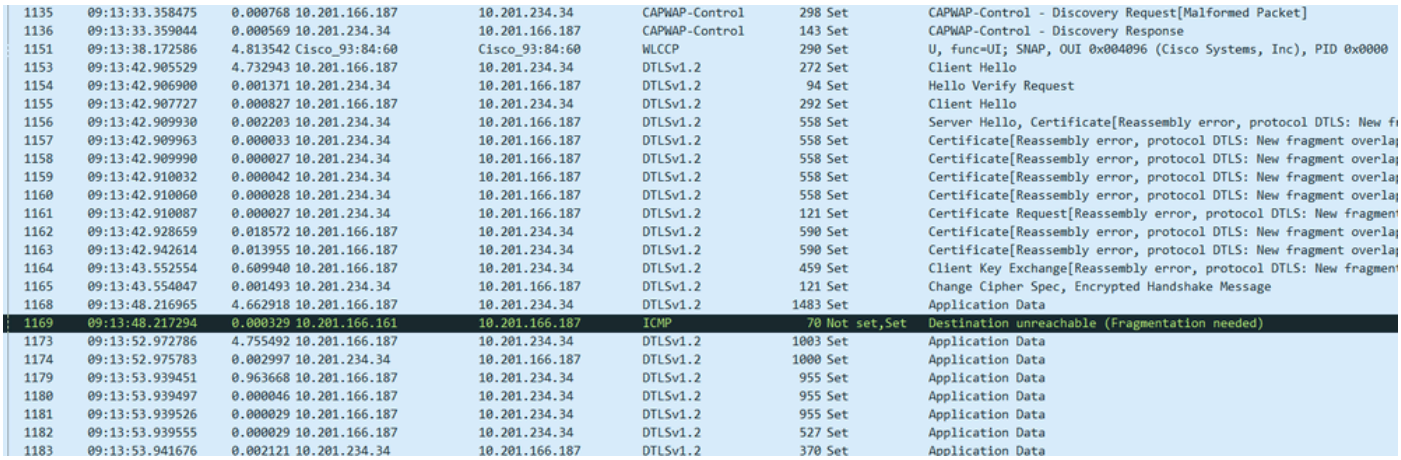
Se não houver resposta, ele tentará novamente e esperará mais cinco segundos.

Se ainda não houver resposta, ele enviará outra solicitação de junção com 1005 bytes. Se obtiver êxito, ele atualizará o PMTU e continuará (por exemplo, download de imagem). Se a sonda DF de 1005 bytes ainda não conseguir alcançar o controlador, ela cai para o mínimo de 576 e tenta novamente.

Aqui está o comando debug capwap client pmtu no nível de AP:

```
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
```

Observe que o tamanho do pacote é de 1483 bytes, que é o valor pmtu sem o cabeçalho ethernet como esperado para AP-COS. Você vê isso no pacote número 1168 aqui:



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 fragment
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 fragment
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

Fase de estado de EXECUÇÃO

Depois que o AP atingir o estado RUN. ele continua tentando melhorar o PMTU a cada 30 segundos, enviando pacotes CAPWAP com DF definido e o próximo valor embutido em código.

Aqui está a depuração de nível de 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 :: 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
```

Aqui estão as capturas de AP correspondentes. observe os números de pacote 1427 e 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

Conclusão (Resumo do algoritmo)

Em resumo, o algoritmo CAPWAP PMTUD em pontos de acesso funciona assim.

Etapa 1. O CAPWAP PMTU inicial é negociado durante a fase de junção do AP.

Etapa 2. 30 segundos depois, o AP tenta melhorar o CAPWAP PMTU atual enviando o próximo valor superior predefinido (576, 1005, 1485 bytes).

Etapa 3 (opção 1). Se a WLC responder, ajuste o CAPWAP PMTU atual para o novo valor e repita a etapa 2.

Etapa 3 (opção 2). Se não houver resposta, mantenha o CAPWAP PMTU atual e repita a etapa 2.

Etapa 3 (opção 3). Se não houver resposta e um ICMP Inalcançável (Tipo 3, Código 4) incluir um MTU do próximo salto, ajuste o CAPWAP PMTU para esse valor e repita a etapa 2.

NOTE: Consulte as correções para garantir que o CAPWAP PMTU correto seja usado quando um valor do próximo salto de ICMP for fornecido.

CDETs relacionados

Número do problema 1:

ID de bug da Cisco CSCwf52815

Os APs do AP-COS não honram o valor do próximo salto ICMP Unreachable quando as sondas de valor mais alto falham.
Correções: 8.10.190.0, 17.3.8, 17.6.6, 17.9.5, 17.12.2.
Os IOS APs honram o valor do próximo salto e atualizam o PMTU.

Número do problema 2:

ID de bug da Cisco CSCwc05350

O MTU assimétrico (WLC→AP difere do AP→WLC) levou à oscilação de PMTU quando o ICMP não refletia o PMTU bidirecional máximo.
Correções: 8.10.181.0, 17.3.6, 17.6.5, 17.9.2, 17.10.1.
Solução: configure o mesmo MTU em ambas as direções nos dispositivos que controlam o MTU (roteador, firewall, concentrador de VPN) entre o WLC e o AP.

ID de bug relacionado da Cisco no lado do AP CSCwc05364: O COS-AP melhora o mecanismo de PMTU para poder identificar o tamanho máximo de MTU direcional para MTUs assimétricas

ID de bug Cisco CSCwc48316 do lado da WLC relacionado: Melhore os cálculos de PMTU para que o AP possa ter duas MTUs diferentes, uma de upstream e outra (marcada como Fechada pelo DE, pois não tem planos de resolver isso)

Número do problema 3:

ID de bug da Cisco CSCwf91557

O AP-COS interrompe a descoberta de PMTU após atingir o valor máximo codificado.

Fixado em 17.13.1; também via Fixed via Cisco bug ID CSCwf52815 em 17.3.8, 17.6.6, 17.9.5, 17.12.2.

Número do problema 4:

ID de bug da Cisco CSCwk70785

O AP-COS não está atualizando o valor de MTU para a sonda PMTU, causando desconexões.

corrigido no bug da Cisco ID CSCwk90660 - APSP6 17.9.5] Alvo 17.9.6, 17.12.5, 17.15.2, 17.16.

Número do problema 5:

ID de bug da Cisco CSCvv53456

9800 Configuração de MTU de caminho de CAPWAP estático (paridade com AireOS).

Isso permite que o 9800 tenha um MTU de caminho CAPWAP estático configurado em uma base de perfil de junção por AP. Em 17.17.

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.