AARHUS UNIVERSITET

# Software Engineering and Architecture

Networking 101

AARHUS UNIVERSITET

# Motivation

- Networking - not a curriculum issue in SWEA…
  - … it is next semester in another course
- But…
  - You see it everywhere
  - And you need some ‘Network for dummies’ for our Broker…

# A Network

- … in CS is basically two or more machines connected by electrical wires that allows us to send signals between the machines…

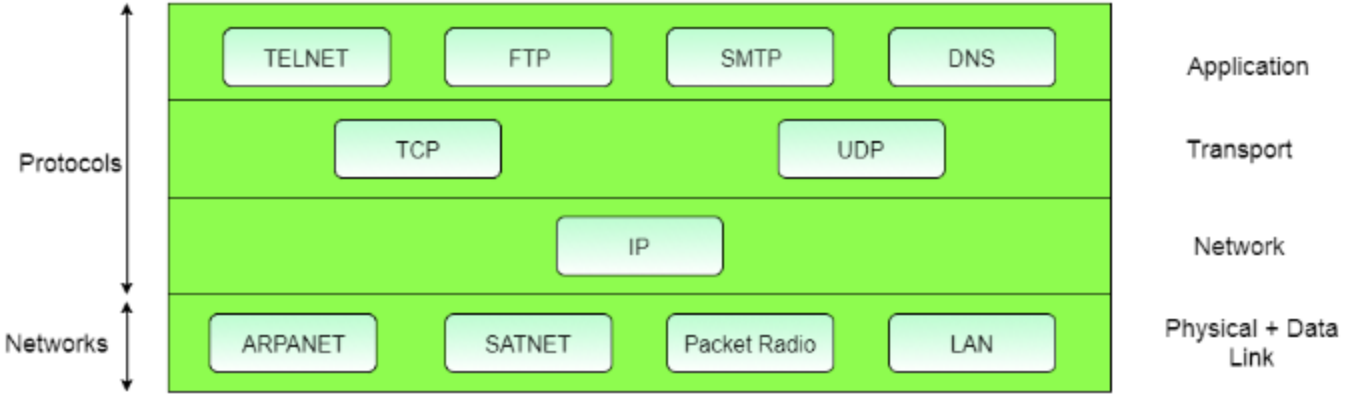
# A Network

- My first exposure: RS232 on Z80 CPUs

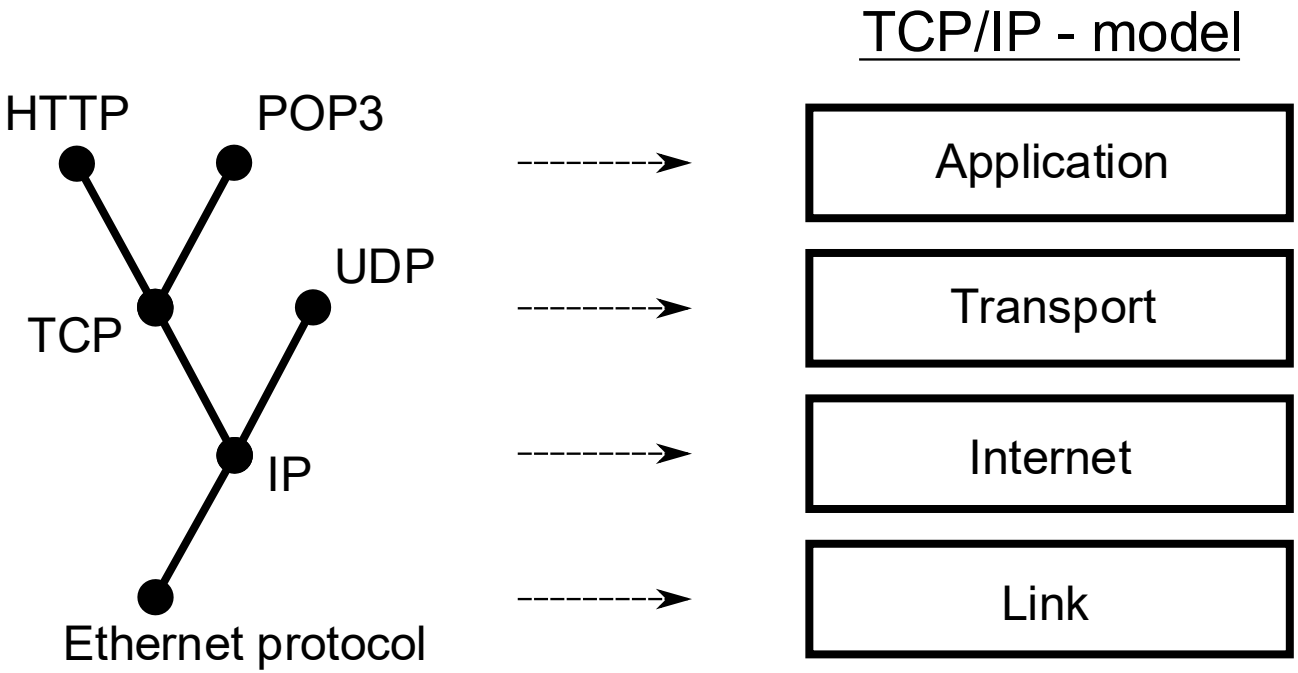
- Today's web: TCP/IP over Ethernet

# TCP/IP

- Transmission Control Protocol and Internet Protocol
  - By the US Department of Defence (DARPA)
- Key Idea
  - Segment transmission into Packets (”Datagrams”)
  - Layered architecture, each with specific responsibilities (roles!)



# TCP/IP



AARHUS UNIVERSITET

# OSI Model

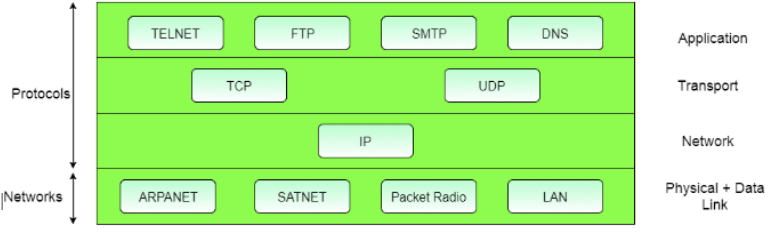
- Another but similar model

| OSI Model | | | |
|---|---|---|---|
| Layer | | Protocol data unit (PDU) | Function[3] |
| Host layers | 7. Application | Data | High-level APIs, including resource sharing, remote file access |
| | 6. Presentation | | Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption |
| | 5. Session | | Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes |
| | 4. Transport | Segment, Datagram | Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing |
| Media layers | 3. Network | Packet | Structuring and managing a multi-node network, including addressing, routing and traffic control |
| | 2. Data link | Frame | Reliable transmission of data frames between two nodes connected by a physical layer |
| | 1. Physical | Symbol | Transmission and reception of raw bit streams over a physical medium |

AARHUS UNIVERSITET

# TCP/IP Layers

- Transport layer
  - TCP — Reliable, ordered, error-checked data delivery
    - Transmission Control Protocol
- Network / Internet Layer
  - IP — Relaying datagrams across networks
    - Internet protocol
- Physical + Data Link Layer
  - 802.3 Ethernet — Hardware and cables
  - 802.11 WiFi — Cables gone



AARHUS UNIVERSITET

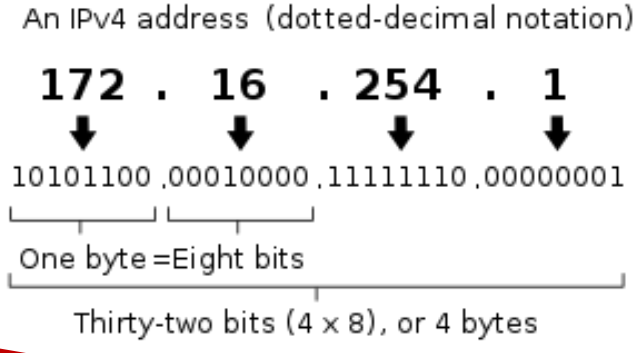
# Internet Protocol

IP: Send datagram

# IPv4

- Defines the terminology that we use and it pops up even at the software level
- *Every* computer on the network has an *address*
  - Type 'ifconfig'/'ipconfig' to find yours

```
d:\proj\Book>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : st.client.au.dk
   Link-local IPv6 Address . . . . . : fe80::9c07:4b59:f478:f9c%4
   IPv4 Address. . . . . . . . . . . : 10.17.103.50
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.17.103.1
```




- Some ranges are reserved
  - 10.*.*.*, 172.16.*.*., 192.168.*.* are private networks
  - 127.0.0.1 is *localhost* = myself

# IP and Ports

- So given an IP (like 91.221.196.224) you uniquely identify a computer
- The OS of that computer expose 64K **ports**
  - Also predefined port numbers
    - 7: echo ('ping')
    - 20: ftp
    - 22: ssh
    - **80: HTTP**
    - **443: HTTPS**

Think: port = dueslag/"postkasse"

- Thus
  - 91.221.196.224**:80** is the HTTP port of a specific computer
    - If port 80 is active it is probably a web server

AARHUS UNIVERSITET

# Ports

- On Linux, all ports below 1024 are reserved for ‘root’
  - Only *superuser* (administrator) can use these…
- Above that, it is ‘free game’ to assign/use a port, but you may interfere with other programs that have picked one...

| | | | | |
|---|---|---|---|---|
| 4000 | Yes | Yes | *Diablo II* game | Unofficial |
| 5000–5500 | No | Yes | *League of Legends*, a multiplayer online battle arena video game[188] | Unofficial |

# Datagram

- So, for node A and node B to communicate some data
  - Say, a request for a web page, and the server reply
- A creates a request
  - N datagrams (the data segmented into packet size)
  - Each datagram contains
    - Part # i (”packet i”) of the full data
    - Destination IP address — Who is to receive?
    - Source IP address — Who should have the reply?
- B creates a reply
    - Of course the same ☺

# Ping

- Port 7 is reserved for ‘ping’
  - A classic availability pattern: ‘ping/echo’
    - Verify that a given machine is currently turned on
- I have a computer running ‘www.baerbak.com’

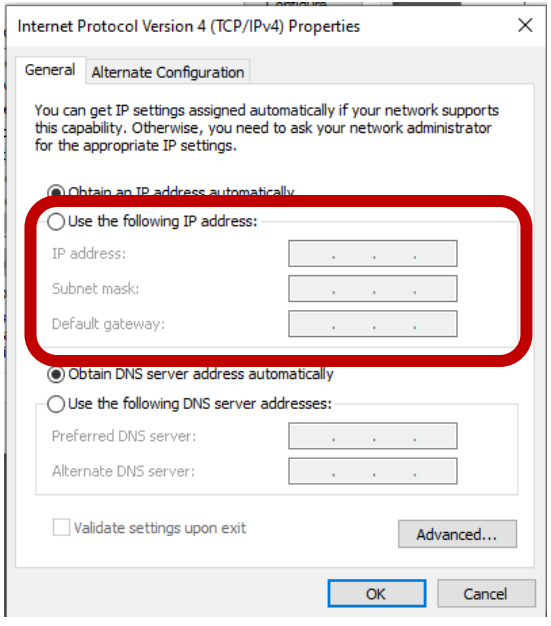
```
d:\work>ping www.baerbak.com

Pinging baerbak.com [91.221.196.224] with 32 bytes of data:
Reply from 91.221.196.224: bytes=32 time=8ms TTL=56
Reply from 91.221.196.224: bytes=32 time=9ms TTL=56
Reply from 91.221.196.224: bytes=32 time=9ms TTL=56
Reply from 91.221.196.224: bytes=32 time=9ms TTL=56

Ping statistics for 91.221.196.224:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 9ms, Average = 8ms
```

Will send datagram on port 7 and print round trip time

```
d:\work>ping www.baerbak.com

Pinging baerbak.com [77.111.240.173] with 32 bytes of data:
Reply from 77.111.240.173: bytes=32 time=7ms TTL=52
Reply from 77.111.240.173: bytes=32 time=7ms TTL=52
Reply from 77.111.240.173: bytes=32 time=8ms TTL=52
Reply from 77.111.240.173: bytes=32 time=8ms TTL=52

Ping statistics for 77.111.240.173:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 8ms, Average = 7ms
```

# Uhum?

- *So how did your machine get that IP address?*
- Either *static IP* (never used except…)
  - You assign an IP directly on the machine!
    - (and hope no other on the network *has the same address!)*
- Or *dynamic IP*
  - DHCP – dynamic host configuration protocol
    - Broadcast a “help – I need an address”
    - DHCP server will assign one and give it
      - Usually your router at home
      - At CS it is a dedicated server machine



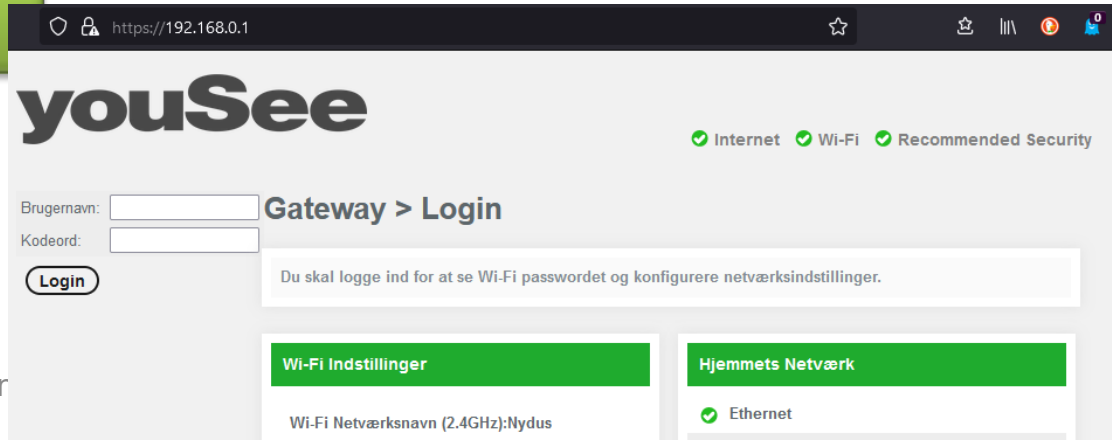
AARHUS UNIVERSITET

# DHCP

- On my Windows machine at home...
  - ipconfig /all

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : home
   Description . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   Physical Address. . . . . . . . . : 4C-ED-FB-68-10-59
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::70bc:3f19:e6a5:e02b%17(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.0.226(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 3. november 2023 07:39:48
   Lease Expires . . . . . . . . . . : 3. november 2023 12:05:54
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-24-91-73-D9-4C-ED-FB-68-10-59
   DNS Servers . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Which is the IP of my router…



# DHCP

- On my Windows machine at my AU office
  - ipconfig /all

And here at AU…

```
Ethernet adapter Ethernet:

   ffix  . : st.client.au.dk
   . . . . : Realtek USB GbE Family Controller
   . . . . : F8-0D-AC-CC-87-62
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::69b3:515f:405e:2775%3(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.17.104.152(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 1. november 2024 12:02:48
   Lease Expires . . . . . . . . . . : 1. november 2024 15:33:09
   Default Gateway . . . . . . . . . : 10.17.104.1
   DHCP Server . . . . . . . . . . . : 10.83.16.53
   DHCPv6 IAID . . . . . . . . . . . : 116947504
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2C-F7-59-9D-C8-E2-65-20-B2-DC
   DNS Servers . . . . . . . . . . . : 10.83.252.137
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

AARHUS UNIVERSITET

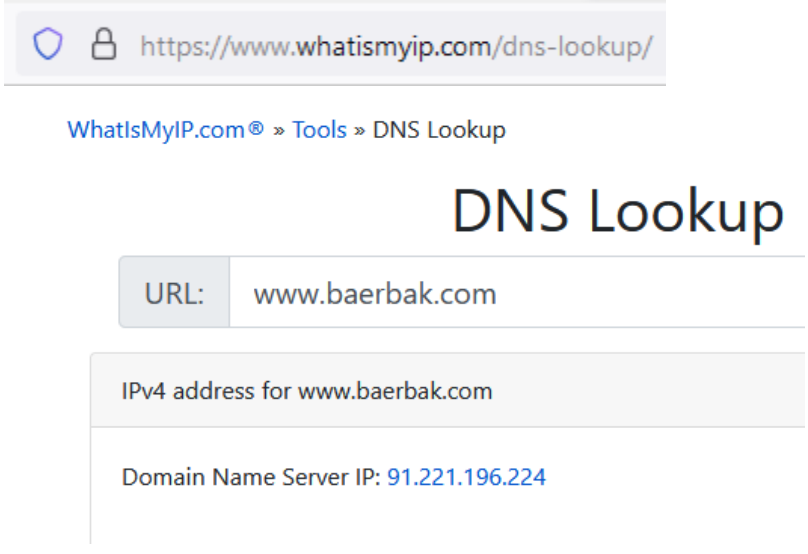
# Domain Name System

IP addresses are a bit hard to remember, right?

# DNS

- Who can remember 87.238.248.136 ???

- DNS (Domain Name System) are *Name Services*
  - Computers that translate names into IP addresses




```
d:\pro >ping www.baerbak.com

Pinging baerbak.com [91.221.196.224] with 32 bytes of data:
Reply from 91.221.1[illegible] time=12ms TTL=59
Reply from 91.221.196.224: bytes=32 time=11ms TTL=59
Reply from 91.221.196.224: bytes=32 time=11ms TTL=59
Reply from 91.221.196.224: bytes=32 time=11ms TTL=59

Ping statistics for 91.221.196.224:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 12ms, Average = 11ms
```

AARHUS UNIVERSITET

# Local names

- *Any* computer has its own name
  - *Normally you give it a name when installing*
- On Linux you may change it by editing a few files

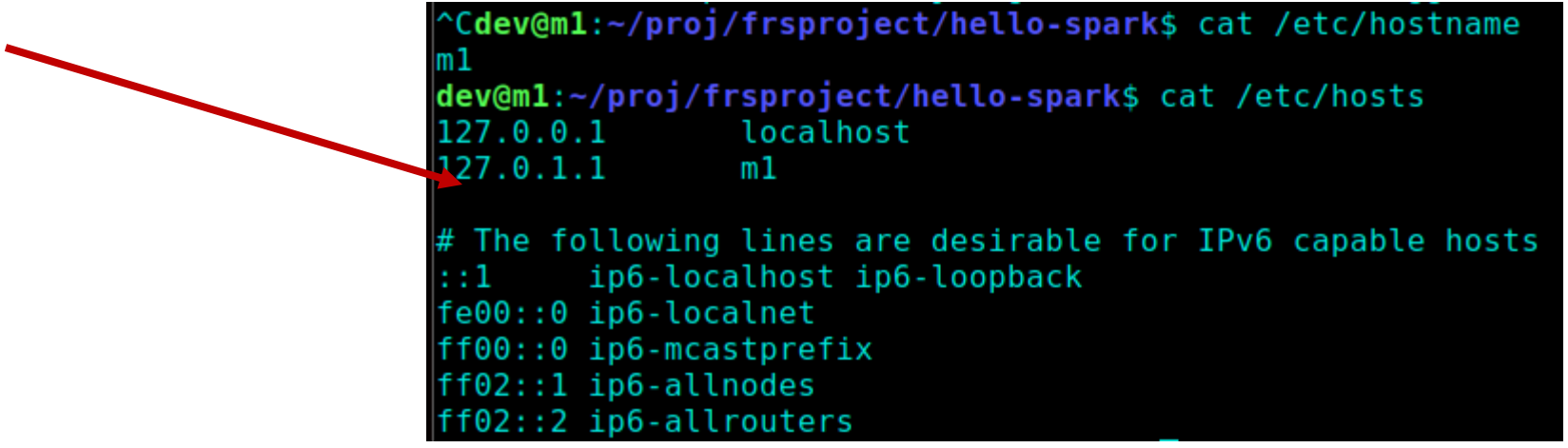
```
^Cdev@m1:~/proj/frsproject/hello-spark$ cat /etc/hostname
m1
dev@m1:~/proj/frsproject/hello-spark$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       m1

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- **Localhost** is 127.0.0.1 which is the IP address of the computer itself! The name of it is ‘localhost’

AARHUS UNIVERSITET

# Your Own DNS

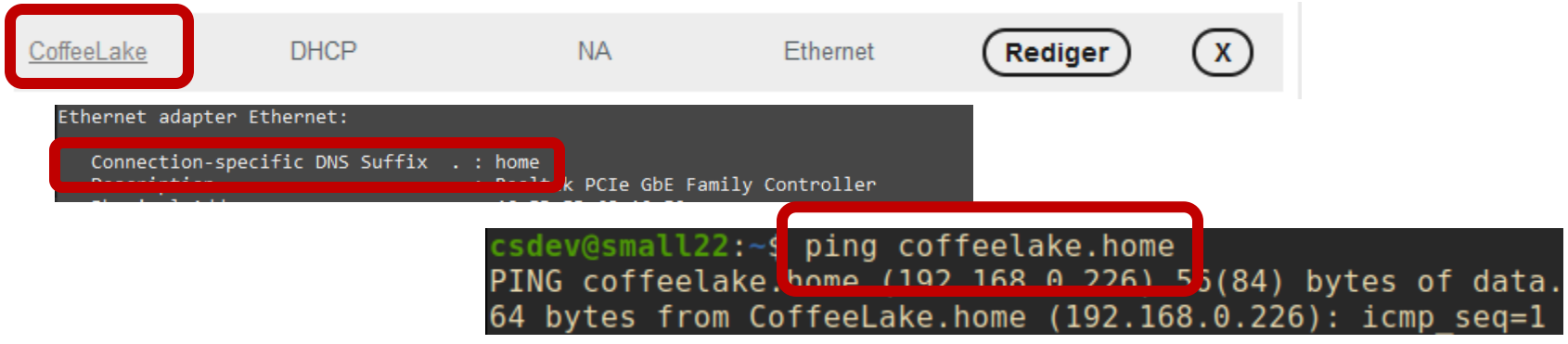
- You can actually maintain your own DNS by editing the *hosts* file on Linux
  - Do not ☺ - *it only works on my machine!*



AARHUS UNIVERSITET

# Local DNS

- At my home I have a **WiFi Router,** supplied by my internet provider. It creates a *local network with local names*

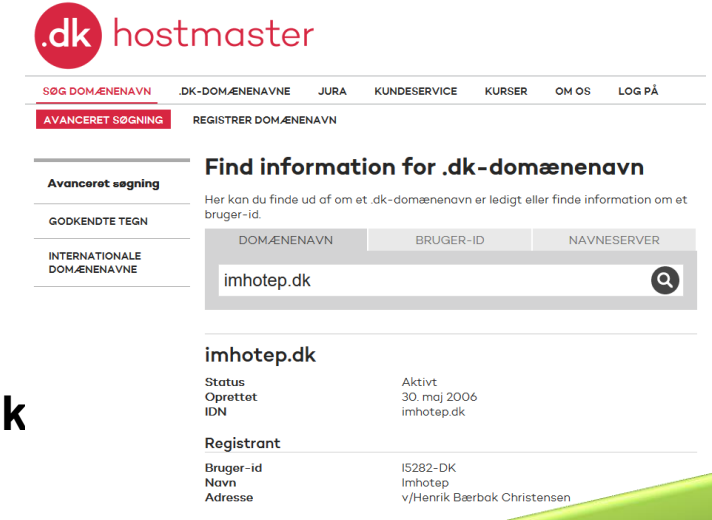



- Of course, only computers on *this* network can access each other by their names
  - Not visible on the ‘full internet’

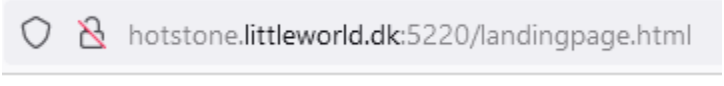
AARHUS UNIVERSITET

# Global DNS

- But what if I want to expose a node on the full internet?
- So – how do I get a **global domain name**?
- For '.dk' domains **DK-Hostmaster** keeps track of all Danish domains




- It costs money
  - ~50 Dkr pr year
    - I own **littleworldk.dk, imhotep.dk, and baerbak.dk**



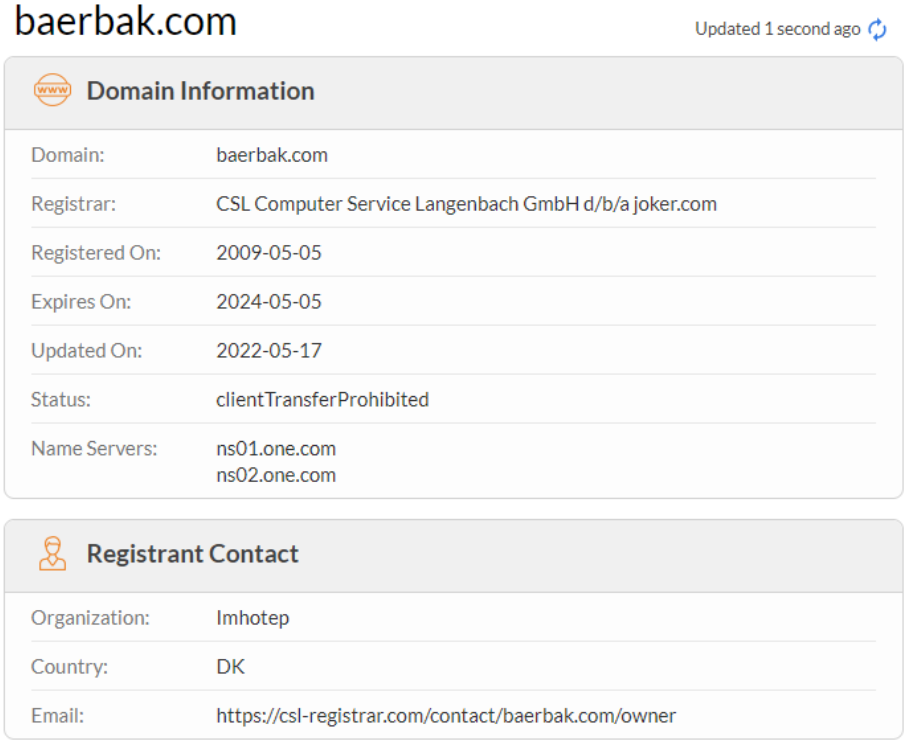

Now: punktum.dk

AARHUS UNIVERSITET

# Global DNS

- For other domains, you need a ‘registrar’ that handles global registration for you

- (From 2025, you also need that for .dk domains)

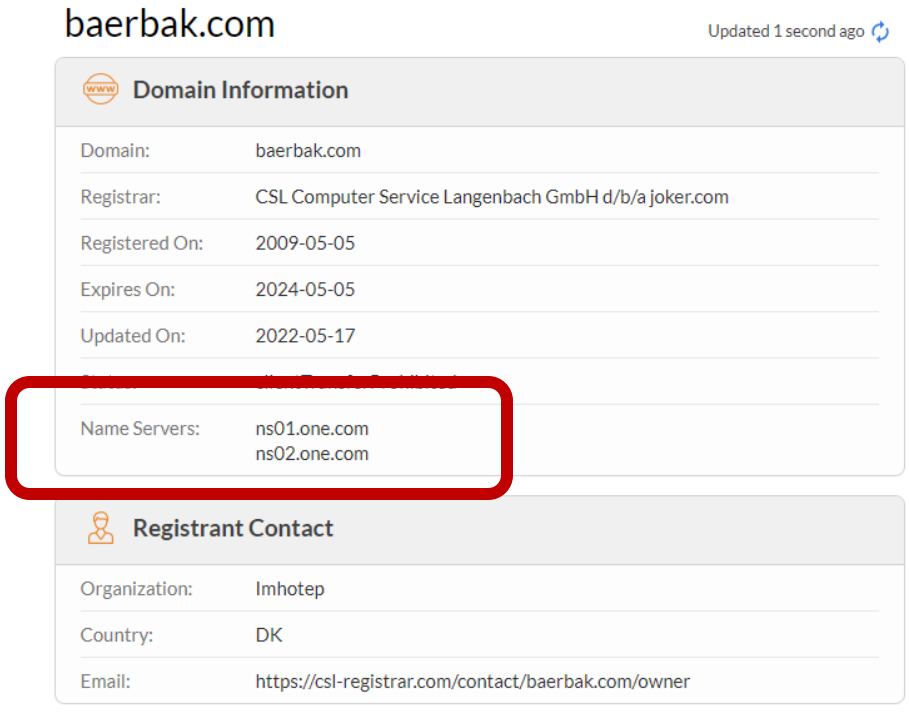


AARHUS UNIVERSITET

# Name Servers

- But you only *register* the domain, you need a Name Server to handle the actual lookup
  - My web sites are hosted by *one.com*, and they provide the name servers

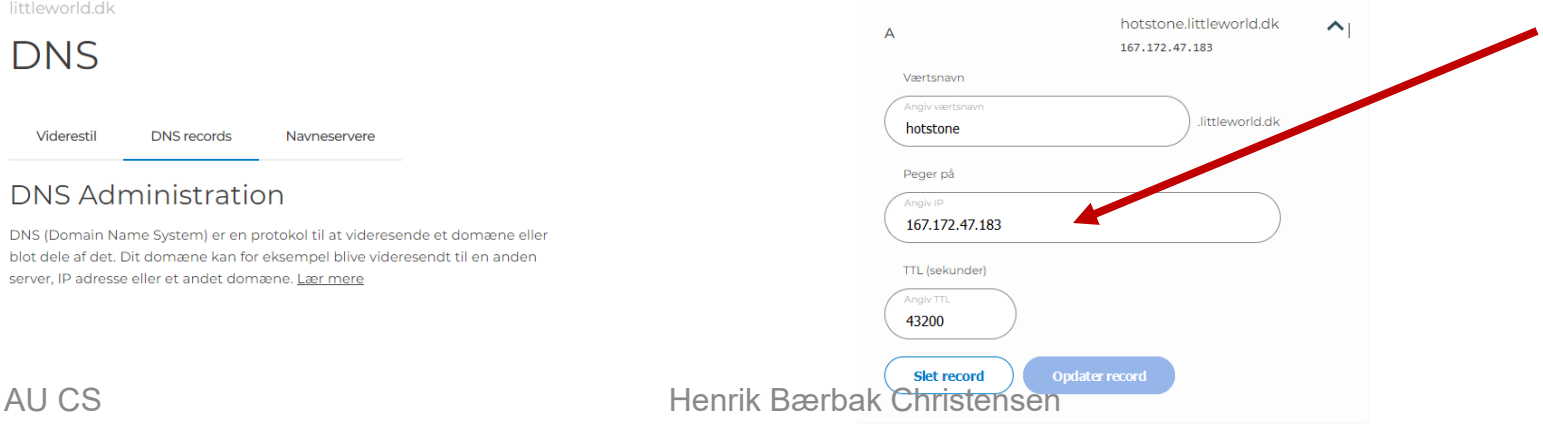
- Exercise
  - Why do you think there are *two* name servers listed ?



AARHUS UNIVERSITET

# Create a New Name

- Scenario: *I want to provide a HotStone game server*
- I do
  - Rent a virtual machine on Hetzner (Serverfarm in Helsinki)
    - So I get an IP address of that machine
- I log into my ‘one.com’ account, go to the ‘DNS Setting’ pane, and create an ‘A record’ with that IP address



AARHUS UNIVERSITET

# Local DNS

- Organizations, like CS, maintain their own local network
  - And thus needs a DNS for the **local** machines
    - Which are 'visible' on the local net, but not on the global (=inter)net

- One of my machines is m1-dev on st.lab.au.dk network

```
d:\proj\Book>ping m1-dev.st.lab.au.dk

Pinging m1-dev.st.lab.au.dk [10.28.27.17] with 32 bytes of data:
Reply from 10.28.27.17: bytes=32 time=1ms TTL=58
Reply from 10.28.27.17: bytes=32 time=1ms TTL=58
```

AARHUS UNIVERSITET

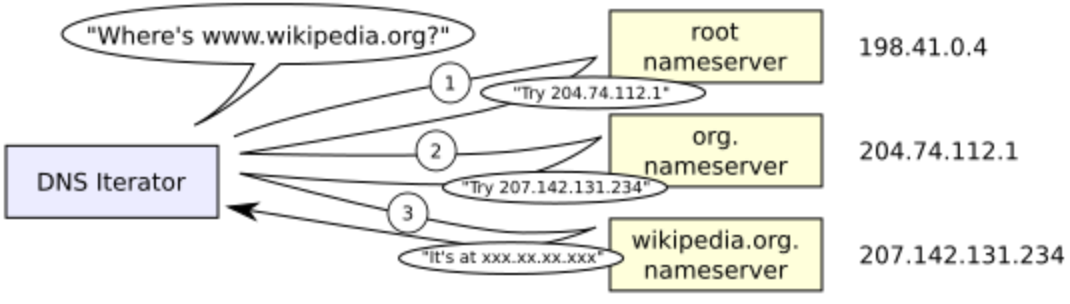
# Resolving Names

# Name Resolution

- *Any* node on the IP network has a (local) Name Server registered, the one to contact *first*
  - *Windows: 'nslookup'*
  - *Linux: 'nslookup'* ☺

```
d:\proj\Book>nslookup
Default Server:  UnKnown
Address:  10.83.252.137
```

- Algorithm: "If I do not know, I know who knows"
  - Picks the name apart **right to left!**
    - org *before* wikipedia *before* www



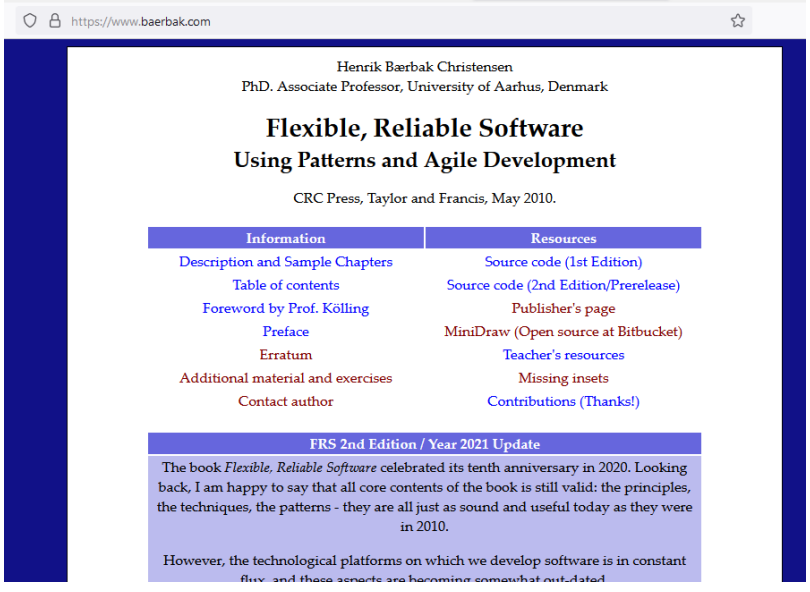
# You may trace the lookup path

- Windows 11

```
D:\proj\SWEA-E25\web>tracert hotstone.littleworld.dk

Tracing route to hotstone.littleworld.dk [157.180.79.190]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  mygateway.home [192.168.0.1]
  2    20 ms    14 ms    15 ms  10.116.18.1
  3     9 ms     9 ms     9 ms  ae13-200.aegnqe10.dk.ip.tdc.net [176.22.62.74]
  4    20 ms    20 ms    24 ms  ae10-0.asd3nqp1.nl.ip.tdc.net [83.88.1.219]
  5    21 ms    19 ms    20 ms  amsix-gw.hetzner.com [80.249.209.55]
  6    19 ms    19 ms    19 ms  core51.ams.hetzner.com [213.239.203.238]
  7    30 ms    32 ms    32 ms  core52.sto.hetzner.com [213.239.252.238]
  8    37 ms    38 ms    36 ms  core32.hel1.hetzner.com [213.239.254.65]
  9    38 ms    37 ms    38 ms  spine15.cloud1.hel1.hetzner.com [213.239.228.6]
 10    37 ms    54 ms    43 ms  spine3.cloud1.hel1.hetzner.com [213.239.228.30]
 11     *        *        *     Request timed out.
 12    37 ms    36 ms    38 ms  15509.your-cloud.host [95.216.134.199]
 13    39 ms    38 ms    38 ms  static.190.79.180.157.clients.your-server.de [157.180.79.190]

Trace complete.
```

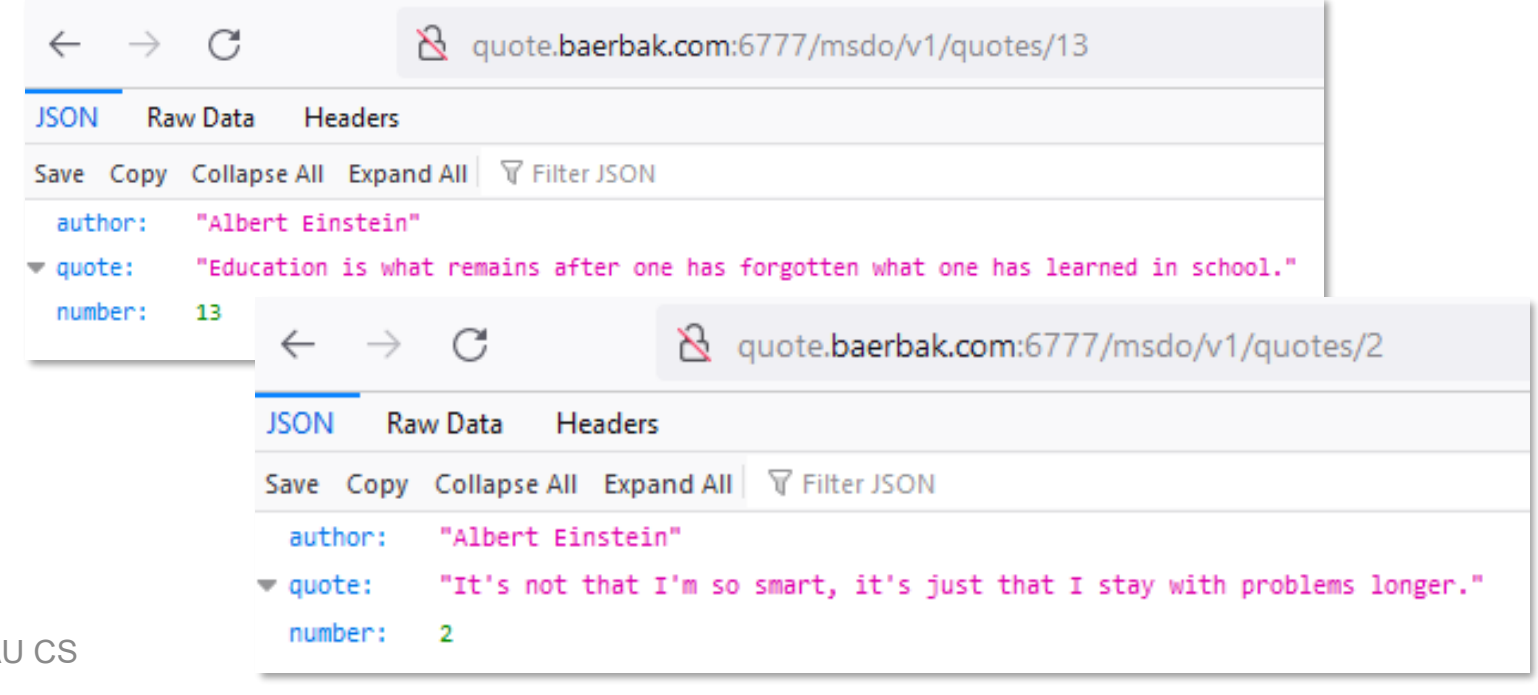
# So…

- baerbak.com will become
  - https://www.baerbak.com
- Firefox calls DNS server
  - Translate it into IP address
- Firefox will then send a http request to port 80 on that ip address (actually 443 as it is the ‘secure http’ port)
- … which will return a HTML document



# Or…

- I start my ‘quote service’ on a DigitalOcean machine in Amsterdam, on port 6777, assign ‘quote.baerbak.com’ to that IP, and can now get famous quotes in JSON format:



# Performance

- Of course, contacting 3-8 servers just to resolve a DNS entry is *extremely expensive*

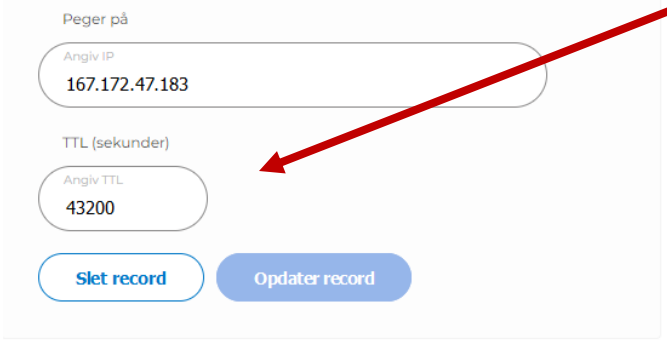
- **Caching** Tactic: 'Maintain multiple copies of data'
  - Each DNS server caches the lookup
    - So my local DNS server knows the address immediately the next time I ask
  - Browsers maintain their own caches!
    - No need to talk to the DNS at all after initial domain name has been resolved…

AARHUS UNIVERSITET

# Time To Live

- But but – what happens when IPs change then?
  - All the caches will send requests to the *old* node?

- The principle of **delegation** is used in DNS
  - I move my HotStone server to another provider – and get a new IP address
  - The DNS system has to adapt: **TTL: TimeToLive** here 12h



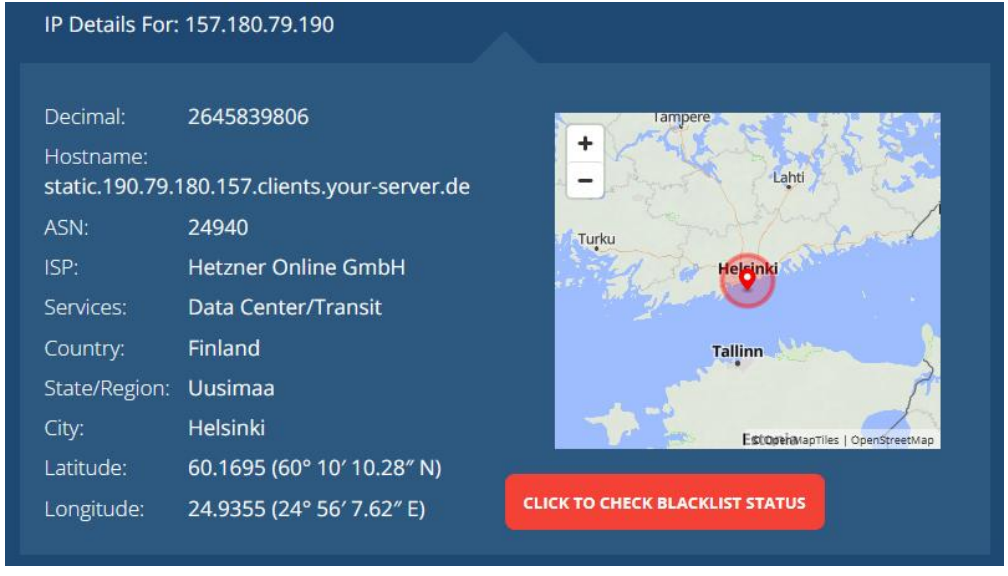

Takes up to 12 hours world-wide

AARHUS UNIVERSITET

# Fun Services

- DNS and IP are of course ‘public knowledge’, so there are quite a few fun services to lookup data out there.

```
D:\proj\SWEA-E25\web>ping hotstone.littleworld.dk

Pinging hotstone.littleworld.dk [157.180.79.190] with
```



AARHUS UNIVERSITET

# Summary

- To send a datagram, you have to know the *address* of the receiver
- Every node in an IP network has an **IP address**
  - IP address xxx.xxx.xxx.xxx (or IPv6)
- Nodes for a wider audience use **DNS** servers to assign a ***hostname*** to a specific IP address
  - www.dr.dk instead of xxx.xxx.xxx.xxx
- Every node has 65.536 **ports**
  - Quite a few below 1024 are reserved

AARHUS UNIVERSITET

# TCP

The last piece of the puzzle

AARHUS UNIVERSITET

# Actually, rather hidden

- IP splits data into packets/datagrams and sends them
  - But they get lost!
  - They become garbled
  - They arrive out-of-order
- TCP introduce **reliability**
  - Get packet 1, 2, 3, 5, ~~7~~, 6...
  - Request packet 4 again, and 7 as it was garbled
  - Forward the full data by putting segments in correct order

AARHUS UNIVERSITET

# Network Address Translation

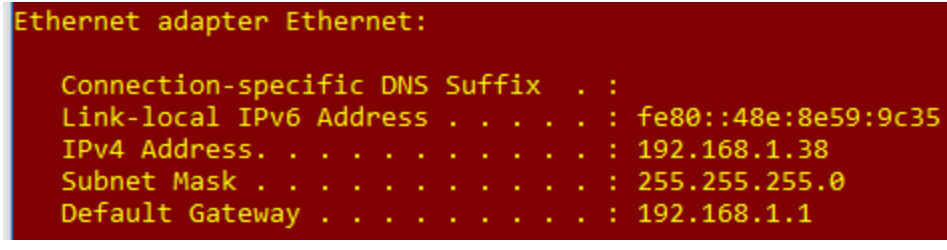
## Weird Behaviour Warning

AARHUS UNIVERSITET

# Segmenting Networks

- Organizations, projects, homes create their own LANs.
  - Security, convenience, performance

- Example:
  - My home router that assigns each connected node an IP in the 192.168.x.x space
    - But at any time there are thousands of machines with IP 192.168.1.38




  - How does ‘www.imhotep.dk’ know which computer to return the HTML document to, then???

# NAT

All IP packets have a source IP address and a destination IP address. Typically packets passing from the private network to the public network will have their source address modified, while packets passing from the public network back to the private network will have their destination address modified. To avoid ambiguity in how replies are translated, further modifications to the packets are required. The vast bulk of Internet traffic uses Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). For these protocols the port numbers are changed so that the combination of IP address and port information on the returned packet can be unambiguously mapped to the corresponding private network destination. RFC 2663 uses the term *network address and port translation* (NAPT) for this type of NAT. Other names include *port address translation* (PAT), *IP masquerading*, *NAT overload* and *many-to-one NAT*. This is the most common type of NAT and has become synonymous with the term "NAT" in common usage.

- So NAT in my router simply change IP:port of the datagrams so the web server returns to the router instead; once it has been received, the router forwards to the local node

AARHUS UNIVERSITET

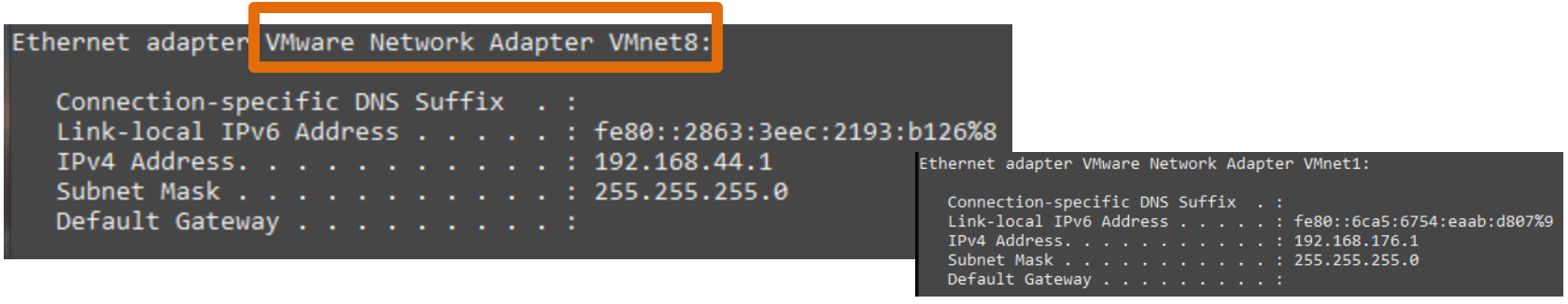
# Implications

- NAT makes networking behave ‘weird’:
  - *I can see you, but you cannot see me!*
  - My home computer can see the full internet, but no computer on the internet can see mine!
    - They can only see my ISP’s (Internet Service Provider) computer, which is the only one that can see my router, which is the only one who can see my computer!

AARHUS UNIVERSITET

# Implications

- VMWare Player does NAT between your host machine and the course VM, Mxx, you are running
  - It installs additional networks on the host, my Win machine




  - Therefore your host has multiple IP addresses, on multiple networks, used by multiple machines
    - Meaning host and VM can communicate on the 192.168.*.* network. ***Remember to use that for local testing!***

AARHUS UNIVERSITET

# So, in my VM

- Running my VM I can see it is also on the 192.168.*.* network

```
csdev@m101:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.44.131  netmask 255.255.255.0  broadcast 192.168.44.255
        inet6 fe80::20c:29ff:fee6:18a4  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:e6:18:a4  txqueuelen 1000  (Ethernet)
        RX packets 452596  bytes 680056387 (680.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 71364  bytes 4344351 (4.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- So they can talk using these IP addresses
  - Win 10 -> m101

```
d:\proj>ping 192.168.44.131

Pinging 192.168.44.131 with 32 bytes of data:
Reply from 192.168.44.131: bytes=32 time<1ms TTL=64
Reply from 192.168.44.131: bytes=32 time<1ms TTL=64
```

- m101 -> Win 10

```
csdev@m101:~$ ping 192.168.176.1
PING 192.168.176.1 (192.168.176.1) 56(84) bytes of data.
64 bytes from 192.168.176.1: icmp_seq=1 ttl=128 time=1.18 ms
```

AARHUS UNIVERSITET

# Example

- Starting my HotStone server on my m1 VM

```
ens33: flags=4163<UP,BROADCAST
        inet 192.168.109.129
```

```
csdev@m1:~/proj/hotstone$ gradle fullserver

> Task :solution:fullserver
2022-11-04T11:07:27.950+01:00 [INFO] hotstone.solution.server.main.MultipleGameServerHttp :: method=constructor, port=5220, eventSourceDB=null
2022-11-04T11:07:27.955+01:00 [INFO] hotstone.solution.lobby.GameLobby :: method=constructor, context=starting
```

- Then I can contact it on that IP address from the host:

# And…

- On EduRoam which most of you are on with your laptops when here at the department…
- … things may easily not work!
- … because EduRoam maintains **firewalls** which often deny any communication on non-standard ports
  - Only on port 80, 443, 22, and some others
    - And subject to changes all the time!

AARHUS UNIVERSITET

# Handy commands

# Some Nice Network Commands

- Debug 101
  - *Can my computer see the other computer???*
    - *'ping www.imhotep.dk'*
    - *'ping 192.168.1.37'*
- What is my IP?
  - Windows: ipconfig / linux: ifconfig

If not installed, issue
*sudo apt install net-tools*
to get it ☺

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::48e:8e59:9c35:9806%18
   IPv4 Address. . . . . . . . . . . : 192.168.1.38
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1
```

```
ens33     Link encap:Ethernet  HWaddr 00:0c:29:58:f5:c2
          inet addr:192.168.85.128  Bcast:192.168.85.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe58:f5c2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:83079 errors:0 dropped:0 overruns:0 frame:0
          TX packets:36489 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:94975896 (94.9 MB)  TX bytes:4403632 (4.4 MB)
```

# Summary

- The Distributed course will go into the more details
  - I think ☺
- Lot of concepts,  but not core curriculum in SWEA
- However, you will bump into some of these issues in the mandatory project on distribution...
  - Client and Server need to talk – using their IP addresses and ports…