SIPSA - one step closer to real anonymity on the Internet

Source IP spoofing for anonymization over UDP

Kirils Solovjovs, BalCCon2k16

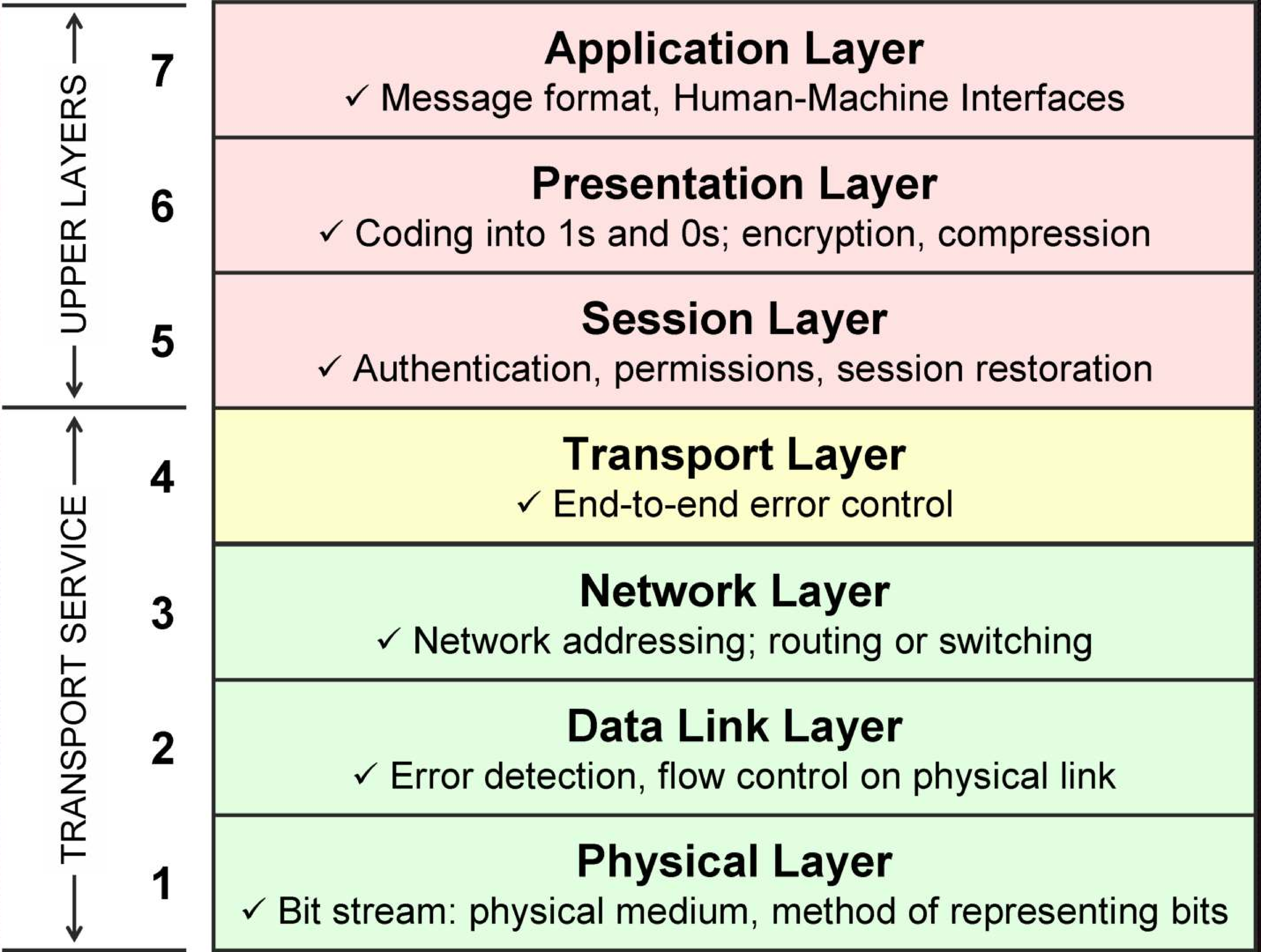
Who is this guy?

- IT security expert; researcher at 1st Ltd, Latvia
- Skills: network flow analysis, reverse engineering, social engineering, penetration testing, security incident investigation, and the legal dimension of cyber security and cyber defence
- The responsible disclosure guy
- Still an inventor at heart

What is this talk? Why does it exist?

- To introduce SIPSA to the world,
- thus encouraging discussion
  - on real-life applications of the technology
  - and improvements to it,
- and hopefully getting merge/pull requests.

Intro crash-course: networking



UPPER LAYERS

7 **Application Layer**

✓ Message format, Human-Machine Interfaces

6 **Presentation Layer**

✓ Coding into 1s and 0s; encryption, compression

5 **Session Layer**

✓ Authentication, permissions, session restoration

TRANSPORT SERVICE

4 **Transport Layer**

✓ End-to-end error control

3 **Network Layer**

✓ Network addressing; routing or switching

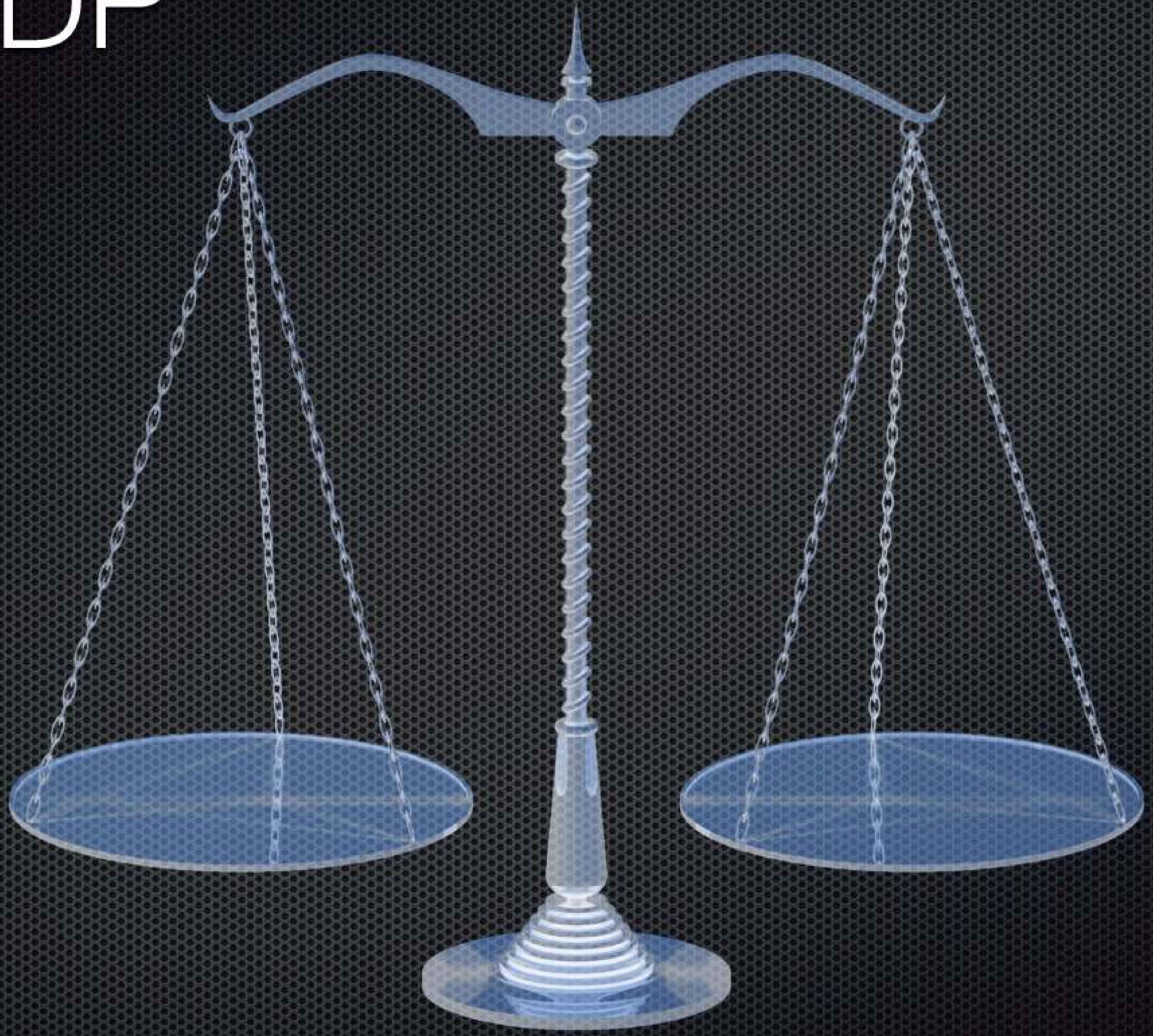
2 **Data Link Layer**

✓ Error detection, flow control on physical link

1 **Physical Layer**

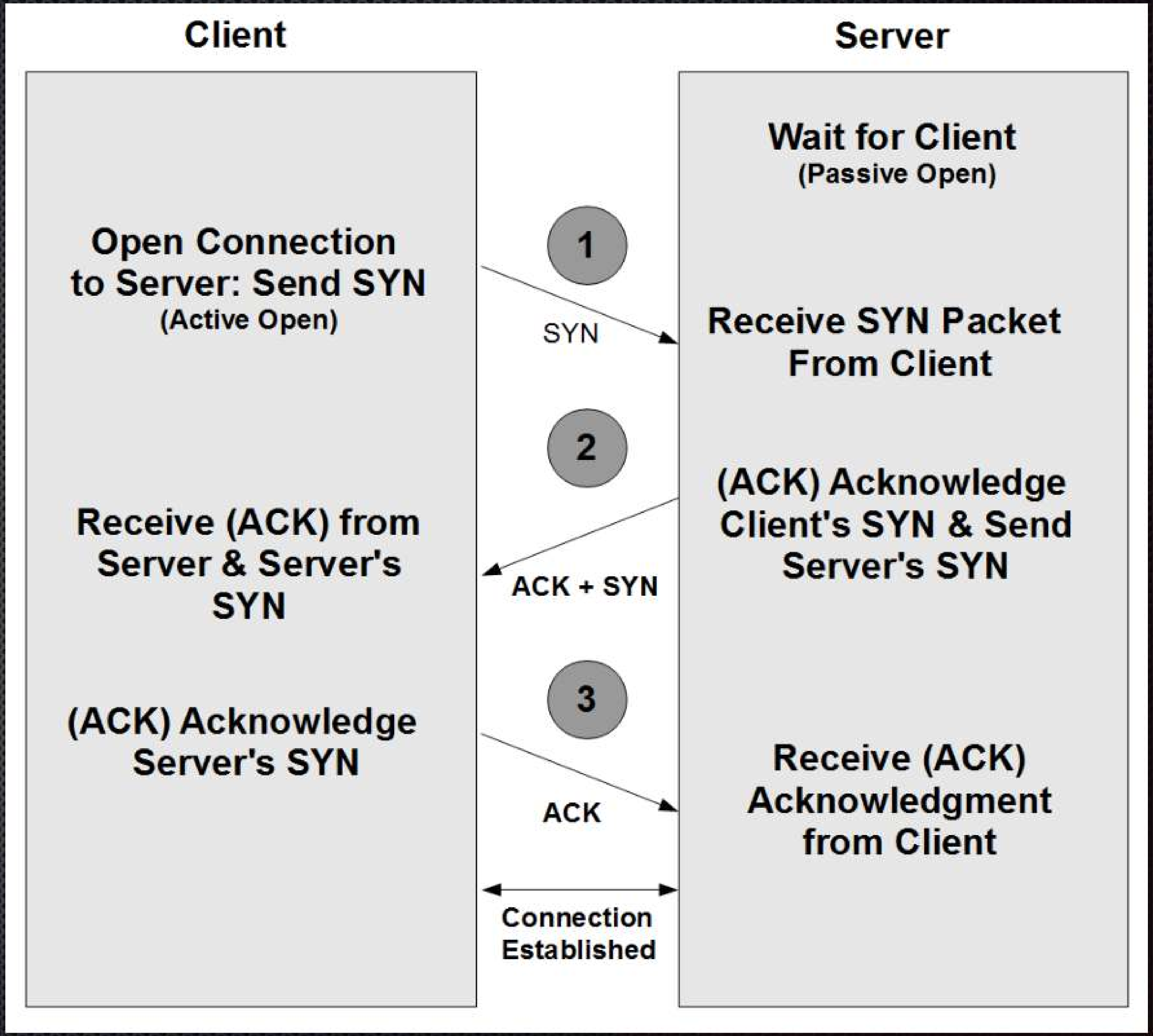
✓ Bit stream: physical medium, method of representing bits

TCP vs UDP



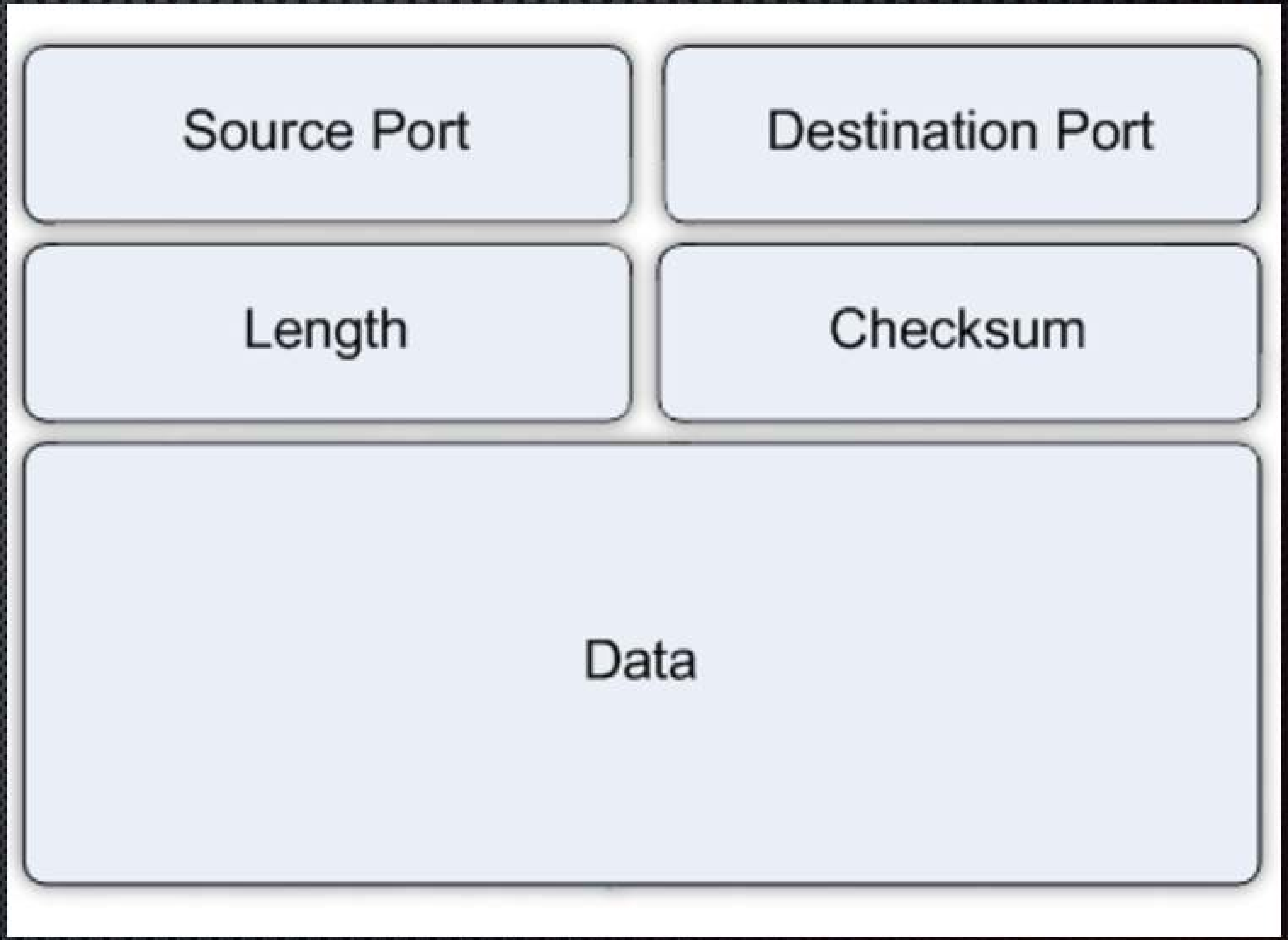
TCP features

- Stateful, connection-oriented
- "Reliable" transport
- Notable features include:
  - 3-way handshake
  - Error detection
  - Ordered transfer
  - Flow control


Client
Server
Wait for Client
(Passive Open)
Open Connection to Server: Send SYN
(Active Open)
1
SYN
Receive SYN Packet From Client
2
(ACK) Acknowledge Client's SYN & Send Server's SYN
ACK + SYN
Receive (ACK) from Server & Server's SYN
3
(ACK) Acknowledge Server's SYN
ACK
Receive (ACK) Acknowledgment from Client
Connection Established

UDP features

- Stateless, transaction-oriented
- "Best effort" transport
- Notable features include:
  - Minimalist design
  - No control
  - No retransmissions



Source Port

Destination Port

Length

Checksum

Data

P. Steiner
"On the Internet, nobody knows you're a dog."

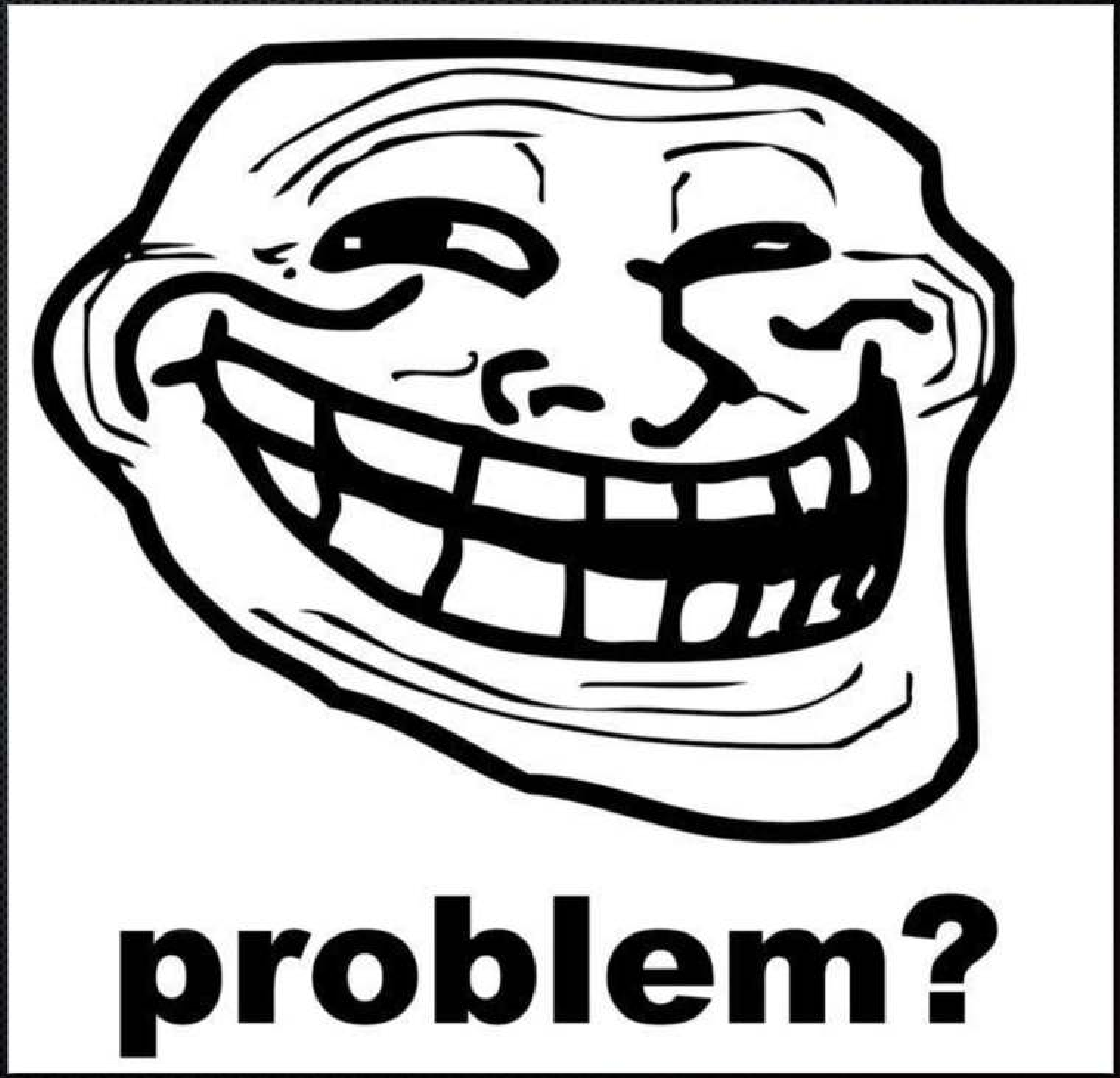
OUR NEXT GUEST HAS CHOSEN TO REMAIN ANONYMOUS, SO HE WILL REMAIN BEHIND THIS SCREEN DURING THE INTERVIEW.
PLEASE WELCOME MR. ISAAC PHILLIPS, AGE 26, FROM AMARILLO, TEXAS!
7
ASHER FREEMAN

Anonymity on the Internet... ?

Network 1 (10.0.0.0)
10.0.0.1
10.0.0.2
PC-PT
PC0
2960-24TT
Switch0
10.0.0.3
Laptop-PT
Laptop0
2620XM
Router1
40.0.0.2
20.0.0.1
40.0.0.1
2620XM
Router2
30.0.0.1
2960-24TT
Switch1
20.0.0.2
PC-PT
PC1
20.0.0.3
PC-PT
PC2
Network 2 (20.0.0.0)
Network 3 (30.0.0.0)
30.0.0.3
PC-PT
PC3
2960-24TT
Switch2
30.0.0.2
Laptop-PT
Laptop1

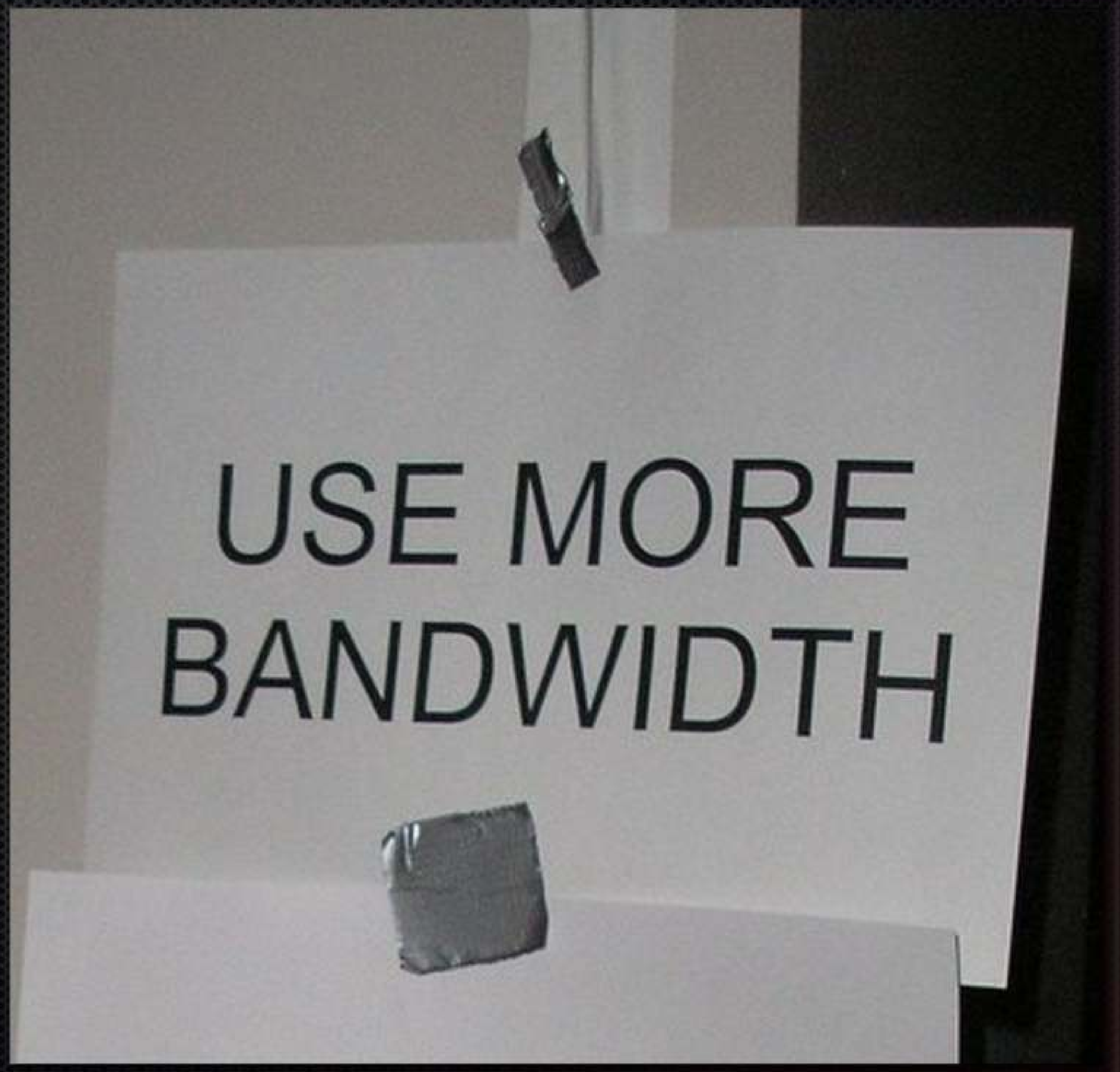
Problem?

- IPs on Layer 3 needed for routing
- Cannot remove or encrypt them
- Yes, problem!



A statistical solution!

- Global bandwidth is increasing by an order of magnitude every 5 years!
- We need MORE DATAGRAMS
- Yes, a statistical solution!



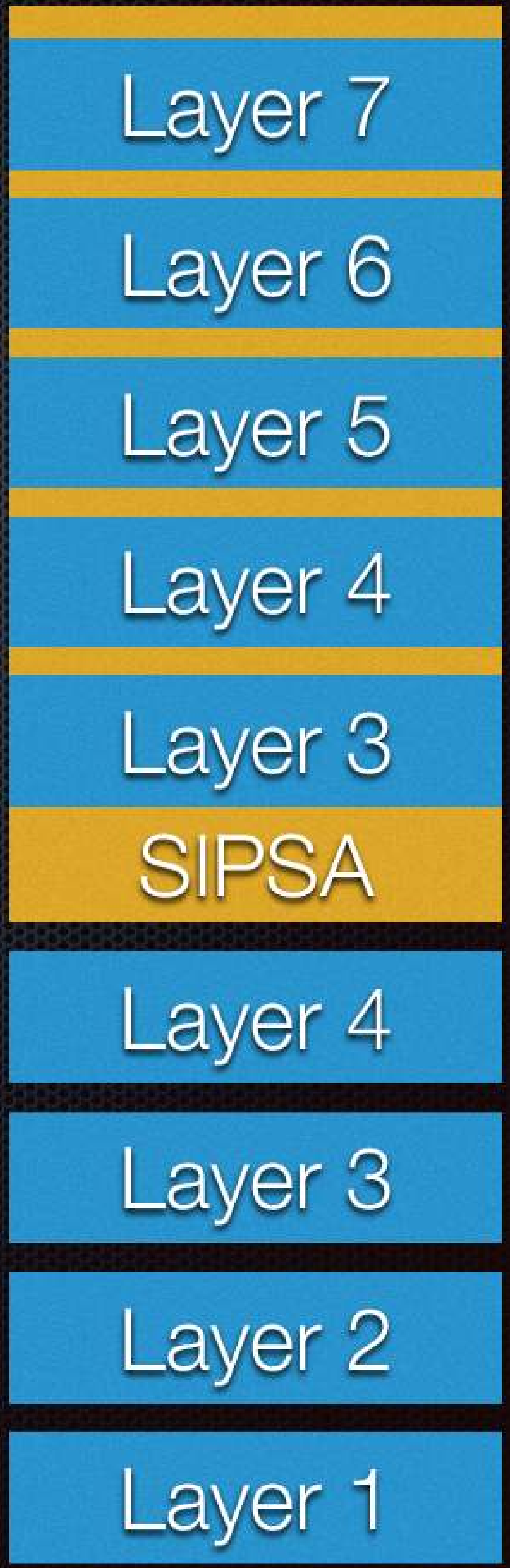
No, it does not come with a logo. It's not a vulnerability, ffs.

SIPSA

A dream come true?

SIPSA overview

- Protocol goes on top of Layer 4, but below Layer 3 [!]
- Instead of sending a single UDP datagram, many are sent
  - Different pairs of (randomised) source and destination IPs
- Protocol allows for the expansion / version support
- Current version (04) chooses IPs in pairs within a class C network
- Metadata currently includes (encrypted) real IPs and a list of the fakes
- Payload is not encrypted

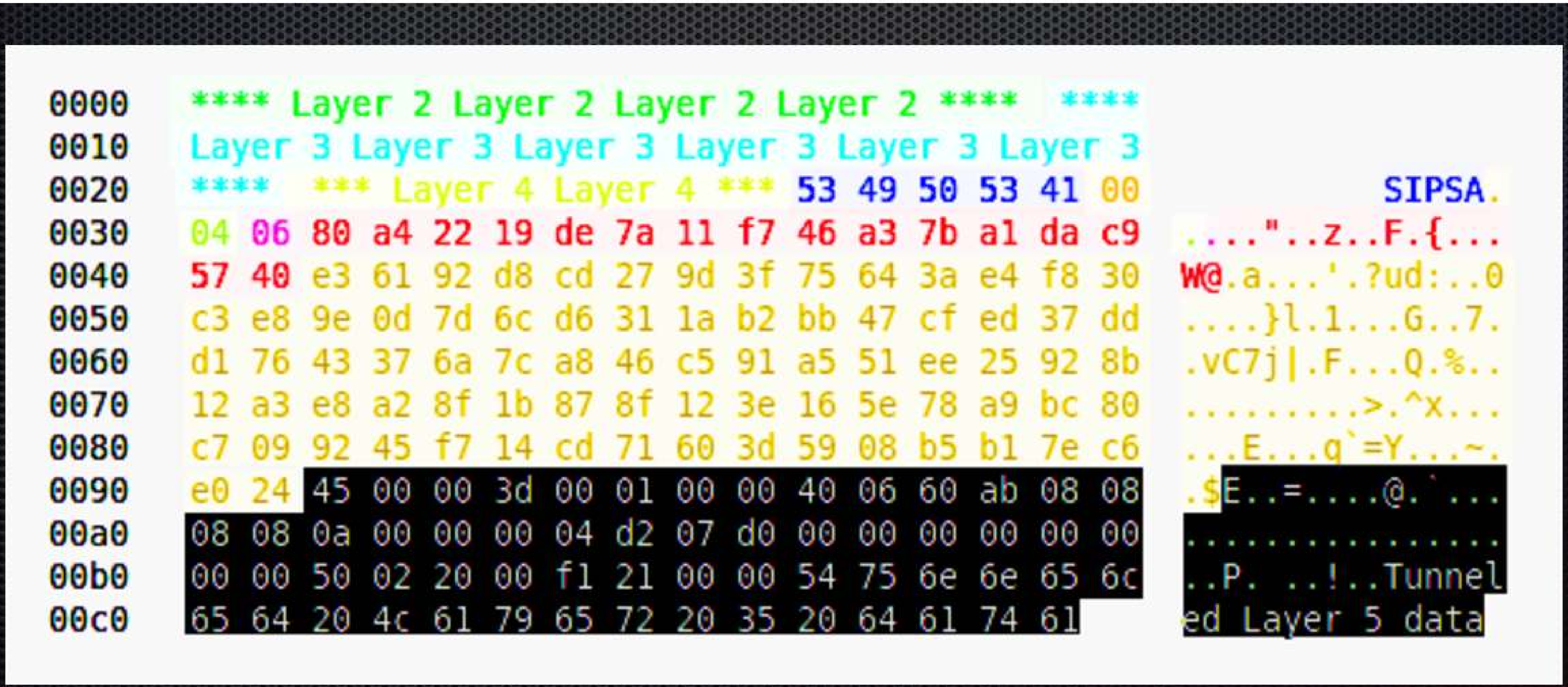


No.	Time	Source	Destination	Protocol	Length	Info
16	0.525493000	73.110.16.23	194.232.119.61	UDP	207	Source port: 51654 Destination port: 51654
17	0.561435000	59.46.124.156	85.254.196.147	UDP	207	Source port: 51654 Destination port: 51654
18	0.606041000	59.46.124.156	53.60.44.232	UDP	207	Source port: 51654 Destination port: 51654
19	0.657318000	59.46.124.156	53.60.44.38	UDP	207	Source port: 51654 Destination port: 51654
20	0.701079000	59.46.124.156	73.22.109.27	UDP	207	Source port: 51654 Destination port: 51654
21	0.725030000	59.46.124.156	194.232.119.26	UDP	207	Source port: 51654 Destination port: 51654
22	0.757072000	59.46.124.156	73.22.109.28	UDP	207	Source port: 51654 Destination port: 51654
23	0.789475000	59.46.124.156	85.254.196.140	UDP	207	Source port: 51654 Destination port: 51654
24	0.833965000	59.46.124.156	194.232.119.61	UDP	207	Source port: 51654 Destination port: 51654
25	0.873479000	5.179.8.176	85.254.196.147	UDP	207	Source port: 51654 Destination port: 51654
26	0.913170000	5.179.8.176	53.60.44.232	UDP	207	Source port: 51654 Destination port: 51654
27	0.949429000	5.179.8.176	53.60.44.38	UDP	207	Source port: 51654 Destination port: 51654
28	0.981160000	5.179.8.176	73.22.109.27	UDP	207	Source port: 51654 Destination port: 51654
29	1.009337000	5.179.8.176	194.232.119.26	UDP	207	Source port: 51654 Destination port: 51654
30	1.041917000	5.179.8.176	73.22.109.28	UDP	207	Source port: 51654 Destination port: 51654
31	1.073366000	5.179.8.176	85.254.196.140	UDP	207	Source port: 51654 Destination port: 51654
32	1.097322000	5.179.8.176	194.232.119.61	UDP	207	Source port: 51654 Destination port: 51654
33	1.141451000	59.46.124.235	85.254.196.147	UDP	207	Source port: 51654 Destination port: 51654
34	1.181288000	59.46.124.235	53.60.44.232	UDP	207	Source port: 51654 Destination port: 51654

Thus SIPSA should provide anonymity and deniability

SIPSA datagram format

					ENCRYPTED with AES256, CBC mode, 16B block, iv=IV, total size = (Metalen-1)*16B							
Header 5B “SIPSA”	Reserved 1B “\00”	Proto ver 1B “\04”	Metalen 1B	IV 16B	Real src IP 4B may be zeros	Real dst IP 4B may be zeros	Src IP list 4B x n n≥0	End marker 1B “\xFF”	Dst IP list 4B x n n≥0	End marker 1B “\xFF”	Padding 0B – 15B “\x00”	Payload 0B +



```
0000   **** Layer 2 Layer 2 Layer 2 Layer 2 ****  ****
0010   Layer 3 Layer 3 Layer 3 Layer 3 Layer 3 Layer 3
0020   ****  *** Layer 4 Layer 4 *** 53 49 50 53 41 00             SIPSA.
0030   04 06 80 a4 22 19 de 7a 11 f7 46 a3 7b a1 da c9   ...."..z..F.{...
0040   57 40 e3 61 92 d8 cd 27 9d 3f 75 64 3a e4 f8 30   W@.a...'.?ud:..0
0050   c3 e8 9e 0d 7d 6c d6 31 1a b2 bb 47 cf ed 37 dd   ....}l.1...G..7.
0060   d1 76 43 37 6a 7c a8 46 c5 91 a5 51 ee 25 92 8b   .vC7j|.F...Q.%..
0070   12 a3 e8 a2 8f 1b 87 8f 12 3e 16 5e 78 a9 bc 80   .........>.^x...
0080   c7 09 92 45 f7 14 cd 71 60 3d 59 08 b5 b1 7e c6   ...E...q`=Y...~.
0090   e0 24 45 00 00 3d 00 01 00 00 40 06 60 ab 08 08   .$E..=....@.`...
00a0   08 08 0a 00 00 00 04 d2 07 d0 00 00 00 00 00 00   ................
00b0   00 00 50 02 20 00 f1 21 00 00 54 75 6e 6e 65 6c   ..P. ..!..Tunnel
00c0   65 64 20 4c 61 79 65 72 20 35 20 64 61 74 61      ed Layer 5 data
```

BCP38

- Best Current Practice, May 2000 [!]
- Network Ingress Filtering
- Drops packets having unknown source prefix
- Supposed to solve DoS
- Worked well, but did not solve DoS in the long term (today)

The good

- BCP38 has been sparsely implemented
- SIPSA may provide an additional layer of anonymity as part of a larger suite
- SIPSA provides deniability by virtue of UDP (and having fixed port numbering)

> “No, your honour. My devices neither requested nor acknowledged receipt of the communication in question.”
>
> *–You, on SIPSA*

The bad

- BCP38 is not going away; it's being slowly deployed on additional networks
- SIPSA gives only statistical improvement not 100% anonymity, so statistical attacks are likely possible
- Success largely depends on the ISPs involved
- Network load increase
  - 3x3 addresses 8x
  - 2x5 addresses 9x
  - 6x6 addresses 35x

And the demo

Future testing and research

- Anonymity
- General security
- Other ideas

Anonymity

- Consider not including real source IP in the metadata
  - Even the server has no way of knowing or logging client IPs
- Consider not sending packet from the real source at all
  - It's of course impossible to do both

General security

- Check validity of the crypto
- Key management
- Try attacking to find weak spots
- Obfuscate the protocol

Other ideas

- IPv6 support
- Include the random seed instead of the IP map in the metadata
- Stateful SIPSA
  - = a bit less bandwidth usage
- NAT? :(

1. All feedback is welcome
2. Please fork and send merge/pull requests!
3. https://GitHub.com/0ki/SIPSA
4. I am @KirilsSolovjovs
5. Thank you for joining me and have a great con!

* Presentation features fair-use of images found on the web