
cisco Live!
Let's go
#CiscoLive

Cisco U.

Tech learning: shaped to you, built for Us.

The bridge to possible

Flowin' with Wireshark:

Hacks and Tips to Rule the Network

Joe Clarke, Distinguished Engineer

CISCOU-2000

cisco Live!

Packet Capture or It Didn't Happen

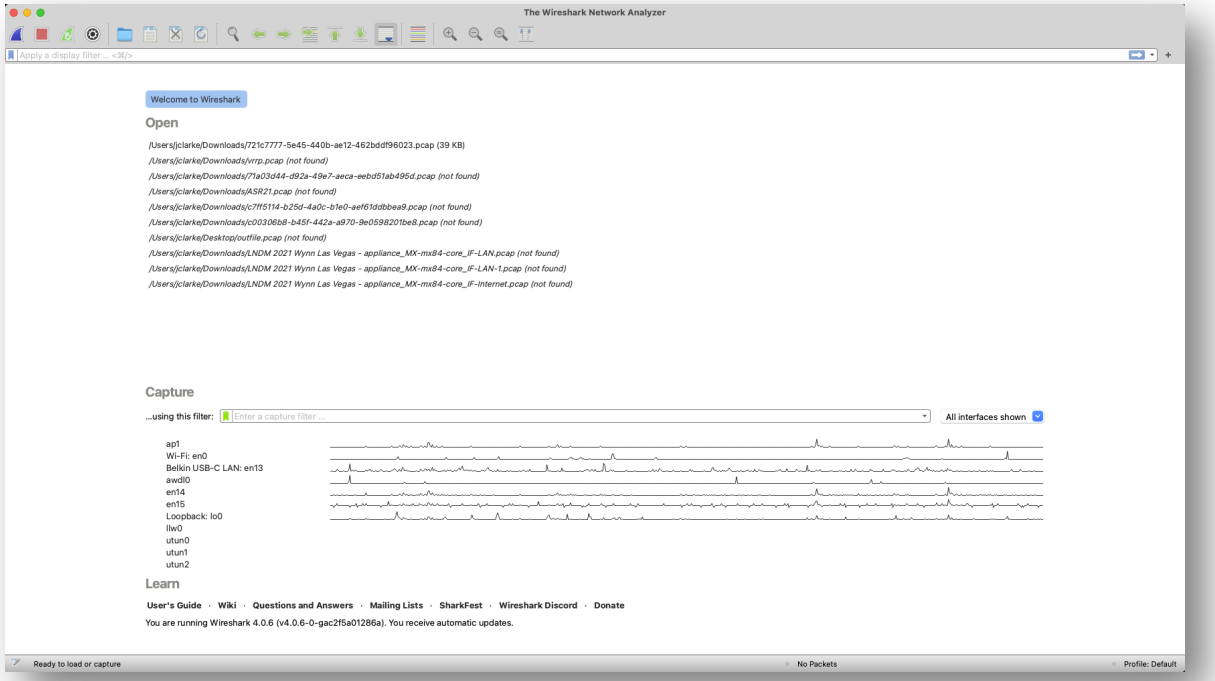
– Some wise network engineer

Agenda

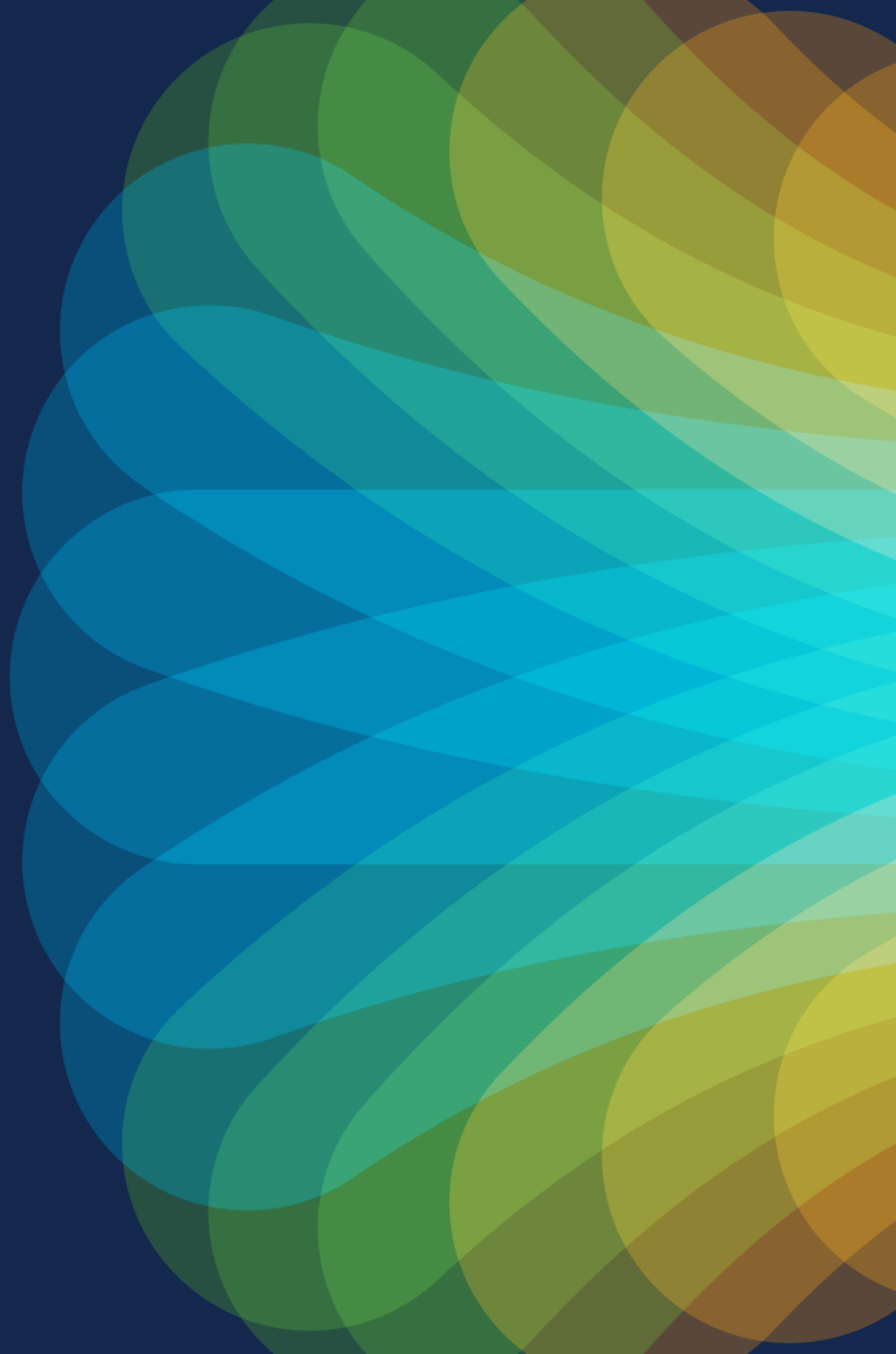
- Following Streams
- Decode As...
- Packet Diagrams
- Remote Capture
- Decrypt the Things!

But First...

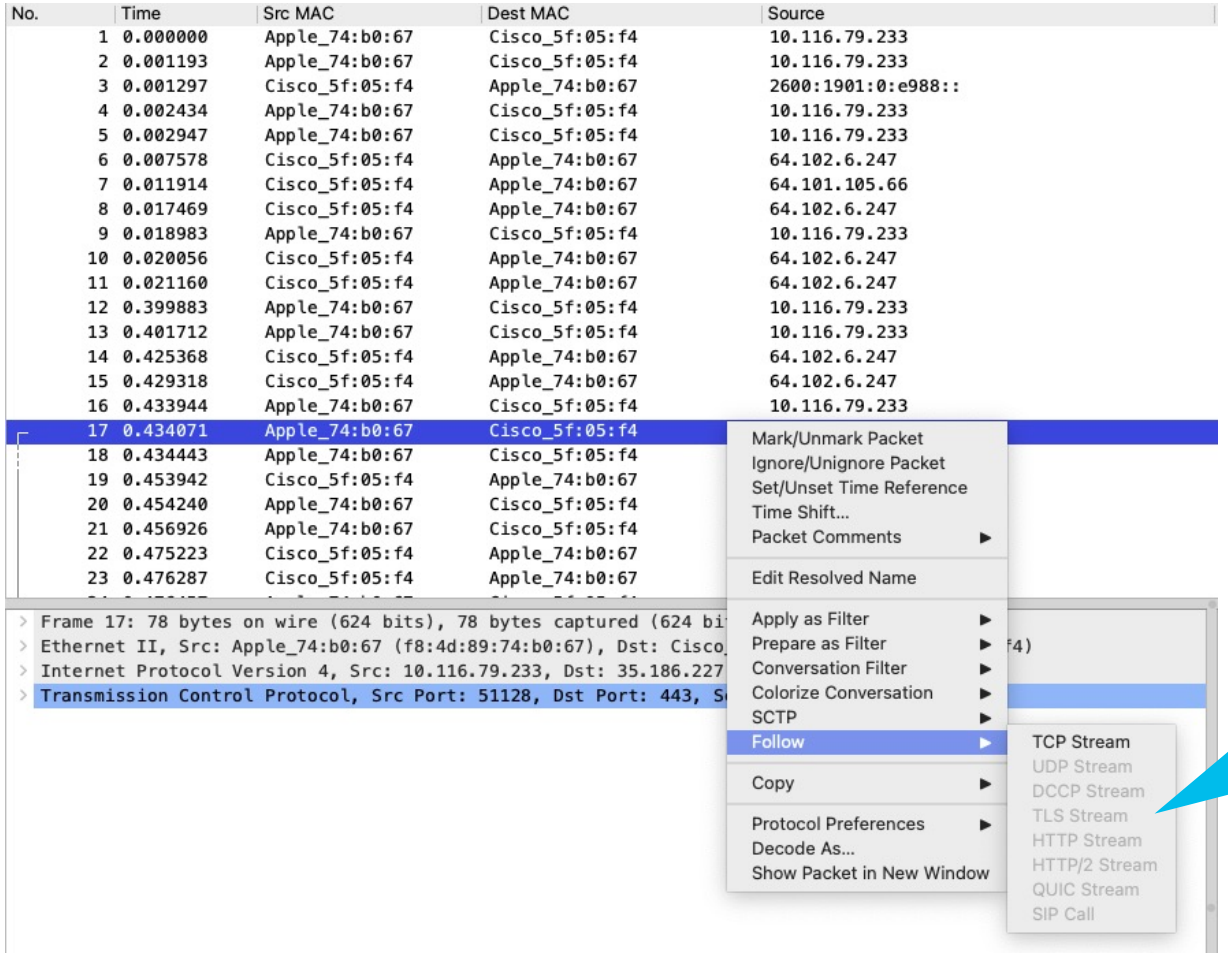
- If you don't know or use Wireshark yet...
  - GET IT NOW!
  - https://wireshark.org



Following Streams



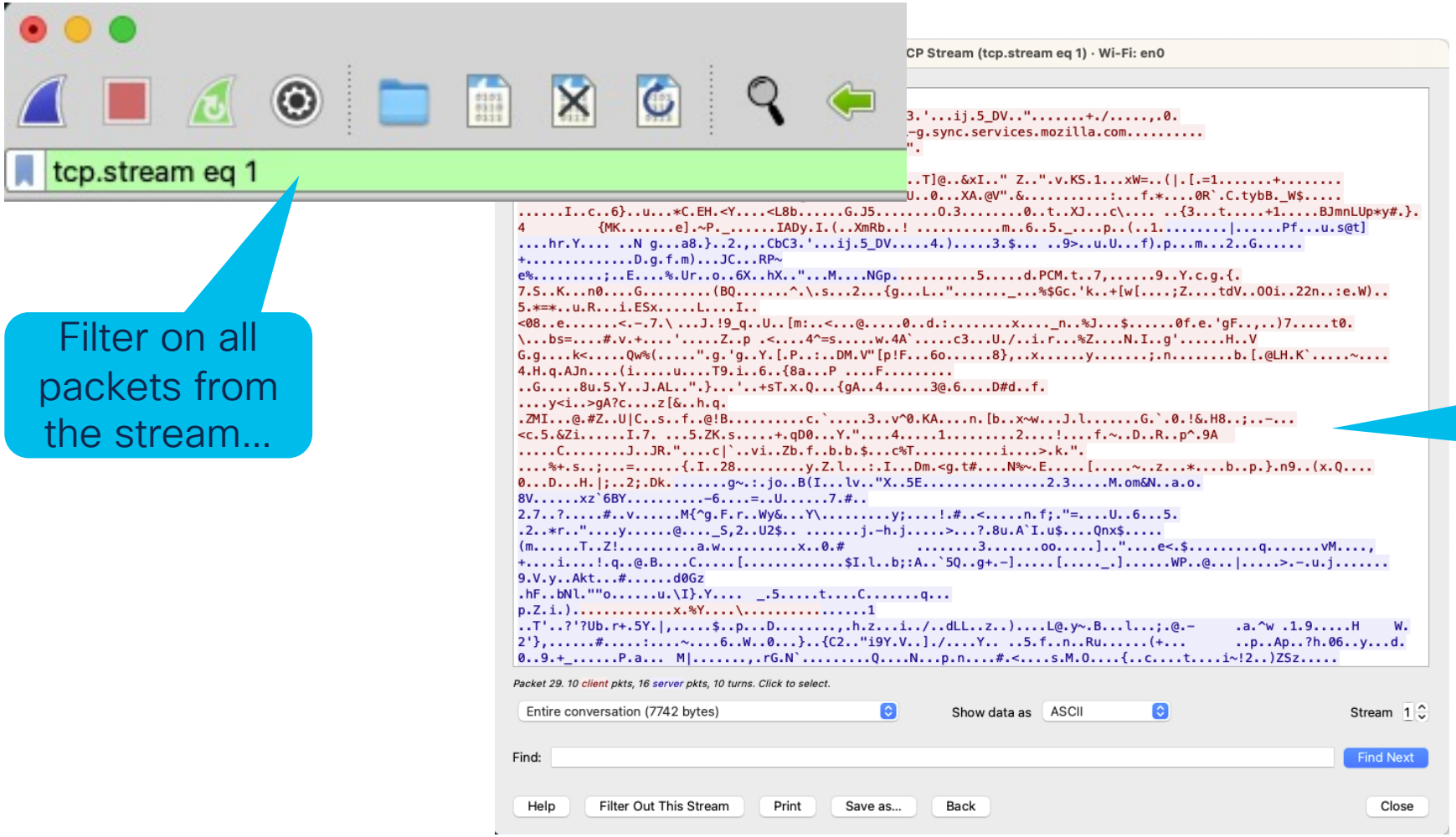
Following Streams



- TCP
- UDP
- DCCP
- TLS
- HTTP[/2]
- QUIC
- SIP

Pick the stream to follow based on initial packet

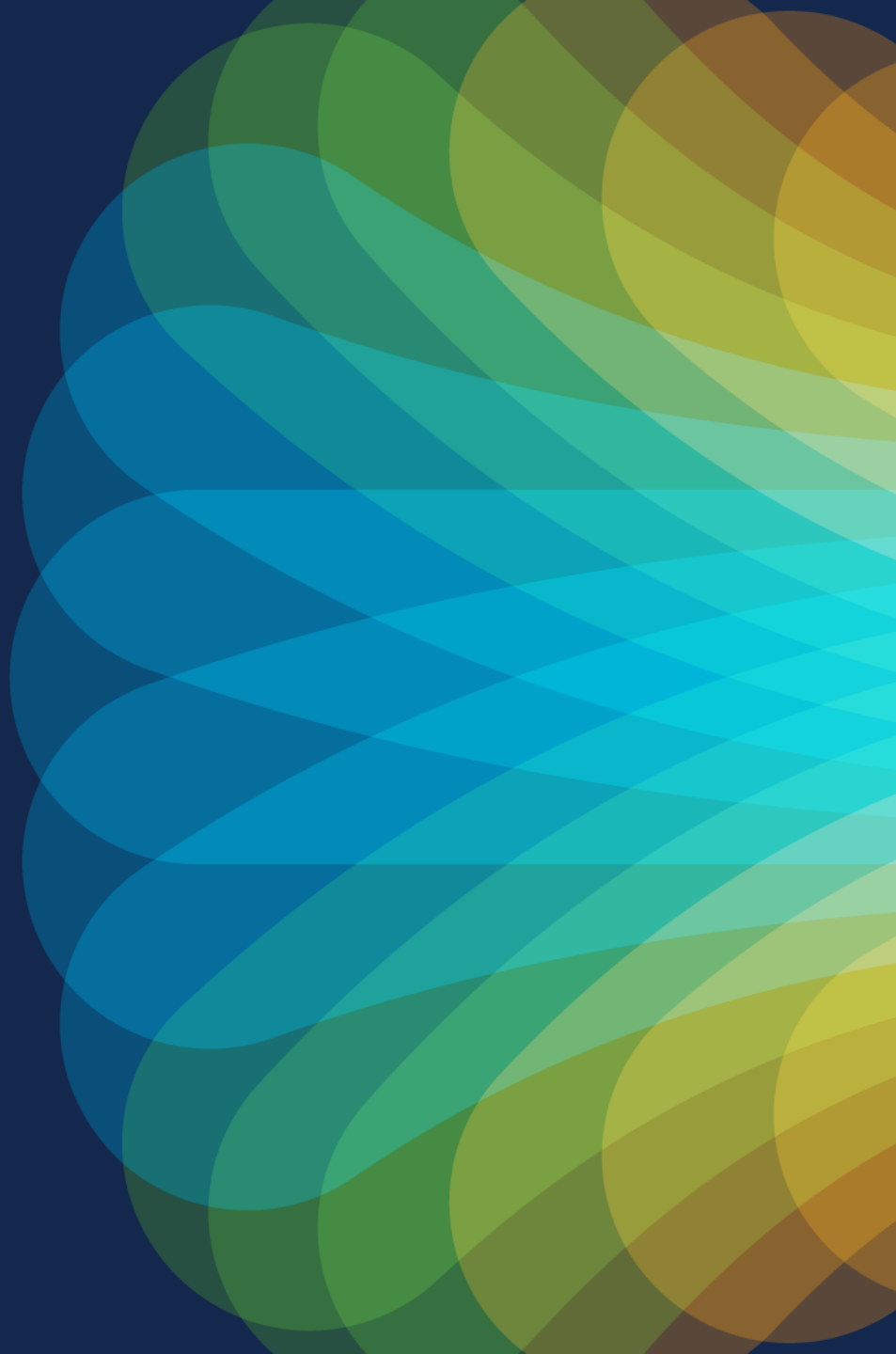
Following Streams



Filter on all packets from the stream...

...And assembles all data bytes in one screen

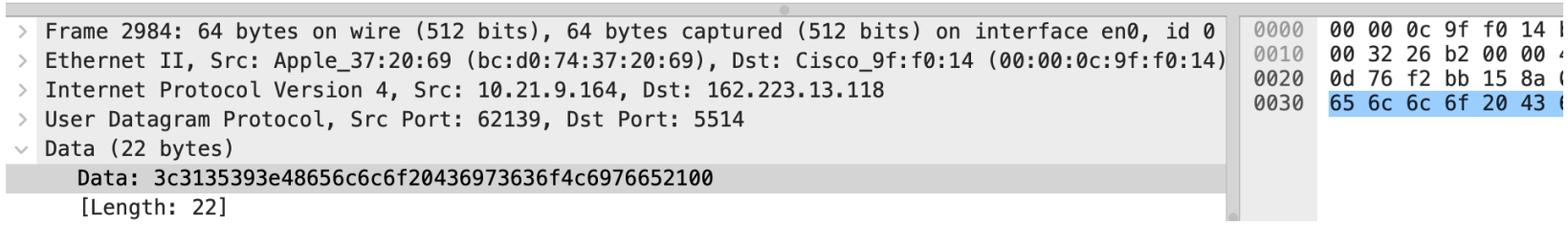
Decode As...



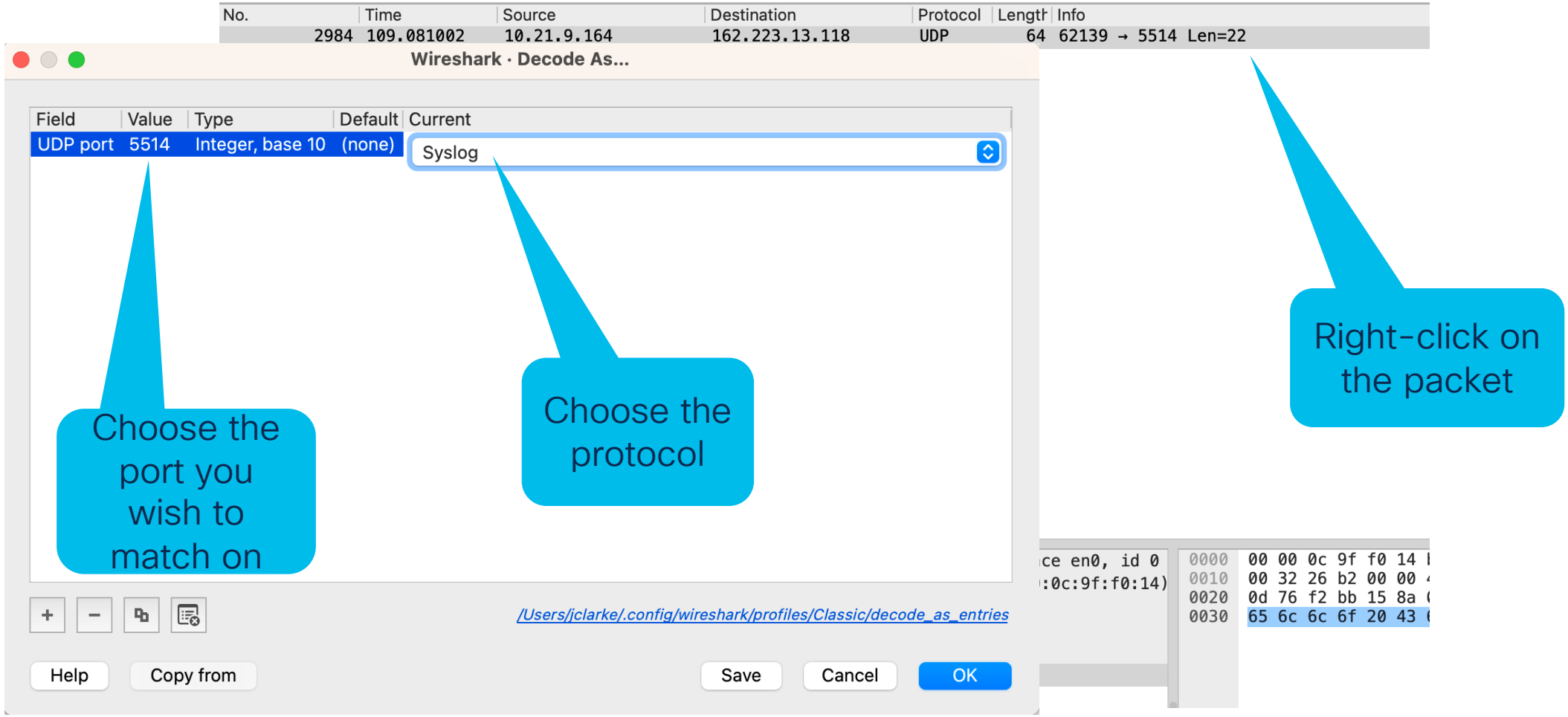
Decode As...

No.	Time	Source	Destination	Protocol	Length	Info
2984	109.081002	10.21.9.164	162.223.13.118	UDP	64	62139 → 5514 Len=22

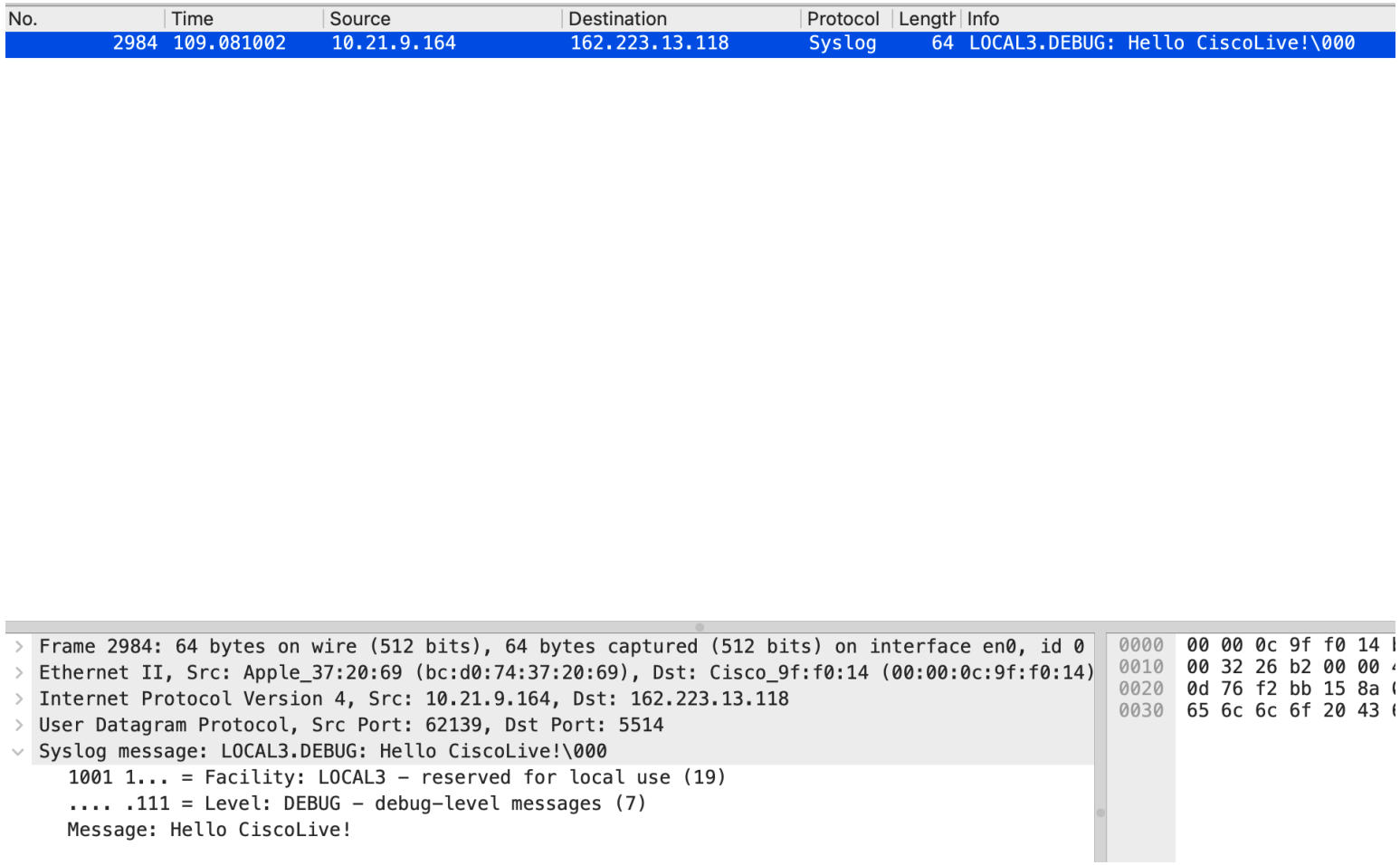
Let's say you have a data stream using non-standard ports. You still want to make use of wireshark's dissectors.



Decode As...



Decode As...

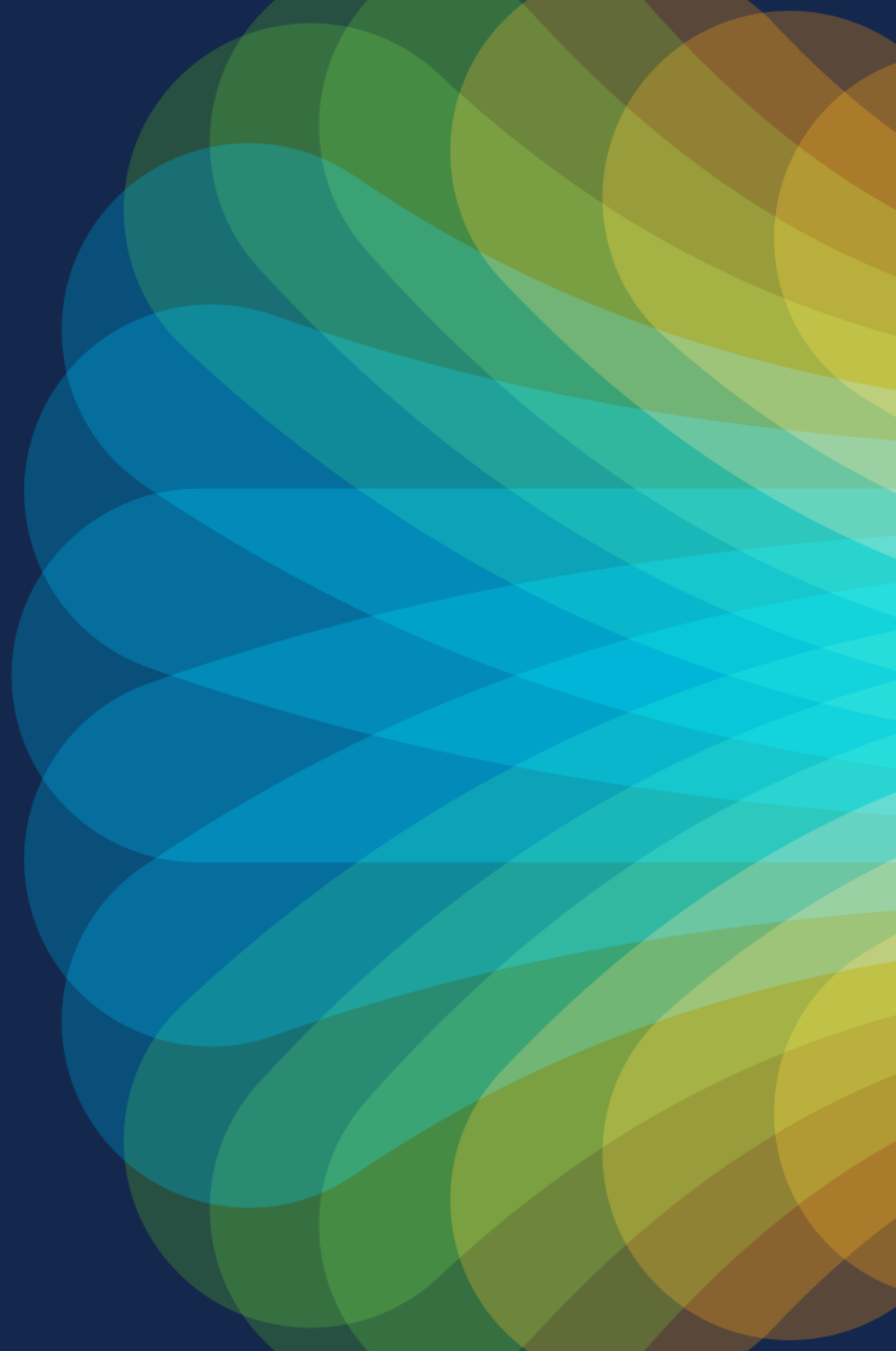


No.	Time	Source	Destination	Protocol	Length	Info
2984	109.081002	10.21.9.164	162.223.13.118	Syslog	64	LOCAL3.DEBUG: Hello CiscoLive!\000

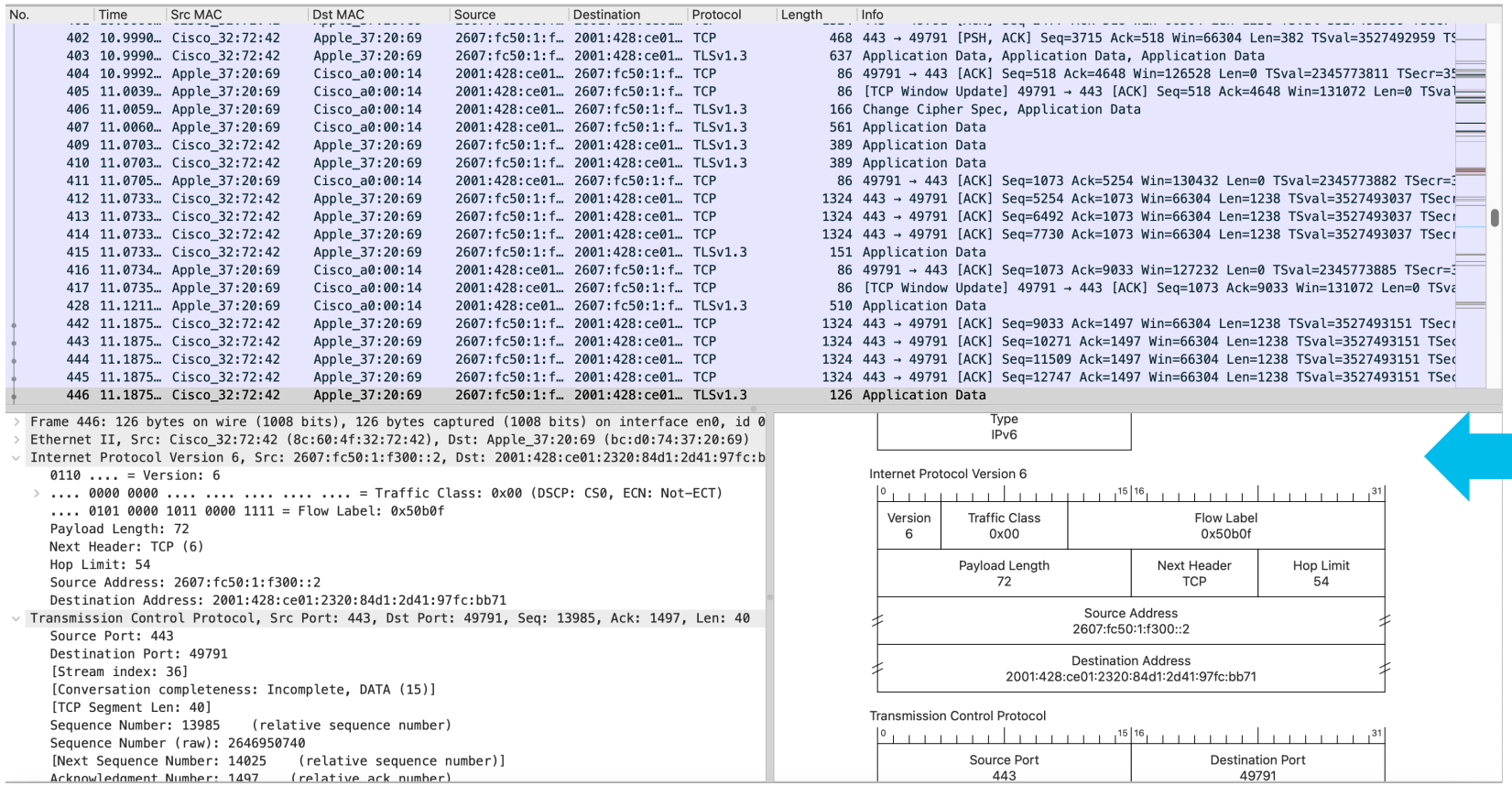
```
> Frame 2984: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface en0, id 0
> Ethernet II, Src: Apple_37:20:69 (bc:d0:74:37:20:69), Dst: Cisco_9f:f0:14 (00:00:0c:9f:f0:14)
> Internet Protocol Version 4, Src: 10.21.9.164, Dst: 162.223.13.118
> User Datagram Protocol, Src Port: 62139, Dst Port: 5514
v Syslog message: LOCAL3.DEBUG: Hello CiscoLive!\000
    1001 1... = Facility: LOCAL3 - reserved for local use (19)
    .... .111 = Level: DEBUG - debug-level messages (7)
    Message: Hello CiscoLive!
```

```
0000  00 00 0c 9f f0 14
0010  00 32 26 b2 00 00
0020  0d 76 f2 bb 15 8a
0030  65 6c 6c 6f 20 43
```

Packet Diagrams



Packet Diagrams



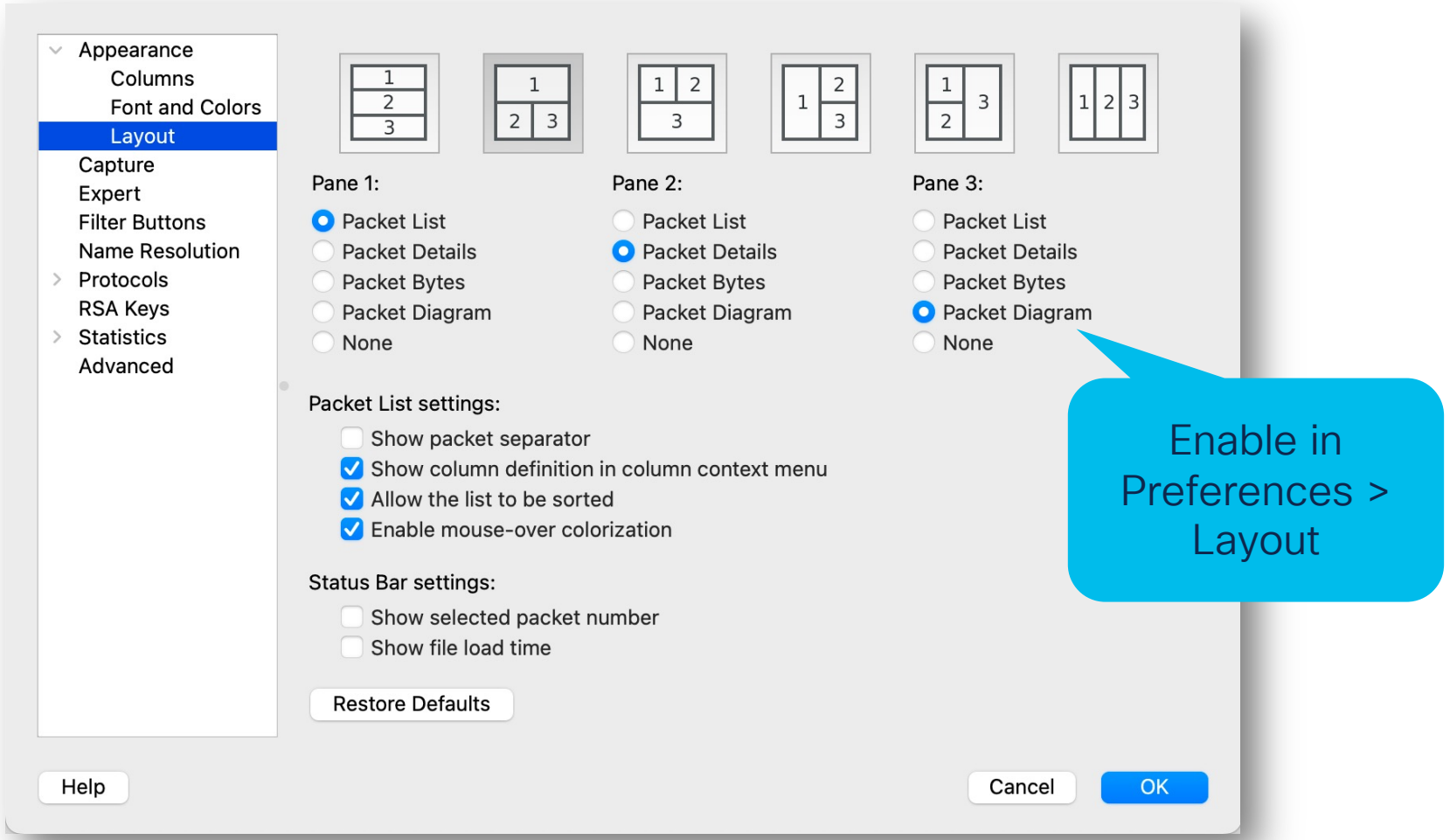
No.	Time	Src MAC	Dst MAC	Source	Destination	Protocol	Length	Info
402	10.9990…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TCP	468	443 → 49791 [PSH, ACK] Seq=3715 Ack=518 Win=66304 Len=382 TSval=3527492959 TS
403	10.9990…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TLSv1.3	637	Application Data, Application Data, Application Data
404	10.9992…	Apple_37:20:69	Cisco_a0:00:14	2001:428:ce01…	2607:fc50:1:f…	TCP	86	49791 → 443 [ACK] Seq=518 Ack=4648 Win=126528 Len=0 TSval=2345773811 TSecr=35
405	11.0039…	Apple_37:20:69	Cisco_a0:00:14	2001:428:ce01…	2607:fc50:1:f…	TCP	86	[TCP Window Update] 49791 → 443 [ACK] Seq=518 Ack=4648 Win=131072 Len=0 TSval
406	11.0059…	Apple_37:20:69	Cisco_a0:00:14	2001:428:ce01…	2607:fc50:1:f…	TLSv1.3	166	Change Cipher Spec, Application Data
407	11.0060…	Apple_37:20:69	Cisco_a0:00:14	2001:428:ce01…	2607:fc50:1:f…	TLSv1.3	561	Application Data
409	11.0703…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TLSv1.3	389	Application Data
410	11.0703…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TLSv1.3	389	Application Data
411	11.0705…	Apple_37:20:69	Cisco_a0:00:14	2001:428:ce01…	2607:fc50:1:f…	TCP	86	49791 → 443 [ACK] Seq=1073 Ack=5254 Win=130432 Len=0 TSval=2345773882 TSecr=3
412	11.0733…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TCP	1324	443 → 49791 [ACK] Seq=5254 Ack=1073 Win=66304 Len=1238 TSval=3527493037 TSecr
413	11.0733…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TCP	1324	443 → 49791 [ACK] Seq=6492 Ack=1073 Win=66304 Len=1238 TSval=3527493037 TSecr
414	11.0733…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TCP	1324	443 → 49791 [ACK] Seq=7730 Ack=1073 Win=66304 Len=1238 TSval=3527493037 TSecr
415	11.0733…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TLSv1.3	151	Application Data
416	11.0734…	Apple_37:20:69	Cisco_a0:00:14	2001:428:ce01…	2607:fc50:1:f…	TCP	86	49791 → 443 [ACK] Seq=1073 Ack=9033 Win=127232 Len=0 TSval=2345773885 TSecr=3
417	11.0735…	Apple_37:20:69	Cisco_a0:00:14	2001:428:ce01…	2607:fc50:1:f…	TCP	86	[TCP Window Update] 49791 → 443 [ACK] Seq=1073 Ack=9033 Win=131072 Len=0 TSva
428	11.1211…	Apple_37:20:69	Cisco_a0:00:14	2001:428:ce01…	2607:fc50:1:f…	TLSv1.3	510	Application Data
442	11.1875…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TCP	1324	443 → 49791 [ACK] Seq=9033 Ack=1497 Win=66304 Len=1238 TSval=3527493151 TSecr
443	11.1875…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TCP	1324	443 → 49791 [ACK] Seq=10271 Ack=1497 Win=66304 Len=1238 TSval=3527493151 TSec
444	11.1875…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TCP	1324	443 → 49791 [ACK] Seq=11509 Ack=1497 Win=66304 Len=1238 TSval=3527493151 TSec
445	11.1875…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TCP	1324	443 → 49791 [ACK] Seq=12747 Ack=1497 Win=66304 Len=1238 TSval=3527493151 TSec
446	11.1875…	Cisco_32:72:42	Apple_37:20:69	2607:fc50:1:f…	2001:428:ce01…	TLSv1.3	126	Application Data

```
Frame 446: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface en0, id 0
Ethernet II, Src: Cisco_32:72:42 (8c:60:4f:32:72:42), Dst: Apple_37:20:69 (bc:d0:74:37:20:69)
Internet Protocol Version 6, Src: 2607:fc50:1:f300::2, Dst: 2001:428:ce01:2320:84d1:2d41:97fc:b
    0110 .... = Version: 6
  > .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... 0101 0000 1011 0000 1111 = Flow Label: 0x50b0f
    Payload Length: 72
    Next Header: TCP (6)
    Hop Limit: 54
    Source Address: 2607:fc50:1:f300::2
    Destination Address: 2001:428:ce01:2320:84d1:2d41:97fc:bb71
Transmission Control Protocol, Src Port: 443, Dst Port: 49791, Seq: 13985, Ack: 1497, Len: 40
    Source Port: 443
    Destination Port: 49791
    [Stream index: 36]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 40]
    Sequence Number: 13985    (relative sequence number)
    Sequence Number (raw): 2646950740
    [Next Sequence Number: 14025    (relative sequence number)]
    Acknowledgment Number: 1497    (relative ack number)
```

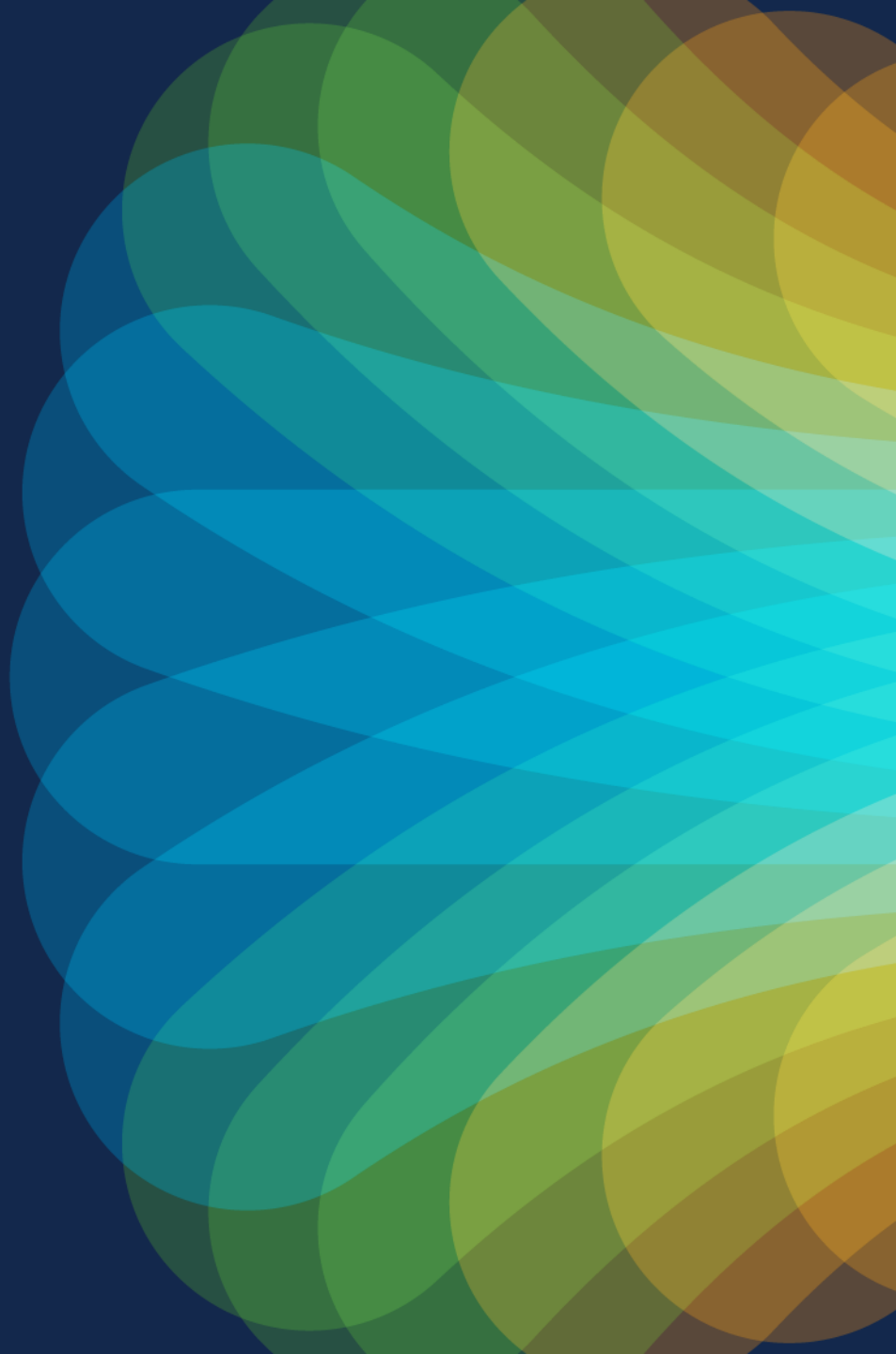

New to Wireshark 4.0

- Save or print them
- Copy as raster images
- Teach new engineers how a frame becomes a packet

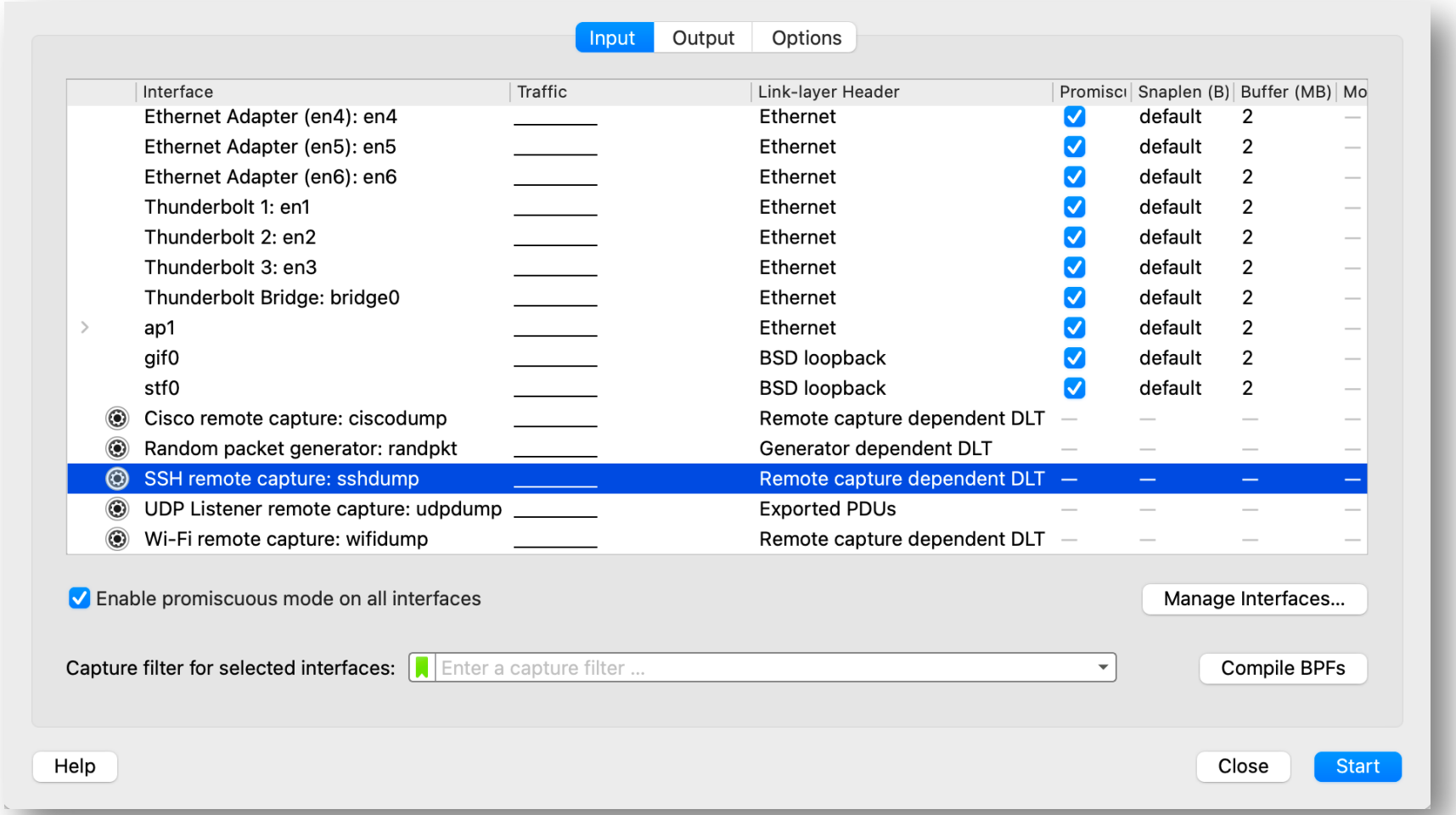
Packet Diagrams



Remote Capture

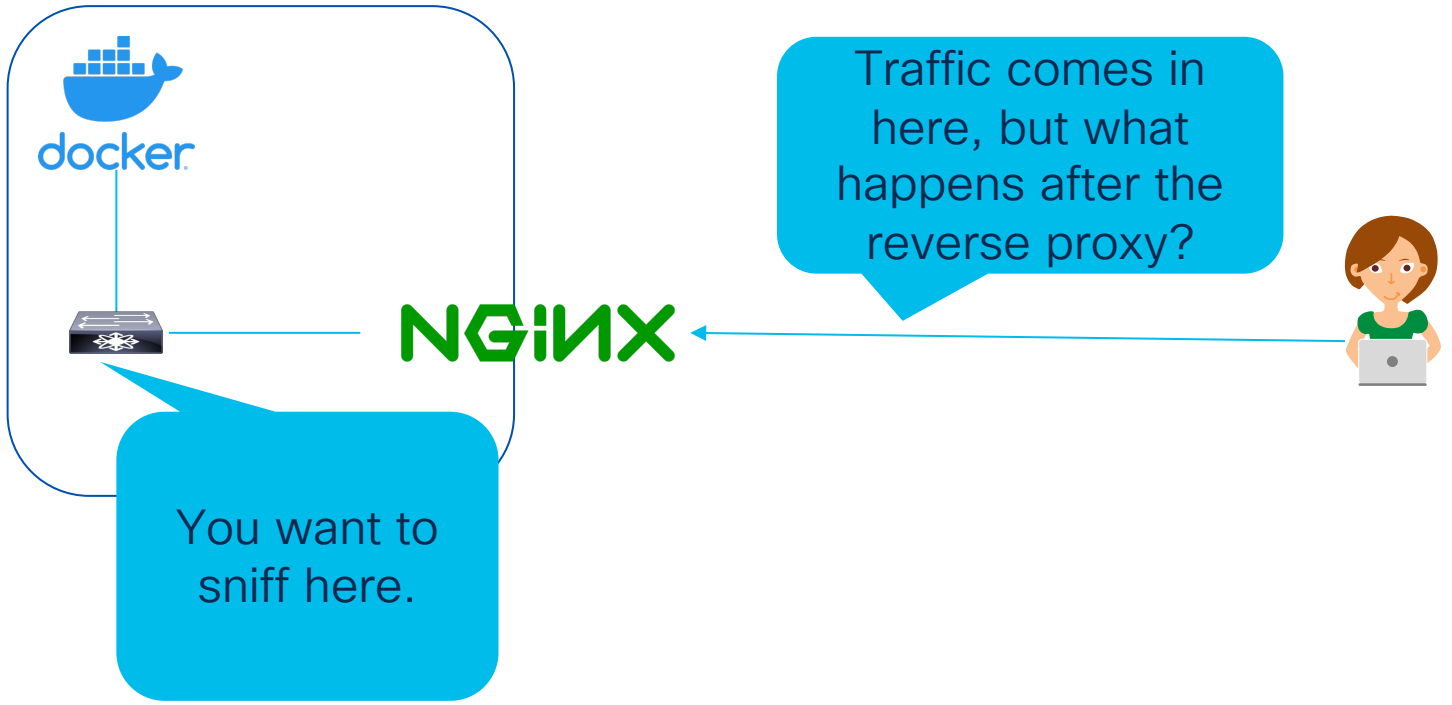


Remote Capture



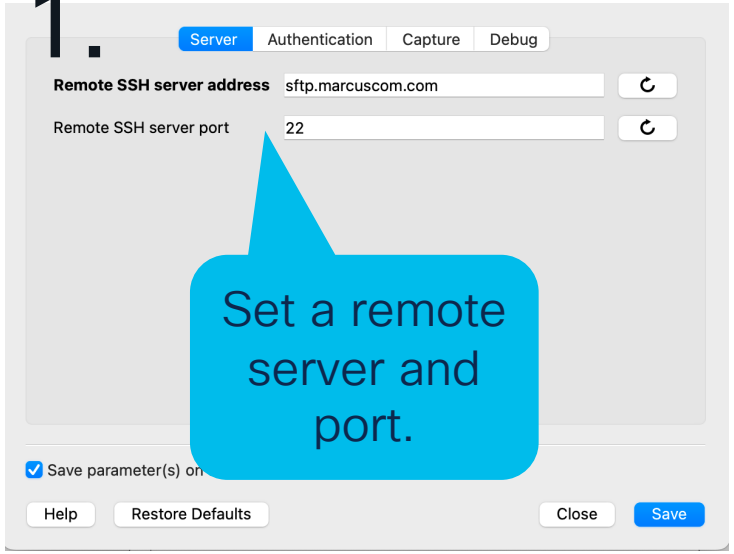
- Don’t forget to scroll down and explore the full interface list
- Remote captures are excellent for troubleshooting embedded services or setting up a SPAN server

Remote Capture



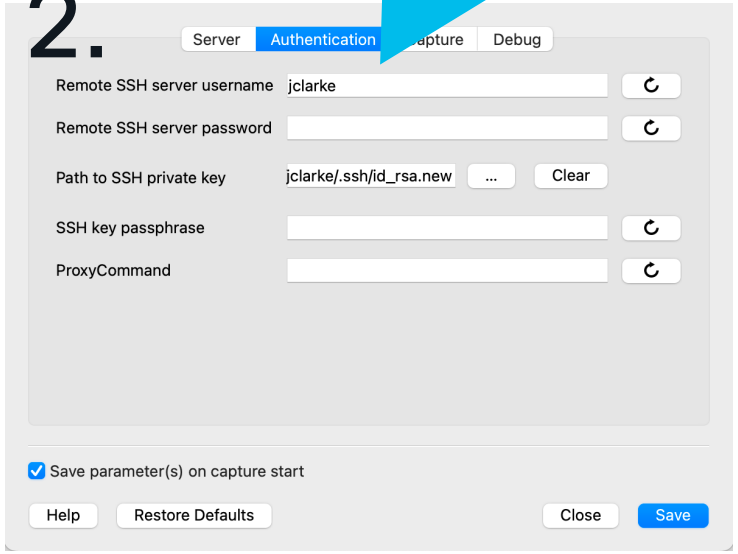
Remote Capture

1.



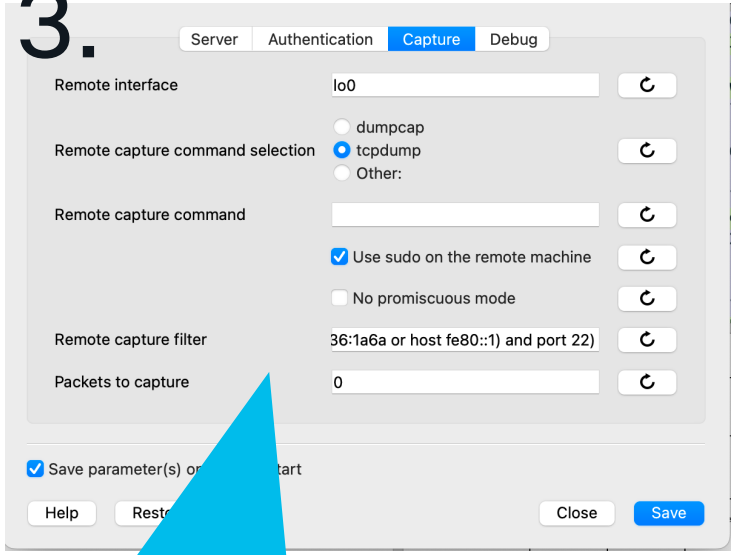
Set a remote server and port.

2.



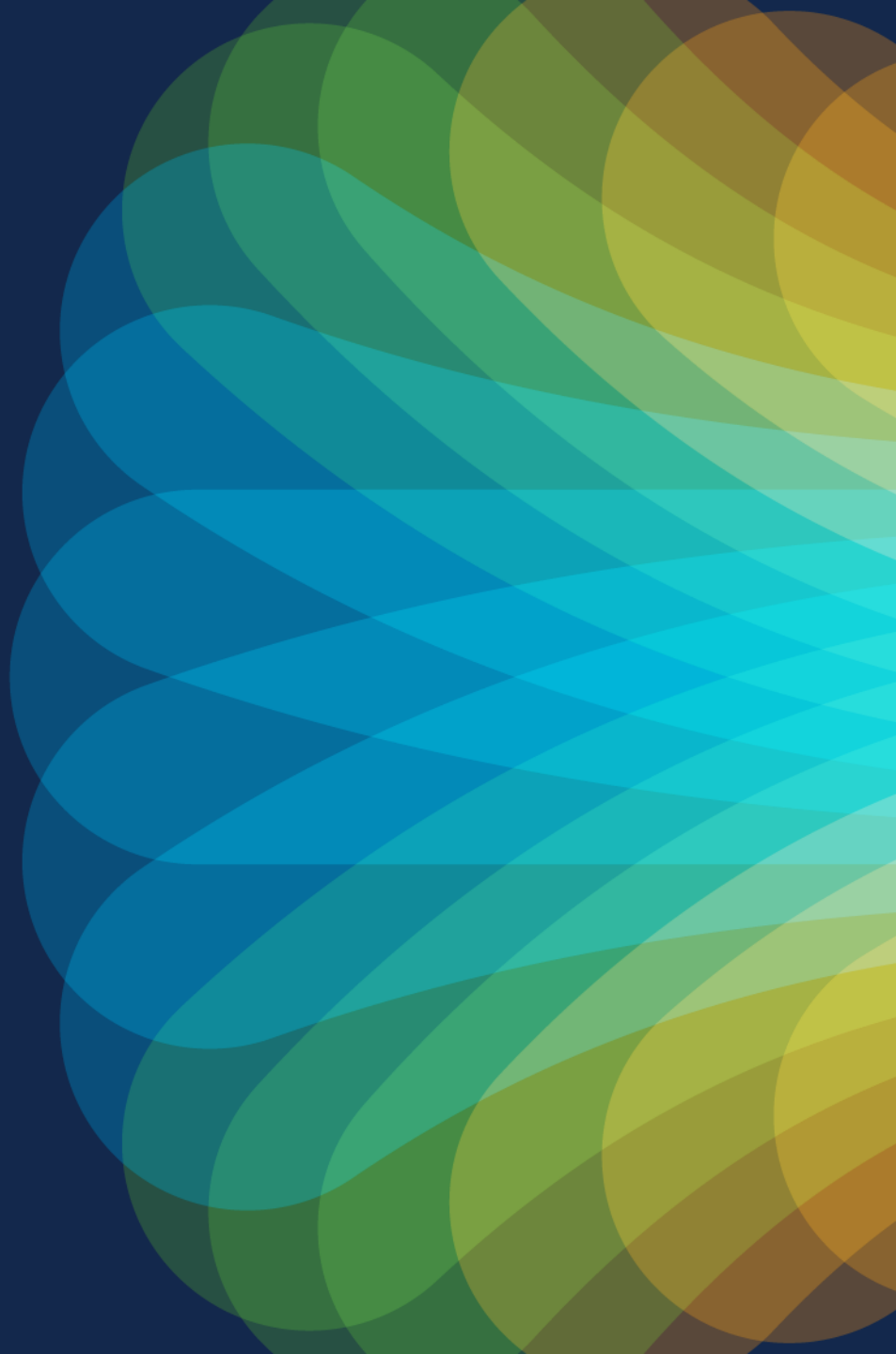
Specify authentication (I recommend SSH keys)

3.

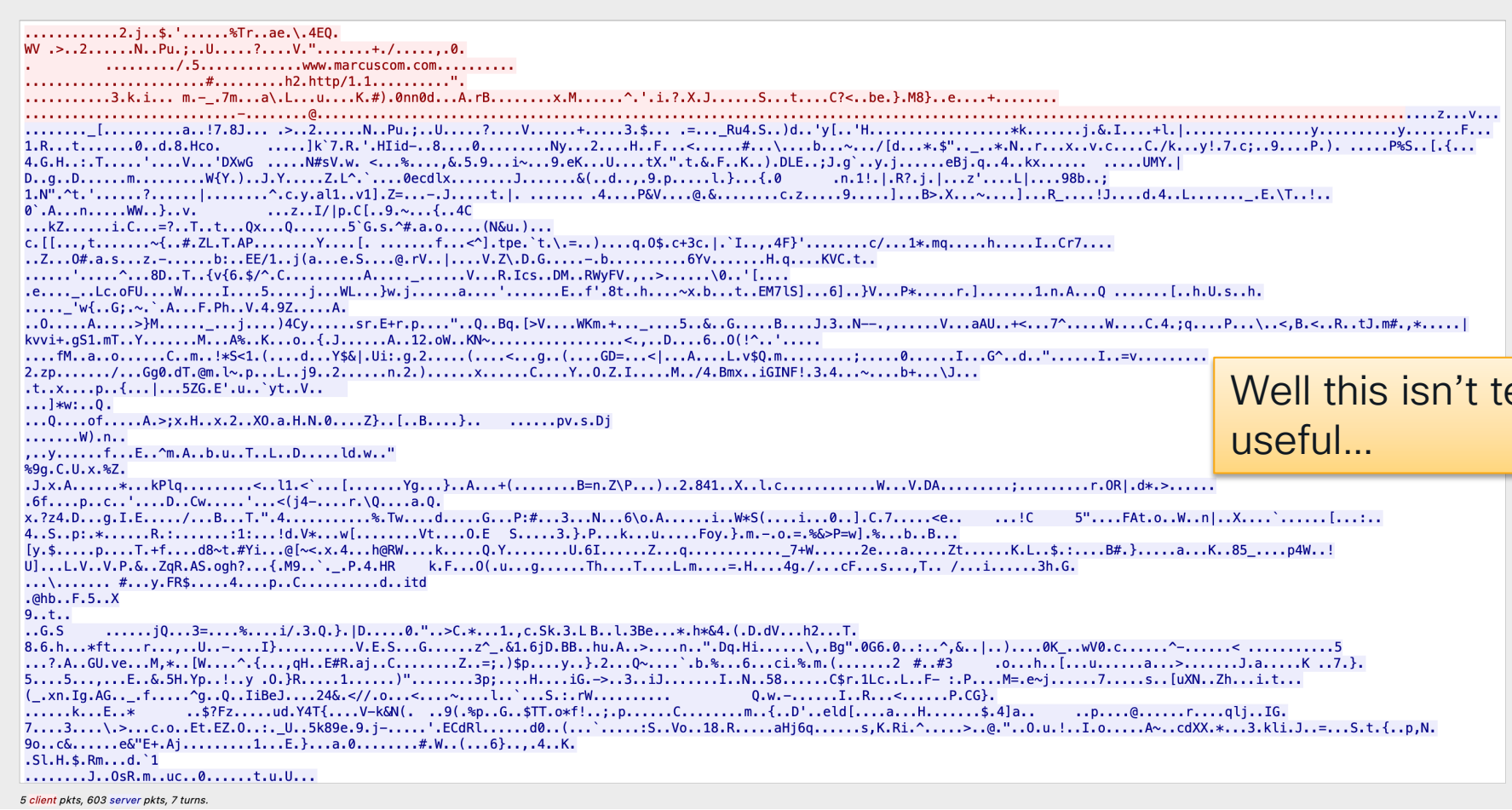


Select an interface. The initial capture filter is usually sufficient. Sudo is typically required.

Decrypt the Things!



Decrypt the Things!



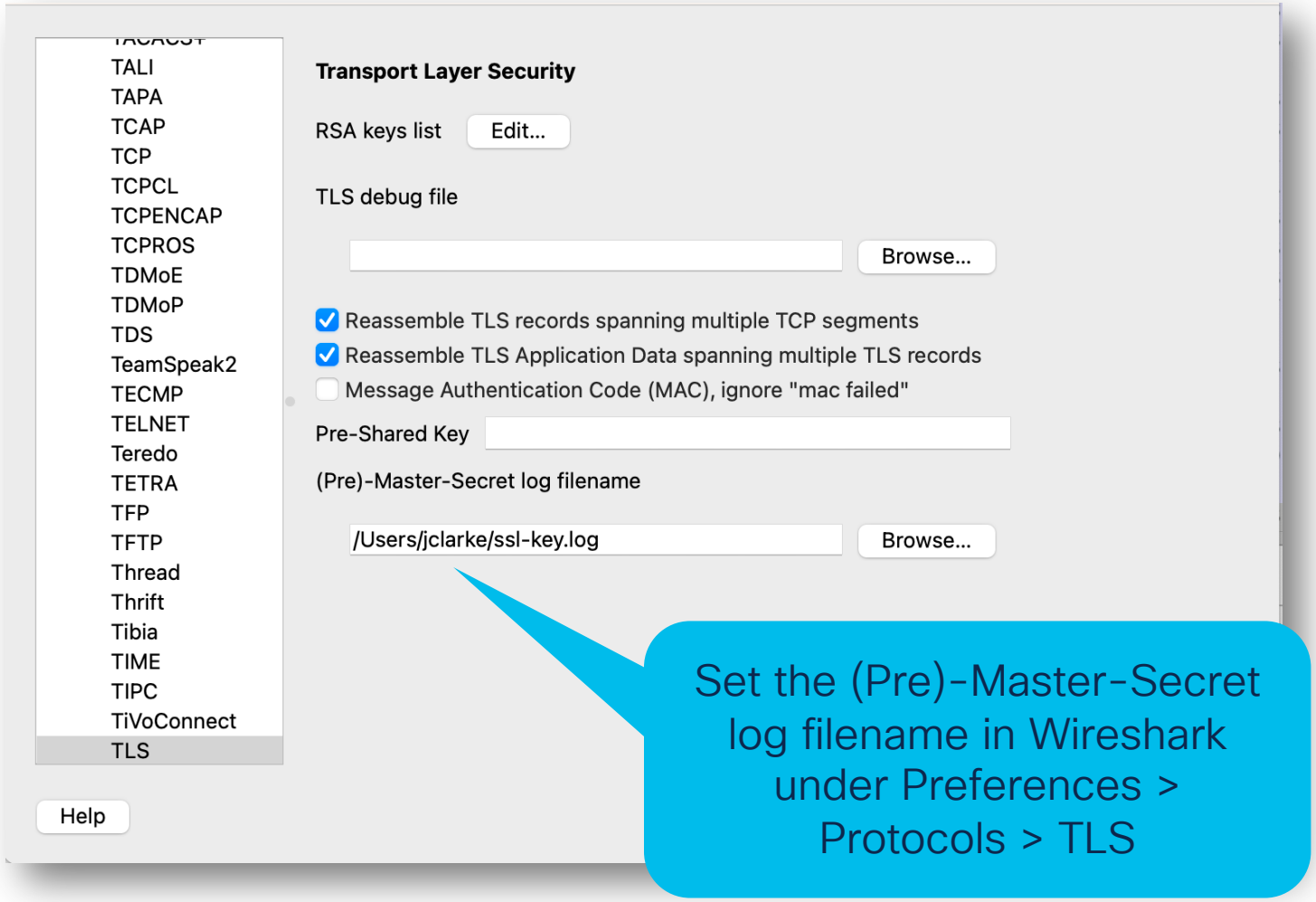
Well this isn't terribly useful...

Decrypt the Things!

- Set the `SSLKEYLOGFILE` environment variable
- Refresh your environment (e.g., source your `.bashrc`)
- Restart your browser

```
$ export SSLKEYLOGFILE=~/ssl-key.log
$ open /Applications/Firefox.app
```

Decrypt the Things!



Decrypt the Things!

```
GET /git HTTP/1.1
Host: www.marcuscom.com
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
sec-ch-ua: "Google Chrome";v="113", "Chromium";v="113", "Not-A.Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "macOS"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: lang=en-US; i_like_gogs=aae7ad4c5bb693b9; _csrf=el25xoAiIe009cwh6Es3ONg3WGU6MTY4NjA2ODcwODA4NTUxMTk3NQ

HTTP/1.1 200 OK
Date: Tue, 06 Jun 2023 16:25:32 GMT
Server: Apache/2.4.57 (FreeBSD) OpenSSL/1.1.1t PHP/8.1.19 SVN/1.14.2
Content-Type: text/html; charset=UTF-8
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked

<!DOCTYPE html>
<html>
<head data-suburl="/git">
          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
          <meta http-equiv="X-UA-Compatible" content="IE=edge"/>

                    <meta name="author" content="Gogs" />
                    <meta name="description" content="Gogs is a painless self-hosted Git service" />
                    <meta name="keywords" content="go, git, self-hosted, gogs">

          <meta name="referrer" content="no-referrer" />
          <meta name="_csrf" content="el25xoAiIe009cwh6Es3ONg3WGU6MTY4NjA2ODcwODA4NTUxMTk3NQ" />
          <meta name="_suburl" content="/git" />


                    <meta property="og:url" content="https://www.marcuscom.com/git/" />
                    <meta property="og:type" content="website" />
```

Profit!

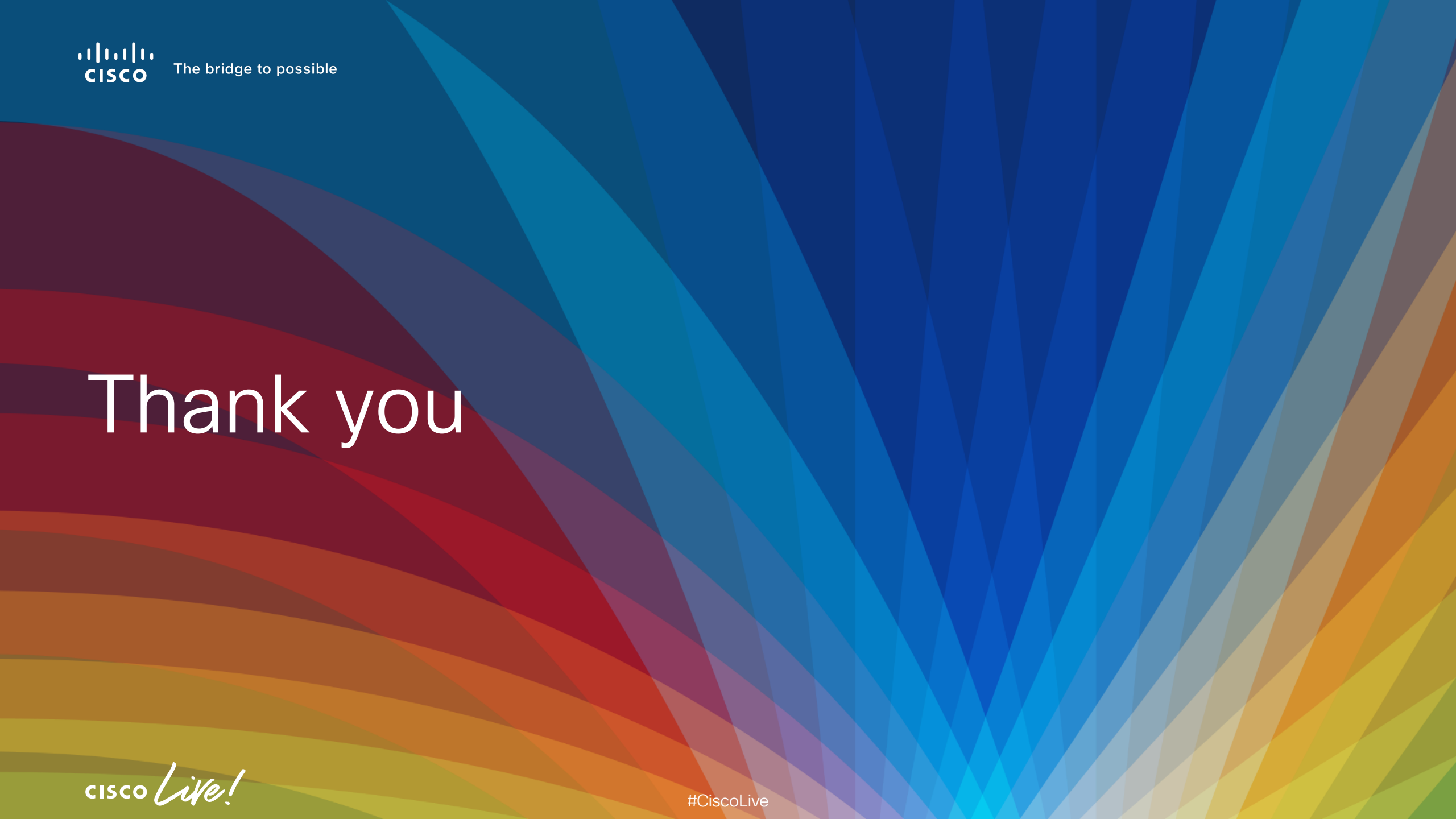
Packet 32. 3 client pkts, 3 server pkts, 5 turns. Click to select.

It’s Your Turn

- Get Wireshark (if you don’t have it)
- Make use of its powerful features to rule your network
- Keep flowin’

Get Wireshark

These Slides


cisco
The bridge to possible
Thank you
cisco Live!
#CiscoLive


cisco Live!
Let's go
#CiscoLive