Using LLMs in Networking

The case for understanding protocol "languages"

November 2024

IETF-121 Side Meeting on LLMs for Networking

Jari Arkko, **Ericsson Research**, Finland

Context and goals

Lots of excitement on generative AI for

- Human languages, chat bots
- Image and video creation
- Programming assistance
- Search and documents

Cool, but not at the heart of things from a protocol or network engineer perspective

And excitement on Gen AI for networking

- Network design
- Intents, configuration, etc.
- Identify patterns or anomalies
- Incident management

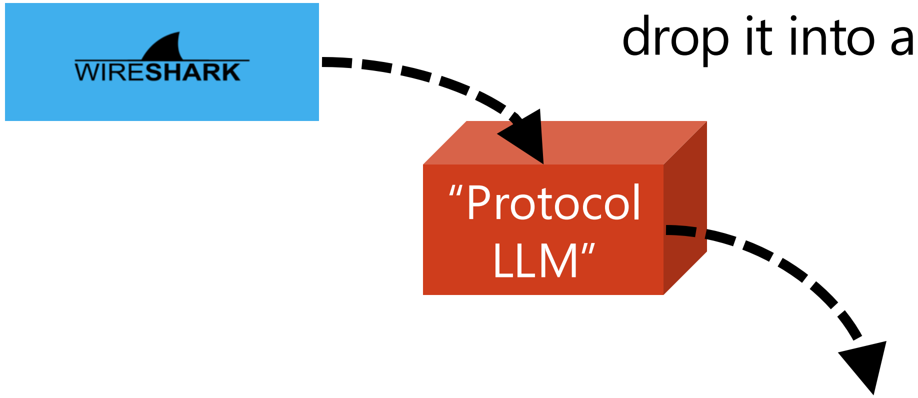
Cool and more at the heart of things, but is this an exhaustive list?

Context and goals

What if LLMs were able to also understand and converse natively in protocol messages?

- There’s multi-modal generative AI and support for multiple languages
- Could we “speak” protocols, too?

“Take a PCAP file and drop it into an LLM"



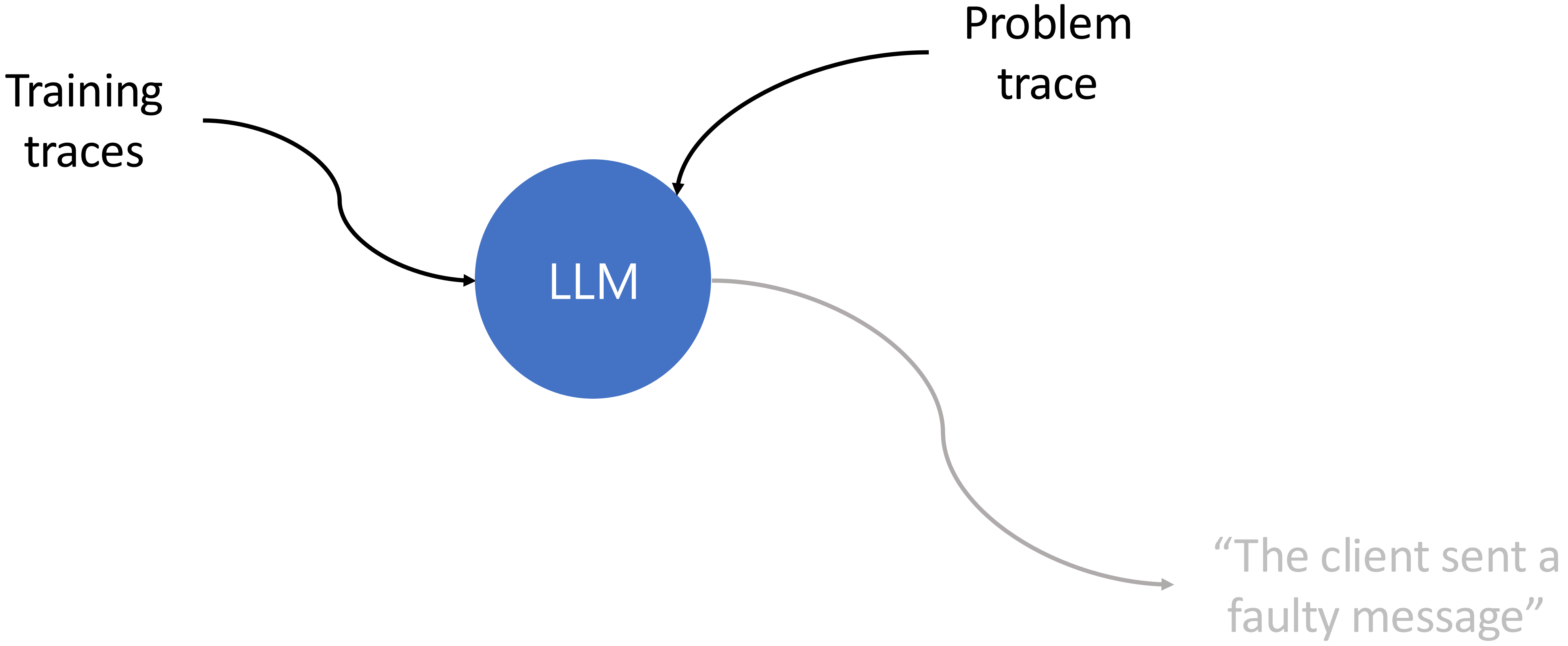
Could we make something interesting happen?

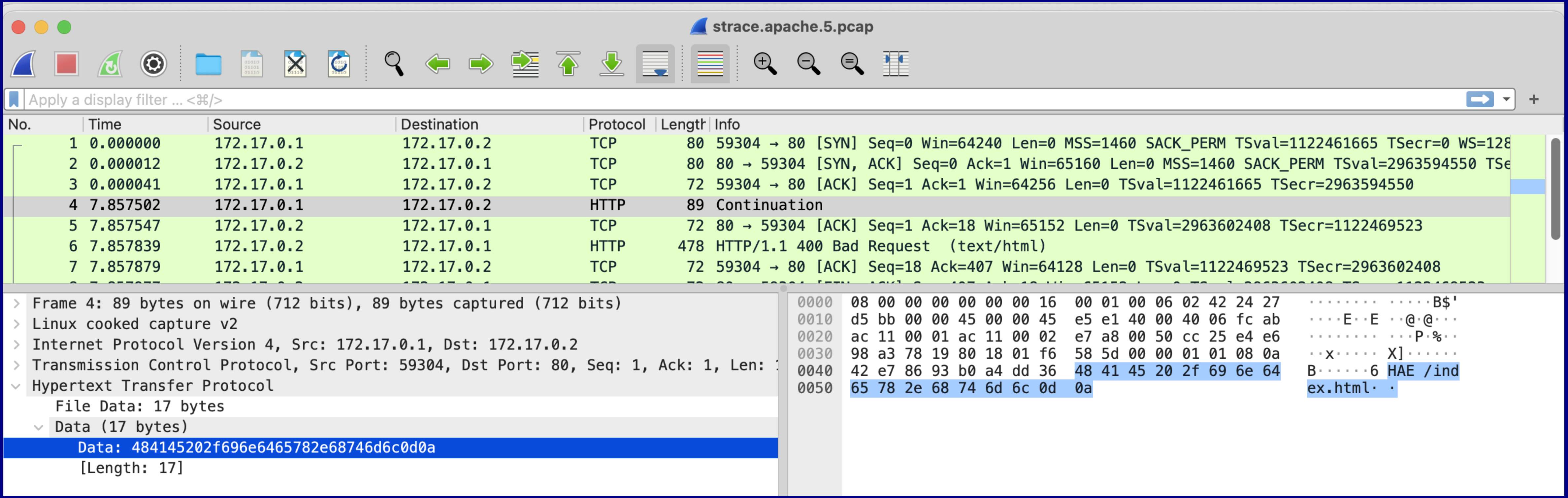
- Explain what is going on
- Generate test data
- Quick prototyping / simulation

Example Use Case:

Diagnostics

Use Case Context: Training traces & Problem traces





```
No.   Time       Source       Destination  Protocol Length Info
    1 0.000000   172.17.0.1   172.17.0.2   TCP       80 59304 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1122461665 TSecr=0 WS=128
    2 0.000012   172.17.0.2   172.17.0.1   TCP       80 80 → 59304 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2963594550 TSe
    3 0.000041   172.17.0.1   172.17.0.2   TCP       72 59304 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1122461665 TSecr=2963594550
    4 7.857502   172.17.0.1   172.17.0.2   HTTP      89 Continuation
    5 7.857547   172.17.0.2   172.17.0.1   TCP       72 80 → 59304 [ACK] Seq=1 Ack=18 Win=65152 Len=0 TSval=2963602408 TSecr=1122469523
    6 7.857839   172.17.0.2   172.17.0.1   HTTP     478 HTTP/1.1 400 Bad Request  (text/html)
    7 7.857879   172.17.0.1   172.17.0.2   TCP       72 59304 → 80 [ACK] Seq=18 Ack=407 Win=64128 Len=0 TSval=1122469523 TSecr=2963602408
```

```
> Frame 4: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)
> Linux cooked capture v2
> Internet Protocol Version 4, Src: 172.17.0.1, Dst: 172.17.0.2
> Transmission Control Protocol, Src Port: 59304, Dst Port: 80, Seq: 1, Ack: 1, Len: 1
v Hypertext Transfer Protocol
    File Data: 17 bytes
  v Data (17 bytes)
      Data: 484145202f696e6465782e68746d6c0d0a
      [Length: 17]
```

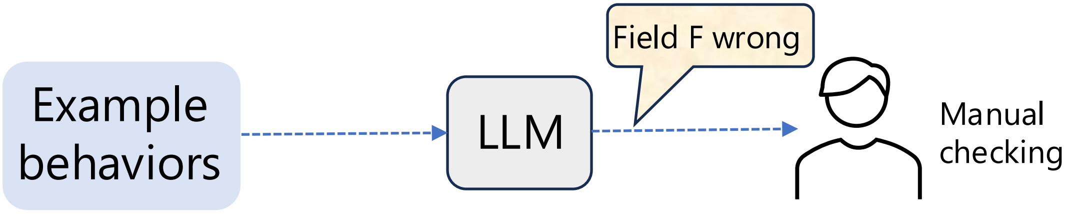
```
0000  08 00 00 00 00 00 00 16  00 01 00 06 02 42 24 27   ········ ·····B$'
0010  d5 bb 00 00 45 00 00 45  e5 e1 40 00 40 06 fc ab   ····E··E ··@·@···
0020  ac 11 00 01 ac 11 00 02  e7 a8 00 50 cc 25 e4 e6   ········ ···P·%··
0030  98 a3 78 19 80 18 01 f6  58 5d 00 00 01 01 08 0a   ··x····· X]······
0040  42 e7 86 93 b0 a4 dd 36  48 41 45 20 2f 69 6e 64   B······6 HAE /ind
0050  65 78 2e 68 74 6d 6c 0d  0a                        ex.html· ·
```

AI:

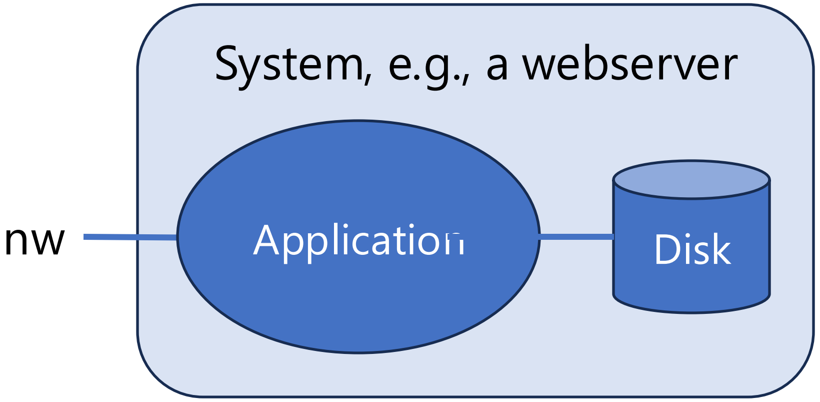
“Due to the unrecognized or invalid HTTP method ("HAE"), the server responds with a "400 Bad Request" status code.”

Some Challenges

- **Quantifying the quality of the results** – how well would this work in practice?
- **Complex fields** – length, checksum, encryption, …)
- **Protocols are not everything** – real system behavior is not explained by protocols only
- **Can we rely on this?** – correctness



As an AI, I'm unable to perform real-time calculations or generate dynamic content such as calculating a UDP checksum for a specific packet.



But what does this mean?

We’ve found this exciting

Further work needed

But are there broader implications?

There seems to many levels at which generative AI can be applied, even in networking

Please consider what patterns, languages, inputs your problem needs