

Computer Organization and Networks

Chapter 8: Networking III

Winter 2022/2023



Jakob Heher, www.iaik.tugraz.at
he/his



The Application Layer

(some select applications)

Distance	Transversal Velocity	Velocity	Name	
21.5 AU			Okkamon	Stargate (Ca)
20.5 AU			Ikoskio	Stargate (Ca)
19.2 AU			Mushikegi	Stargate (Ca)
19.2 AU			Vaaralen	Stargate (Ca)
12 km	263 m/s	36 m/s	Lady Karin	Revelation [SMA]
18 km	279 m/s	34 m/s	Radspike Dlg	Revelation [GENTS]
23 km	221 m/s	56 m/s	Ergingblood	Revelation [FCOM]
17 km	289 m/s	26 m/s	Bel Shazar	Revelation [CONDI]
13 km	254 m/s	38 m/s	Agnes Marta	Revelation [CONDI]
20 km	277 m/s	17 m/s	Ntrails	Revelation [CONDI]
20 km	252 m/s	21 m/s	Jonney-Jran	Revelation [CONDI]
22 km	210 m/s	38 m/s	Lord Thorian	Revelation [CONDI]
21 km	285 m/s	23 m/s	Cplonel Shep	Revelation [CONDI]
16 km	293 m/s	32 m/s	CyberRaver	Revelation [CONDI]
28 km	256 m/s	3 m/s	Pelleas IC	Revelation [-RZR-]
12 km	270 m/s	40 m/s	Harry Obama	Revelation [-FA-]
24 km	235 m/s	15 m/s	Kes Lynnair	Revelation [-FA-]
23 km	239 m/s	29 m/s	Justicas Heir	Revelation [-FA-]
23 km	279 m/s	19 m/s	Carissaz	Revelation [-FA-]
26 km	272 m/s	16 m/s	Dubez	Revelation [-FA-]
10 km	258 m/s	41 m/s	Ezrealil	Revelation [-FA-]
23 km	280 m/s	34 m/s	Tryhit'Agheh	Revelation [-FA-]
22 km	258 m/s	24 m/s	Spivaprime	Revelation
55 km	86 m/s	222 m/s	AdeThelmmo	Ragnarok [CONDI]
21 km	316 m/s	76 m/s	Bad Poison	Ragnarok [CONDI]
18 km	289 m/s	40 m/s	Jimm Hawkin	Phoenix [LAWN]
20 km	264 m/s	24 m/s	handypantz	Phoenix [GENTS]
15 km	239 m/s	27 m/s	William Clark	Phoenix [GENTS]
15 km	259 m/s	26 m/s	Sir Nightmar	Phoenix [GENTS]
25 km	255 m/s	15 m/s	Joker 725	Phoenix [FCOM]
14 km	258 m/s	27 m/s	Vumelyan	Phoenix [CONDI]
17 km	270 m/s	33 m/s	Axhind	Phoenix [CONDI]
25 km	228 m/s	33 m/s	Aranorn	Phoenix [CONDI]
21 km	281 m/s	22 m/s	CamelDog	Phoenix [CONDI]
18 km	240 m/s	15 m/s	Tobin Goldha	Phoenix [-FA-]
26 km	244 m/s	12 m/s	SubZero101	Phoenix [-FA-]
10 km	260 m/s	32 m/s	Infinint	Phoenix [-FA-]
28 km	250 m/s	8 m/s	ForceMajeure	Phoenix [-FA-]
19 km	282 m/s	21 m/s	Easthir Ravin	Phoenix [-FA-]
16 km	272 m/s	50 m/s	Mirple	Nyx [TNT]
13 km	274 m/s	42 m/s	Links Griffen	Nyx [TNT]
24 km	208 m/s	58 m/s	Bisha Wolf	Nyx [LAWN]
23 km	279 m/s	36 m/s	Collussus	Nyx [GENTS]
17 km	275 m/s	51 m/s	Emshard	Nyx [FCOM]

- [02:48:34] Capt JJ + x
 - [02:41:06] Barkin + x
 - [02:43:08] Tessa Yor + primary?
 - [02:43:16] Ron Mexxico + nixor rixon
 - [02:43:47] Alahna Morrin + nixor or bunny alarm
 - [02:43:54] Alahna Morrin + either or
 - [02:45:19] BIGBADBOB + are we able to be contracted any dreads in hak?
 - [02:48:30] Cern Tiamat + for HICs burnth there
 - [02:48:34] Cern Tiamat + is Tama camped?
 - [02:48:06] Cesar Malari + FC: 2J out in a HIC - any particular target/type I should be tackling to avoid not wasting my focused point?
- Admiral Thelaro
 - Airin Zektul
 - Alahna Morrin
 - Alexei Greshenko
 - Argaven
 - Barkin
 - Battlekiller
 - BIGBADBOB
 - Bishop Conrad
 - Bobbi Attwell
 - Bobicus Maximus

Domain Name System

- UDP port 53
- Transforms *host names* into IP addresses
 - **online.tugraz.at** \Rightarrow **129.27.2.210**
- Hierarchical structure
 - **.** \Rightarrow *root nameservers* (typically hardcoded)
 - **at.** \Rightarrow ask **127.30.48.1** (**dns.nic.at**)
 - **tugraz.at.** \Rightarrow ask **129.27.2.3** (**ns1.tu-graz.ac.at**)
 - **online.tugraz.at.** \Rightarrow it's at **129.27.2.210**

Domain Name System

- Typically, the client queries a *DNS resolver* on port 53
 - Well-known public resolvers:
1 . 1 . 1 . 1 (Cloudflare), 8 . 8 . 8 . 8 (Google), 9 . 9 . 9 . 9 (Quad9)
- The DNS resolver performs the actual recursive lookup if needed
 - This allows centralized caching of responses!
- DNS resolver address can also be determined via DHCP
 - Recall: Dynamic Host Configuration Protocol
 - It does IP address auto-configuration, we talked about it 😊

Network Time Protocol

- UDP port 123
- Time synchronization over the internet
- Synchronized clocks are required for many operations
 - Time-based 2FA tokens, expiry of SSL certificates, Kerberos tokens, ...

Synchronize your clock

Last successful time synchronization: 22.09.2021 19:40:07

Time server: europe.pool.ntp.org

Sync now

```
gallantron@ipn009:~$ ntpdate -d pool.ntp.org
23 Sep 14:11:04 ntpdate[159]: ntpdate 4.2.8p12@1.3728-o (1)
Looking for host pool.ntp.org and service ntp
91.206.8.34 reversed to svn.mediaminvent.at
host found : svn.mediaminvent.at

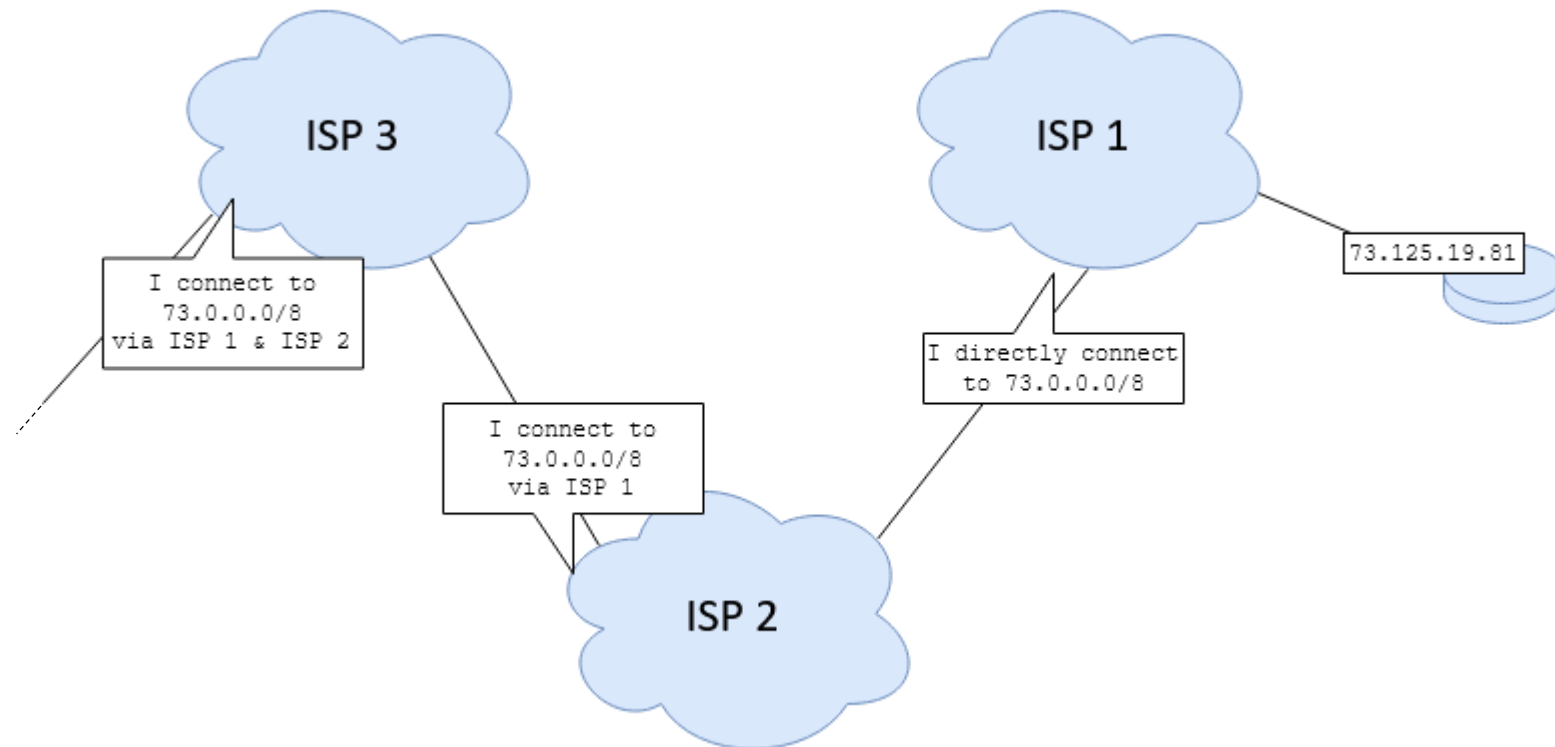
server 91.206.8.34, port 123
stratum 2, precision -21, leap 00, trust 000
refid [161.143.24.141], root delay 0.000610, root dispersion 0.029587
transmitted 4, in filter 4
reference time:      e4f6e9b1.246b51c0  Thu, Sep 23 2021 13:55:29.142
originate timestamp: e4f6ed5e.e6ff1685  Thu, Sep 23 2021 14:11:10.902
transmit timestamp:  e4f6ed5e.e602ffcb  Thu, Sep 23 2021 14:11:10.898
filter delay:  0.04388  0.03754  0.03914  0.03960
               0.00000  0.00000  0.00000  0.00000
filter offset: -0.00477 -0.00492 -0.00568 -0.00318
               0.000000 0.000000 0.000000 0.000000
delay 0.03754, dispersion 0.00082
offset -0.004928
```

Secure SHell

- TCP port 22
- Secure remote administration
 - Unless you use a(n insecure) password...
- Using SSH as a building block in other applications is popular
 - SSH provides authentication + encryption
 - Example: **git**

Border Gateway Protocol

- TCP port 179
- Responsible for maintaining the global IP routing table
 - Essentially a distributed shortest-path graph algorithm



HyperText Transfer Protocol

- TCP port 80 (HTTP), TCP port 443 (HTTP over SSL)
 - SSL adds authentication & encryption – more on this next year!
- Every web page you view uses it

- Simple concept: ask the server for a document
 - The meaning of “document” has evolved greatly over the years
 - Originally: actual *document*, a static piece of content
 - Today: anything you can possibly imagine, often dynamically generated
 - Many applications communicate via HTTP due to its ubiquitous support

HTTP request

Method

Requested resource

```
GET /document.html HTTP/1.1\r\n  
Accept: text/html\r\n  
Host: webserver.net\r\n  
User-Agent: SimpleWebBrowser\r\n  
Connection: keep-alive\r\n  
\r\n
```

Request line

Headers

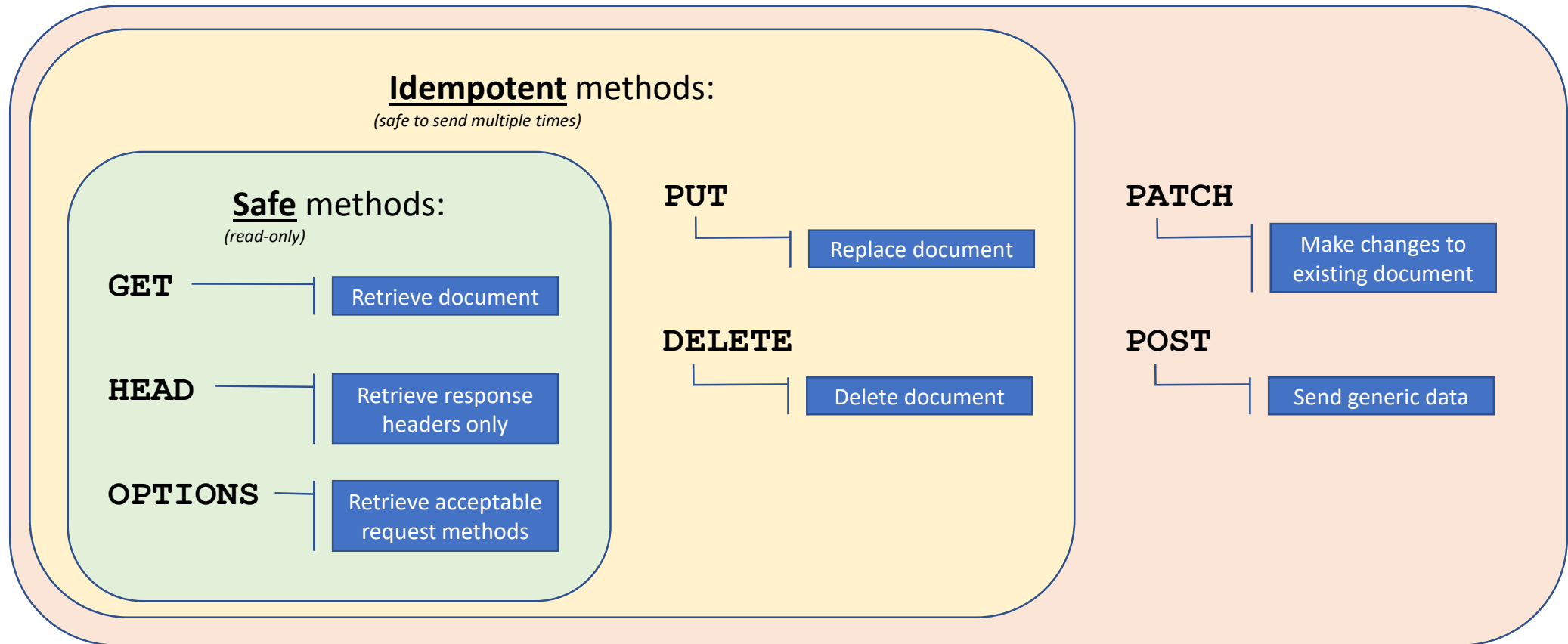
Blank line – end of headers

HTTP response

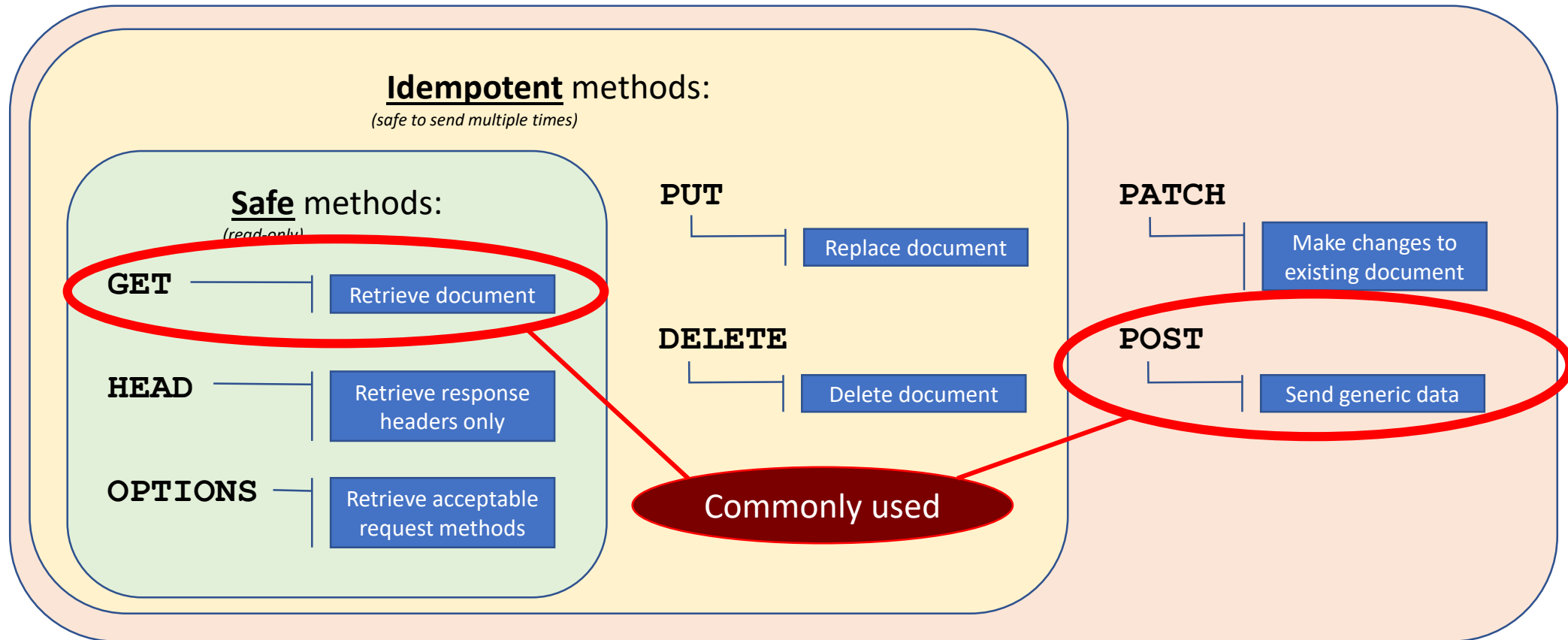
Status code

```
HTTP/1.1 200 OK\r\n } Status line  
Content-Type: text/html; charset=UTF-8\r\n }  
Content-Length: 47\r\n } Headers  
Server: MyWebServer\r\n }  
Connection: close\r\n }  
\r\n } Blank line – end of headers  
<html><head><title>Hello!</title></head></html>  
Requested document
```

HTTP methods

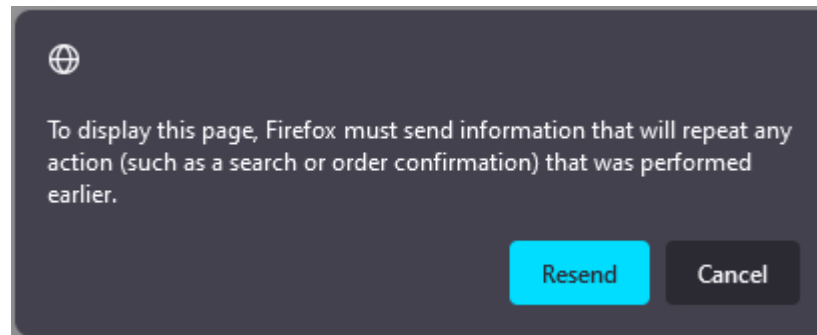


HTTP methods



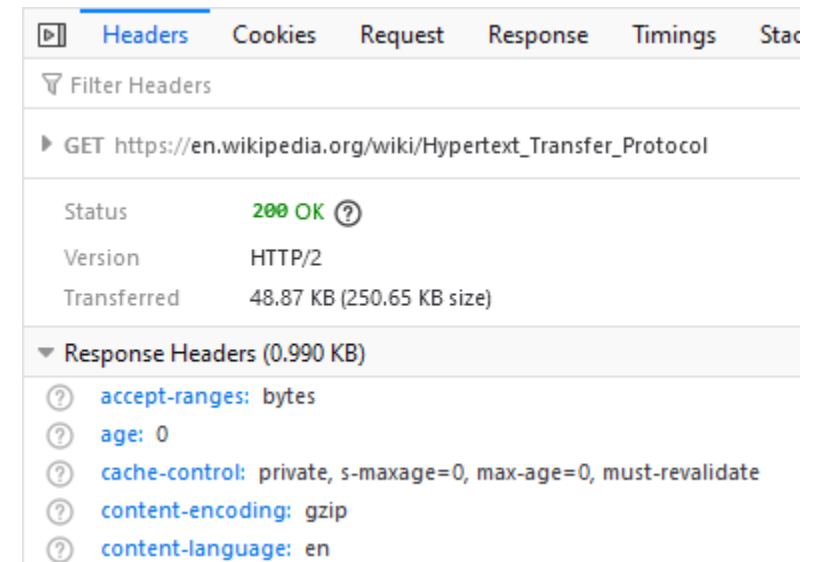
HTTP methods

- Method functionality is purely by convention
 - There's nothing stopping you from deleting a file when a **GET** request is made
 - Just because you *can*, doesn't mean you *should*...
- Clients will offer different degrees of safeguards for different methods
 - Example: Reloading the result of a **POST** request triggers a dialog box



HTTP – you can try this at home!

- Open a new browser tab
- Enable the developer tools (F12 in Firefox and Chrome)
- Switch to the “Network” tab
- Open your favorite website
- Each line is one HTTP request being made
 - Click on them to see what’s happening!



HTTP/1.1: Head-of-Line blocking

- Multiple requests can be sent over a single connection!
 - But: the responses still need to come in order...
 - **search.php?search=bismuth**

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8"><meta http-equiv="x-ua-compatible" content="ie-edge">
    <title>Searching for 'bismuth'...</title>
    <link rel="stylesheet" href="css/search.css">
    <script defer src="js/search.js"></script>
  </head>
  <body>
    <h1>Your results are:</h1>
    <ul>
      <li><a href="gem.php?id=83">Bismuth<a></li>
      <li><a href="alloy.php?id=LBE">Lead-Bismuth Eutectic</a></li>
    </ul>
  </body>
</html>
```

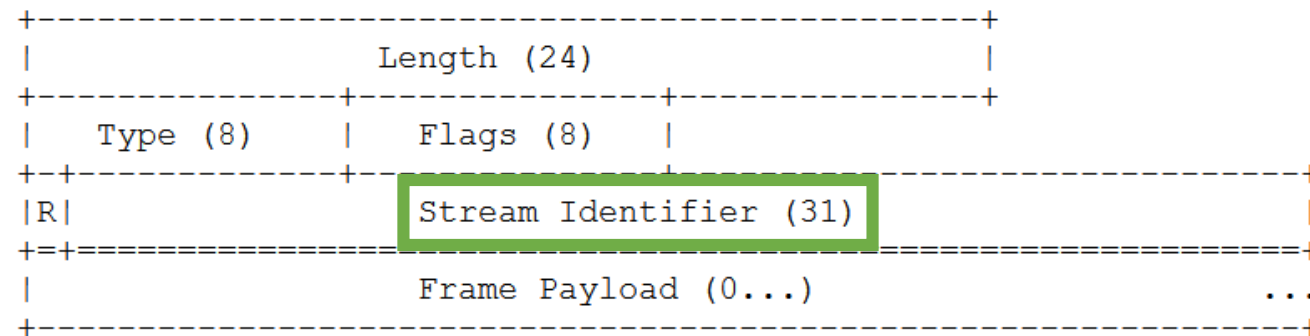
We'll need these two, too!

HTTP/1.1: Head-of-Line blocking

- Multiple requests can be sent over a single connection!
 - But: the responses still need to come in order...
 - `search.php?search=bismuth` ← Slow database query
 - `search.css` ← Fast file request → Needs to wait →
 - Even though we know we need the file, we can't retrieve it...

HTTP/2

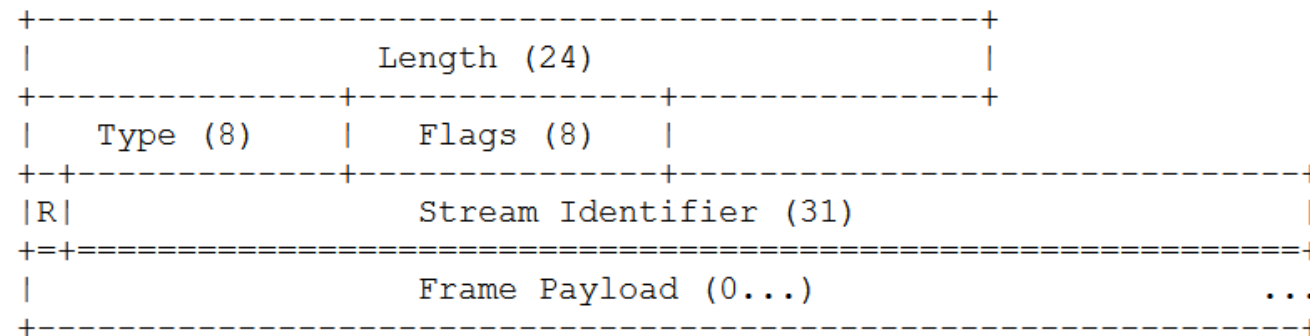
- No longer human readable
 - Binary representation is more compact



- Same request/response semantics, header fields, etc...
- Stream Identifier allows multiple responses to be sent in parallel

HTTP/2

- No longer human readable
 - Binary representation is more compact



- Same request/response semantics, header fields, etc...
- Stream Identifier allows multiple responses to be sent in parallel
- Still runs over a TCP connection...

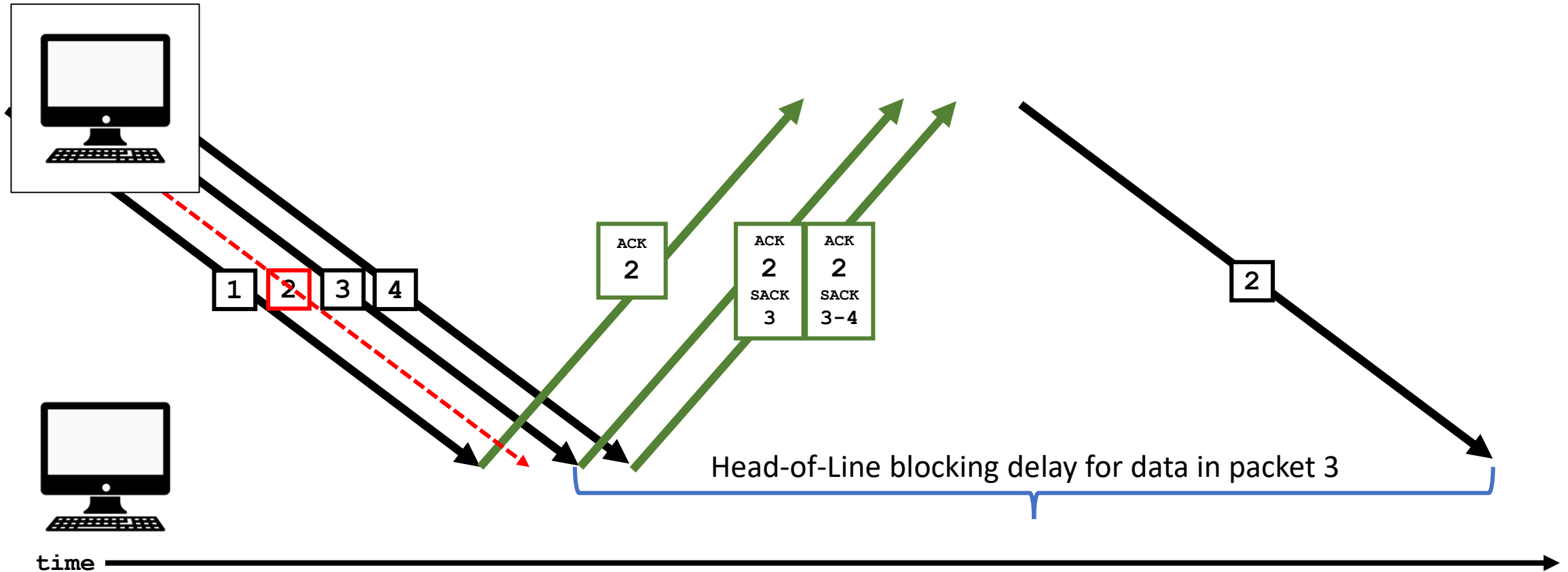


The Transport Layer

(again, for a brief interlude)

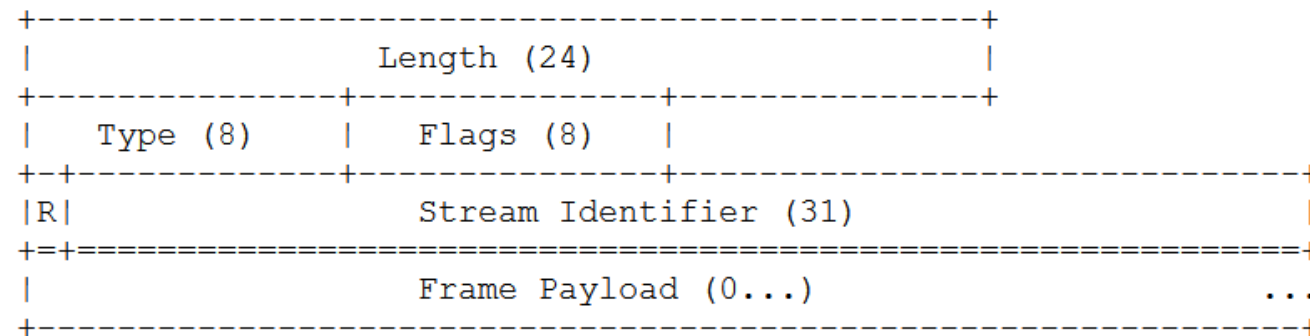
TCP: Head-of-Line blocking

- The TCP connection is still a single byte stream
 - While we wait for packet 2, we cannot process packets 3 and 4...



HTTP/2

- No longer human readable
 - Binary representation is more compact



- Same request/response semantics, header fields, etc...
- Stream Identifier allows multiple responses to be sent in parallel
- Still runs over a TCP connection...
 - Packet loss on any stream “pauses” all streams’ data

HTTP/3

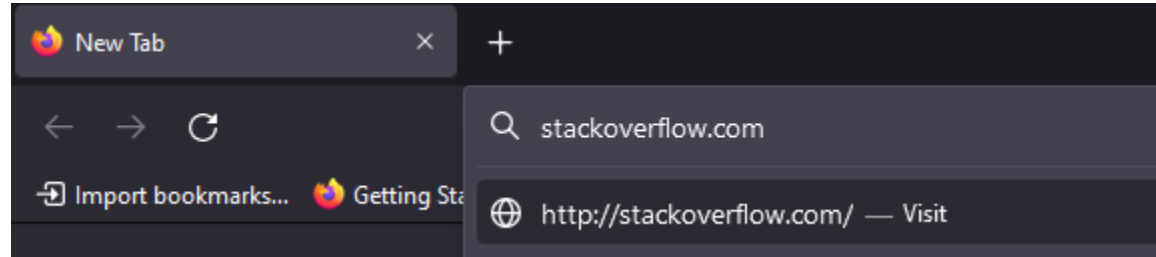
- QUIC to replace TCP
 - **Q**uick **U**DP **I**nternet **C**onnection
 - Originally developed by Google, since standardized by IETF
 - UDP at the transport layer for minimal overhead
 - Provides byte stream facilities similar to TCP
 - Aware of multiple data streams within a single connection
 - ... re-invents the wheel, in a way
- Otherwise identical to HTTP/2

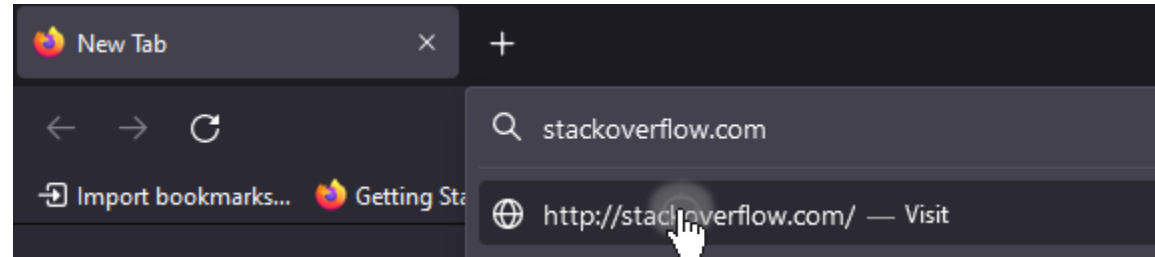
... and many, many, many, many more ...

- Any program with network features is part of “the application layer”
 - Try running **netstat** while logged into your favorite game!

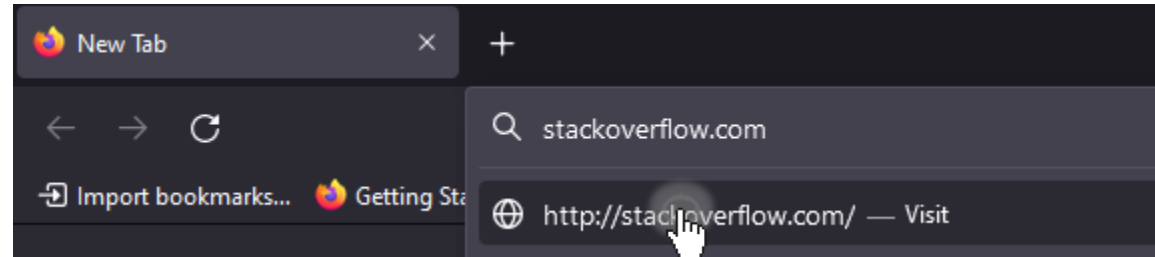


Putting it all together

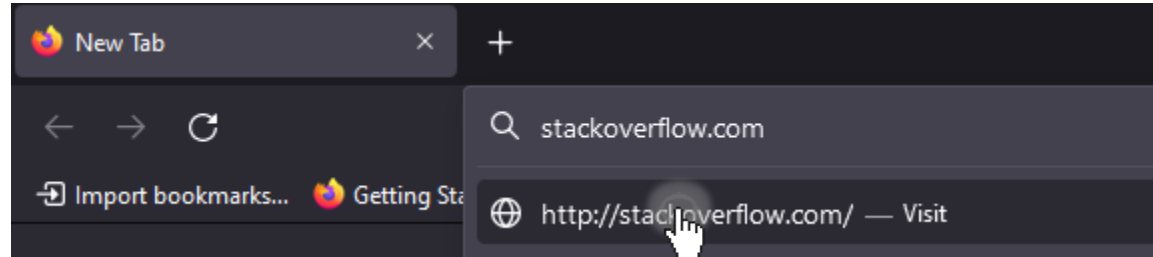




- OK, what happens?



- Step 1: We need to figure out where **stackoverflow.com** is!
 - Let's ask our favorite DNS resolver!
 - DNS resolvers listen on UDP port 53
 - UDP is stateless, so we can just send our DNS request



- Let's ask our favorite DNS resolver where **stackoverflow.com** is!



Hi, transport layer. Please send the following data to 8.8.4.4 on UDP port 53:

b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01

- Let's ask our favorite DNS server where **stackoverflow.com** is!



Hi, transport layer. Please send the following data to 8.8.4.4 on UDP port 53:

b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01

Pick an unused UDP port... 49640

Attach a UDP header... done.

Network layer, please send the following data to 8.8.4.4:

c1 e8 00 35 00 23 00 00

b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01



UDP datagram header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															



Pick an unused UDP port... 49640
 Attach an UDP header... done.
 Network layer, please send the following data to 8.8.4.4:
 c1 e8 00 35 00 23 00 00
 b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01



Our IP address... it's 192.168.42.18.
 What's the next hop towards 8.8.4.4? The default gateway, 192.168.42.254.
 What's the MAC address for that... ah, it's f2:d3:09:9d:53:1a.
 Link Layer, please send the following data to f2:d3:09:9d:53:1a via port eth0:
 45 00 00 3f 1a 8a 00 00 ff 11 aa 5d c0 a8 2a 12 08 08 04 04 c1 e8 00 35 00 23 00 00
 b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01

45 00 00 3f 1a 8a 00 00 ff 11 aa 5d c0 a8 2a 12 08 08 04 04 c1 e8 00 35 00 23 00 00
 b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01

Offsets	Octet	0				1				2				3																			
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				Header Length				DSCP				ECN				Total Length															
4	32	Identification								Flags				Fragment Offset																			
8	64	Time To Live				Protocol				Header Checksum																							
12	96	Source IP Address																															
16	128	Destination IP Address																															



Our IP address... it's 192.168.42.18.

What's the next hop towards 8.8.4.4? The default gateway, 192.168.42.254.

What's the MAC address for that... ah, it's f2:d3:09:9d:53:1a.

Link Layer, please send the following data to f2:d3:09:9d:53:1a via port eth0:

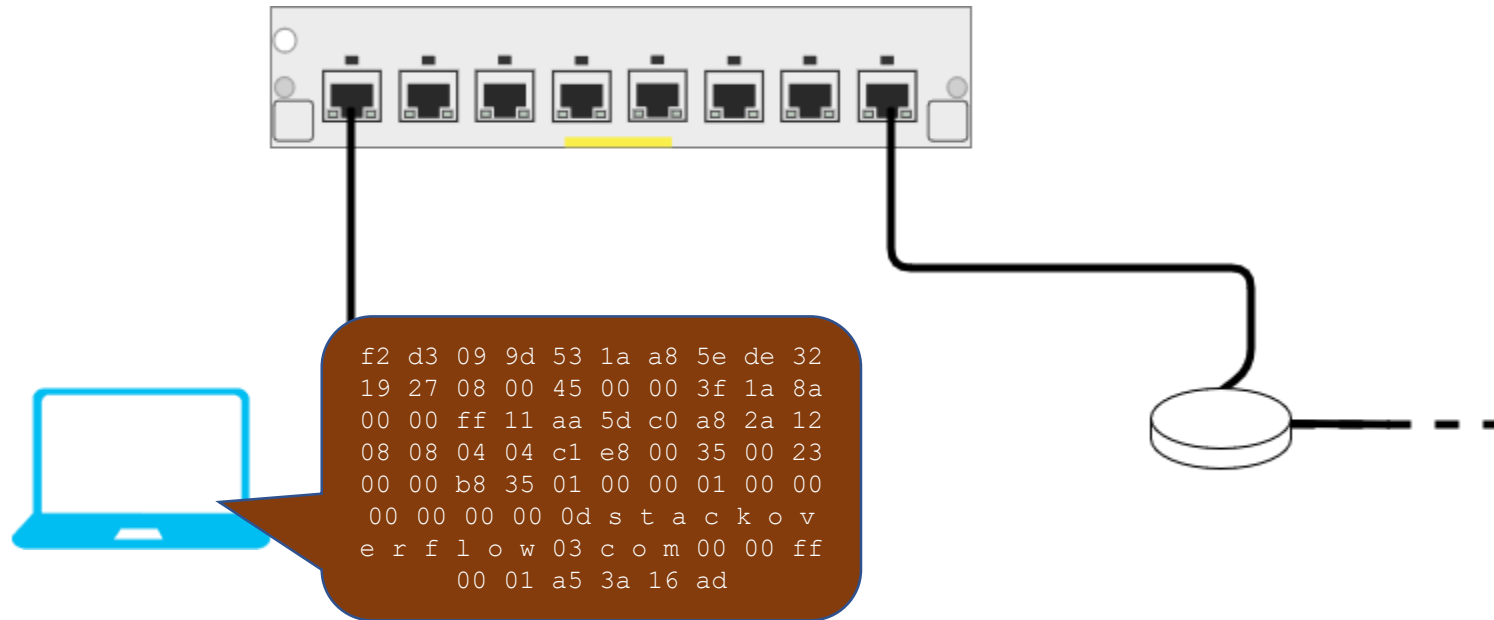
```
45 00 00 3f 1a 8a 00 00 ff 11 aa 5d c0 a8 2a 12 08 08 04 04 c1 e8 00 35 00 23 00 00
b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01
```

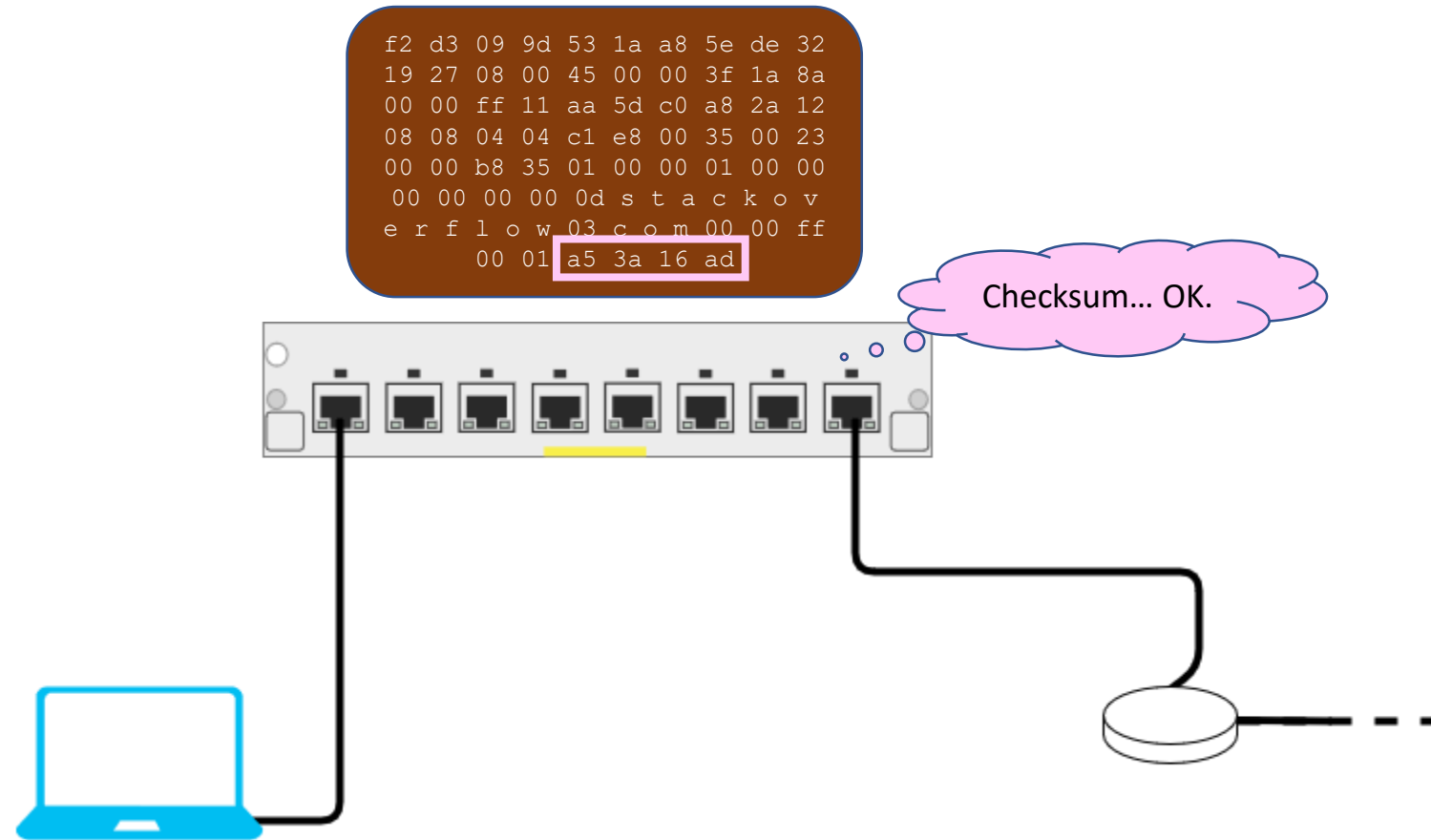
Our MAC address is a8:5e:de:32:19:27.

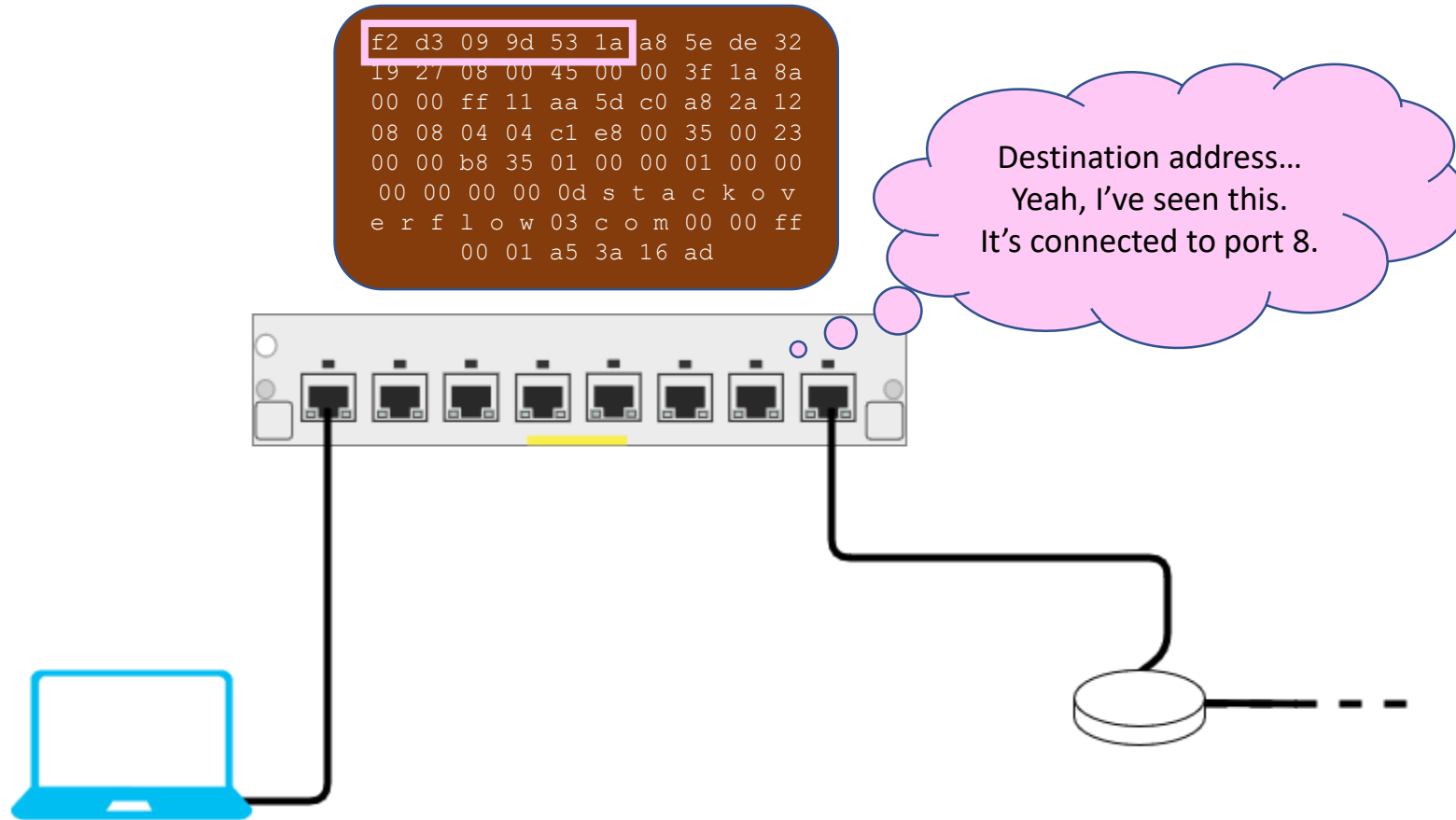
Hi, other side. Here's frame data modulated into the electrical current flowing over the wire:

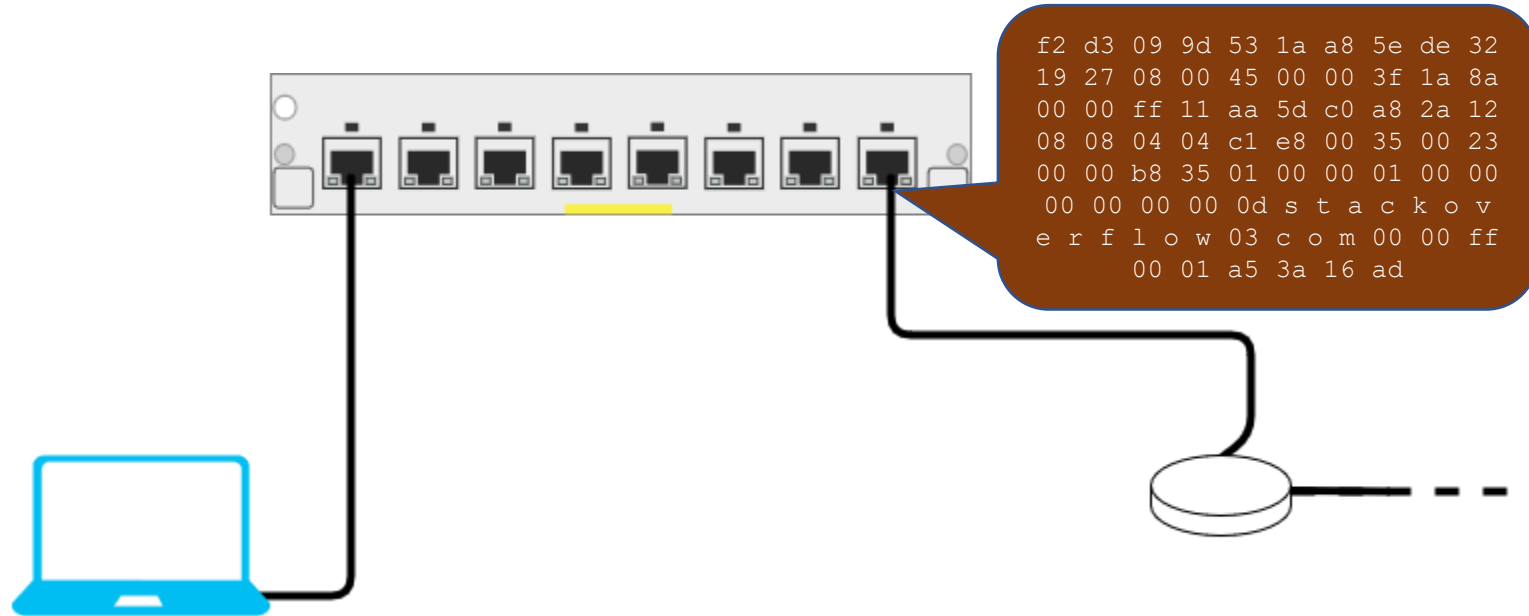
```
f2 d3 09 9d 53 1a a8 5e de 32 19 27 08 00 45 00 00 3f 1a 8a 00 00 ff 11 aa 5d c0 a8 2a 12 08
08 04 04 c1 e8 00 35 00 23 00 00 b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f
l o w 03 c o m 00 00 ff 00 01 a5 3a 16 ad
```

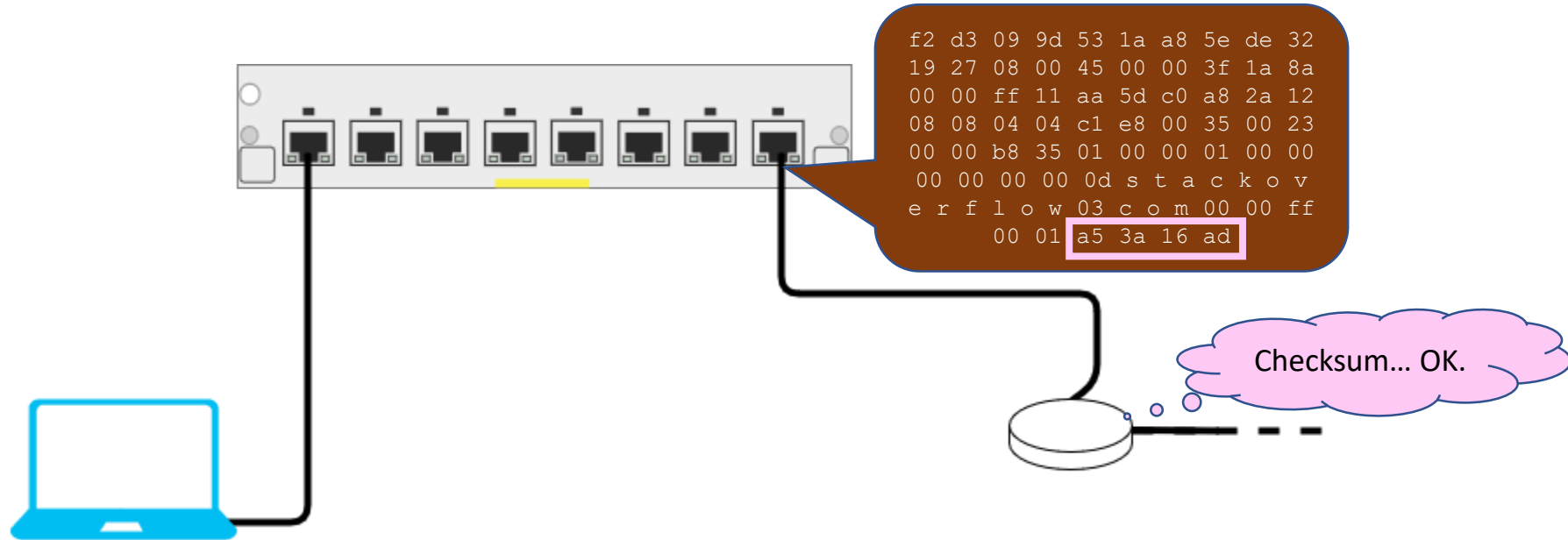


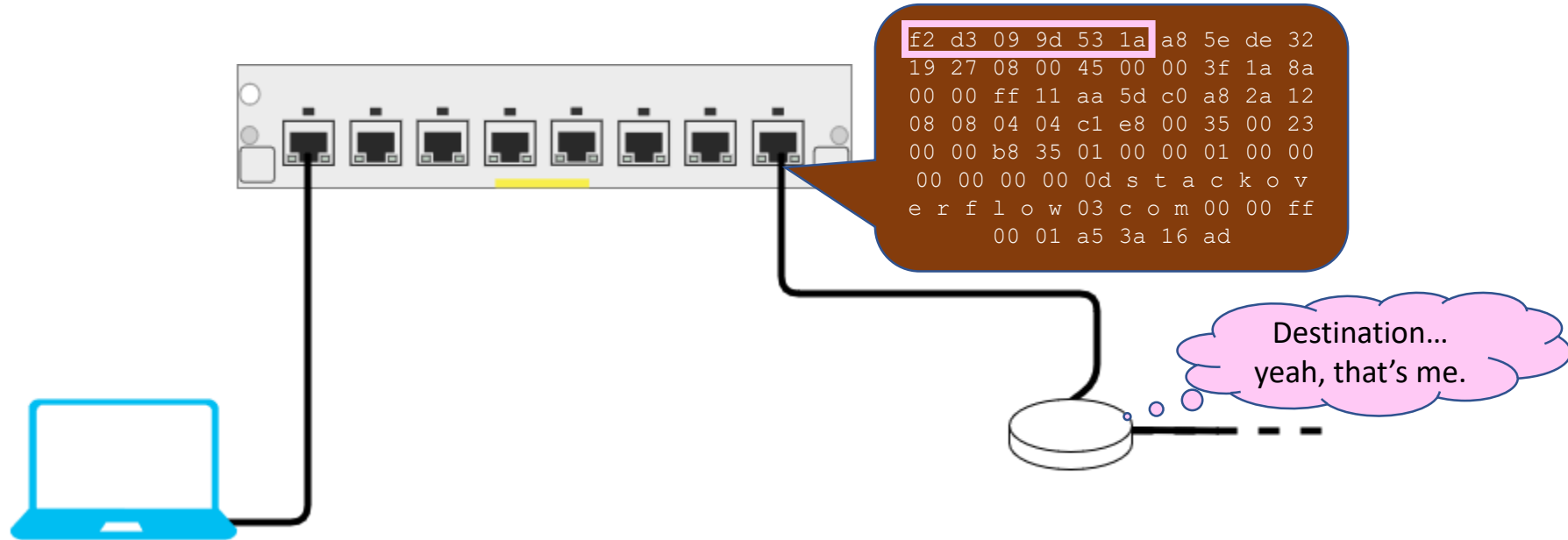














Hey, Network Layer, data arrived for you:

```
45 00 00 3f 1a 8a 00 00 ff 11 aa 5d c0 a8 2a 12 08 08 04 04 c1 e8 00 35 00 23 00 00  
b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01
```



Hey, Network Layer, data arrived for you:

```
45 00 00 3f 1a 8a 00 00 ff 11 aa 5d c0 a8 2a 12 08 08 04 04 c1 e8 00 35 00 23 00 00
b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01
```

Let's see... destination address 8.8.4.4? That's not me, I need to pass this on...

First, decrease the TTL field, and recalculate the checksum... then...

Next hop in that direction is... 192.168.255.254...

I have its MAC address cached... a7:a2:23:95:d6:a6.

Link Layer, please send this data to a7:a2:23:95:d6:a6 via port eth1:

```
45 00 00 3f 1a 8a 00 00 fe 11 ab 5d c0 a8 2a 12 08 08 04 04 c1 e8 00 35 00 23 00 00
b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01
```



Let's see... destination address 8.8.4.4? That's not me, I need to pass this on...

First, decrease the TTL field, and recalculate the checksum... then...

Next hop in that direction is... 192.168.255.254...

I have its MAC address cached... a7:a2:23:95:d6:a6.

Link Layer, please send this data to a7:a2:23:95:d6:a6 via port eth1:

```
45 00 00 3f 1a 8a 00 00 fe 11 ab 5d c0 a8 2a 12 08 08 04 04 c1 e8 00 35 00 23 00 00
b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f l o w 03 c o m 00 00 ff 00 01
```



Our MAC address on that interface... 58:42:30:35:8a:08.

Hi, other side, here's data:

```
a7 a2 23 95 d6 a6 58 42 30 35 8a 08 08 00 45 00 00 3f 1a 8a 00 00 ff 11 aa 5d c0 a8 2a 12 08
08 04 04 c1 e8 00 35 00 23 00 00 b8 35 01 00 00 01 00 00 00 00 00 00 0d s t a c k o v e r f
l o w 03 c o m 00 00 ff 00 01 f6 11 22 44
```



...



Looking Forward...

Let's Get Dangerous

- So far, everyone has played nice...
 - Real life is difficult, and people are people

- INP.33404UF Information Security
 - Suddenly, everyone is evil – and so are you
 - See you next year!