

No IP
Broadcast
WLC C
WLC A
AP wired E
AP wired A

capwap-discovery_ipv4_dns.pcapng

DHCP Discover

Transaction ID	0x000013eb
IP	0.0.0.0
Client MAC address	84:b2:61:0e:0c:18
Client MAC address	84:b2:61:0e:0c:18
Hostname	AP84b2.610e.0c18

DHCP Offer

Transaction ID	0x000013eb
IP	10.0.41.99
Client MAC address	84:b2:61:0e:0c:18

DHCP Request

Transaction ID	0x000013eb
IP	0.0.0.0
Client MAC address	84:b2:61:0e:0c:18
Client MAC address	84:b2:61:0e:0c:18
Hostname	AP84b2.610e.0c18

DHCP ACK

Transaction ID	0x000013eb
IP	10.0.41.99
Client MAC address	84:b2:61:0e:0c:18

DNS Query

Name	CISCO-CAPWAP-CONTROLLER.lab.local
Type	A (1) (Host Address)

DNS Query

Name	CISCO-CAPWAP-CONTROLLER.lab.local
Type	A (1) (Host Address)
Address	10.0.40.71
Address	10.0.10.31

CAPWAP Discovery Request

Sequence Number	0
-----------------	---

CAPWAP Discovery Request

Sequence Number	0
-----------------	---

CAPWAP Discovery Request

Sequence Number	0
-----------------	---

CAPWAP Discovery Response

Sequence Number	0
WLC	WLC2504_A

CAPWAP Discovery Response

Sequence Number	0
WLC	WLC2504_A

CAPWAP Discovery Response

Sequence Number	0
WLC	WLC2504_A

DTLS Handshake (22)

Version	DTLS 1.0 (0xfeff)
---------	-------------------

DTLS Handshake (22)

💡 DHCP – client obtains IP address after 802.11 association and EAPOL key exchange complete; DORA: Discover→Offer→Request→ACK; in WLAN, DHCP may traverse CAPWAP tunnel to WLC

💡 DHCP – client obtains IP address after 802.11 association and EAPOL key exchange complete; DORA: Discover→Offer→Request→ACK; in WLAN, DHCP may traverse CAPWAP tunnel to WLC

💡 DHCP – client obtains IP address after 802.11 association and EAPOL key exchange complete; DORA: Discover→Offer→Request→ACK; in WLAN, DHCP may traverse CAPWAP tunnel to WLC

💡 DHCP – client obtains IP address after 802.11 association and EAPOL key exchange complete; DORA: Discover→Offer→Request→ACK; in WLAN, DHCP may traverse CAPWAP tunnel to WLC

Frame 54 | 2017-01-14T12:59:58.318113Z

Frame 55 | 2017-01-14T12:59:58.31851Z

💡 CAPWAP – controller manages lightweight APs; Discovery finds WLC, Join establishes DTLS tunnel, Config provisions AP (SSID, channel, power)

💡 CAPWAP – controller manages lightweight APs; Discovery finds WLC, Join establishes DTLS tunnel, Config provisions AP (SSID, channel, power)

💡 CAPWAP – controller manages lightweight APs; Discovery finds WLC, Join establishes DTLS tunnel, Config provisions AP (SSID, channel, power)

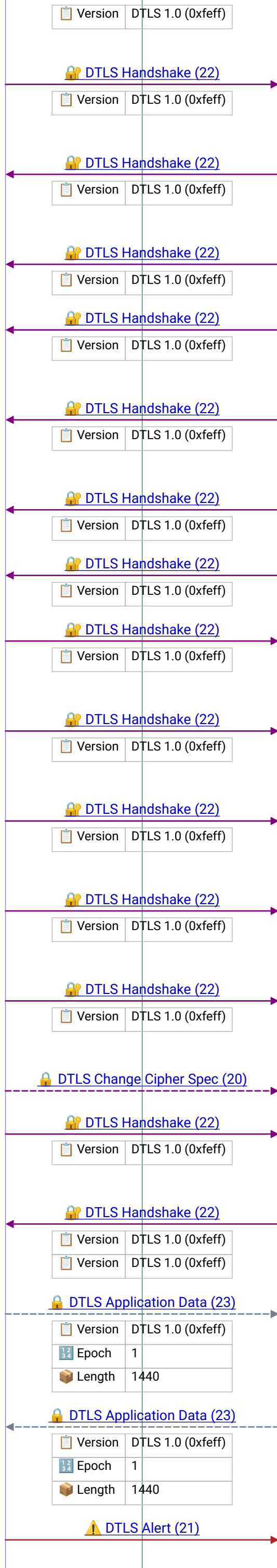
💡 CAPWAP – controller manages lightweight APs; Discovery finds WLC, Join establishes DTLS tunnel, Config provisions AP (SSID, channel, power)

💡 CAPWAP – controller manages lightweight APs; Discovery finds WLC, Join establishes DTLS tunnel, Config provisions AP (SSID, channel, power)

💡 CAPWAP – controller manages lightweight APs; Discovery finds WLC, Join establishes DTLS tunnel, Config provisions AP (SSID, channel, power)

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and



WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

Frame 75 | 2017-01-14T13:00:08.613427Z

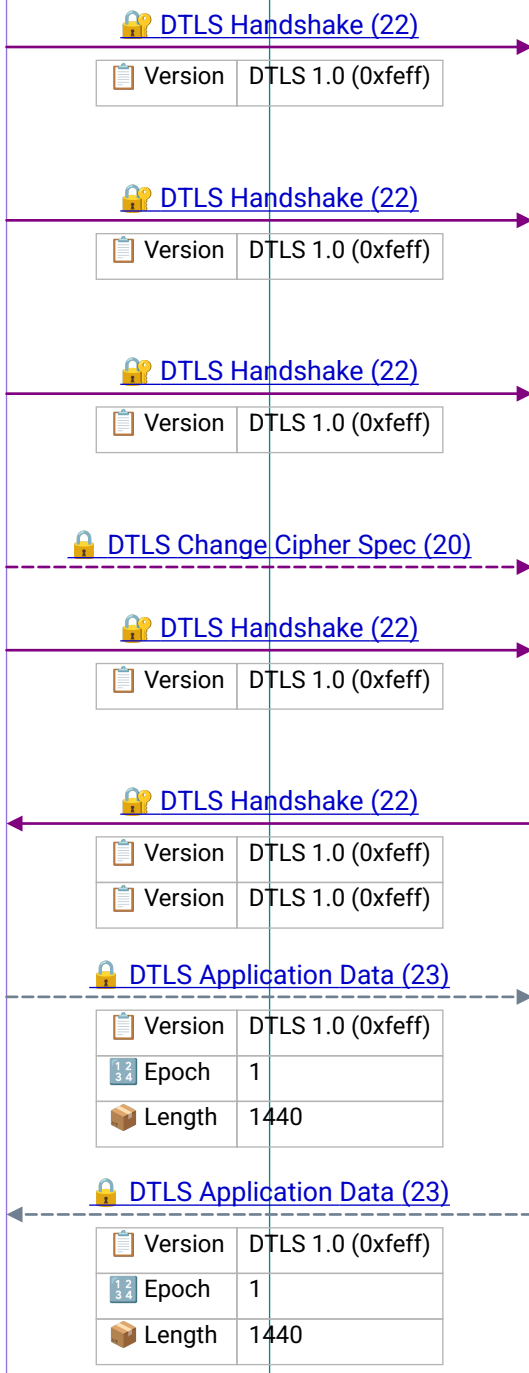
💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

Frame 78 | 2017-01-14T13:00:08.746593Z

Frame 79 | 2017-01-14T13:00:08.747985Z

Frame 85 | 2017-01-14T13:00:10.008246Z



💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

Frame 113 | 2017-01-14T13:00:21.021642Z

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

💡 DTLS Handshake – establishes encrypted tunnel between AP and WLC for CAPWAP control/data; uses UDP (not TCP) so it works with CAPWAP's UDP transport

Frame 116 | 2017-01-14T13:00:21.153169Z

Frame 117 | 2017-01-14T13:00:21.154603Z