

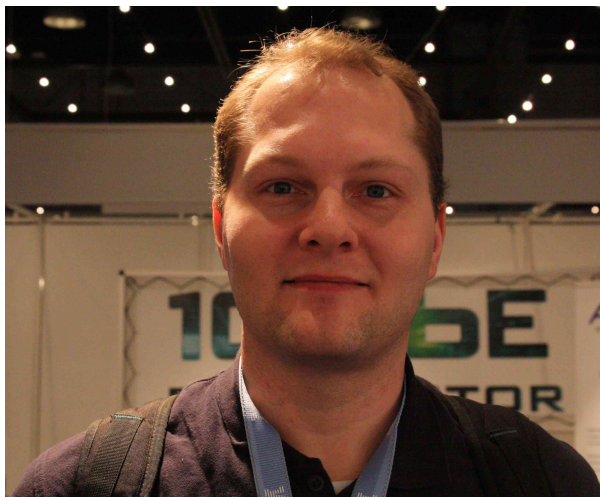
Welcome to

Introduktion til IPv6 workshop

August 2013

Henrik Lund Kramshøj, internet samurai
hlk@solido.net

`http://www.solidonetworks.com`



- Henrik Lund Kramshøj, IT-sikkerhed og internet samurai
- Email: hk@solido.net Mobil: +45 2026 6000
- Cand.scient fra Datalogisk Institut ved Københavns Universitet, DIKU
- CISSP certificeret
- 2003 - 2010 Selvstændig sikkerhedskonsulent
- 2010 Stifter og partner i Solido Networks ApS

Vi skal have glæde af hinanden i følgende kursusforløb

- 1 dag med workshop

I skal udover at lære en masse om protokoller og netværk

Forhåbentlig lærer i nogle gode vaner!

Jeres arbejde med netværk kan lettes betydeligt - hvis I starter rigtigt!



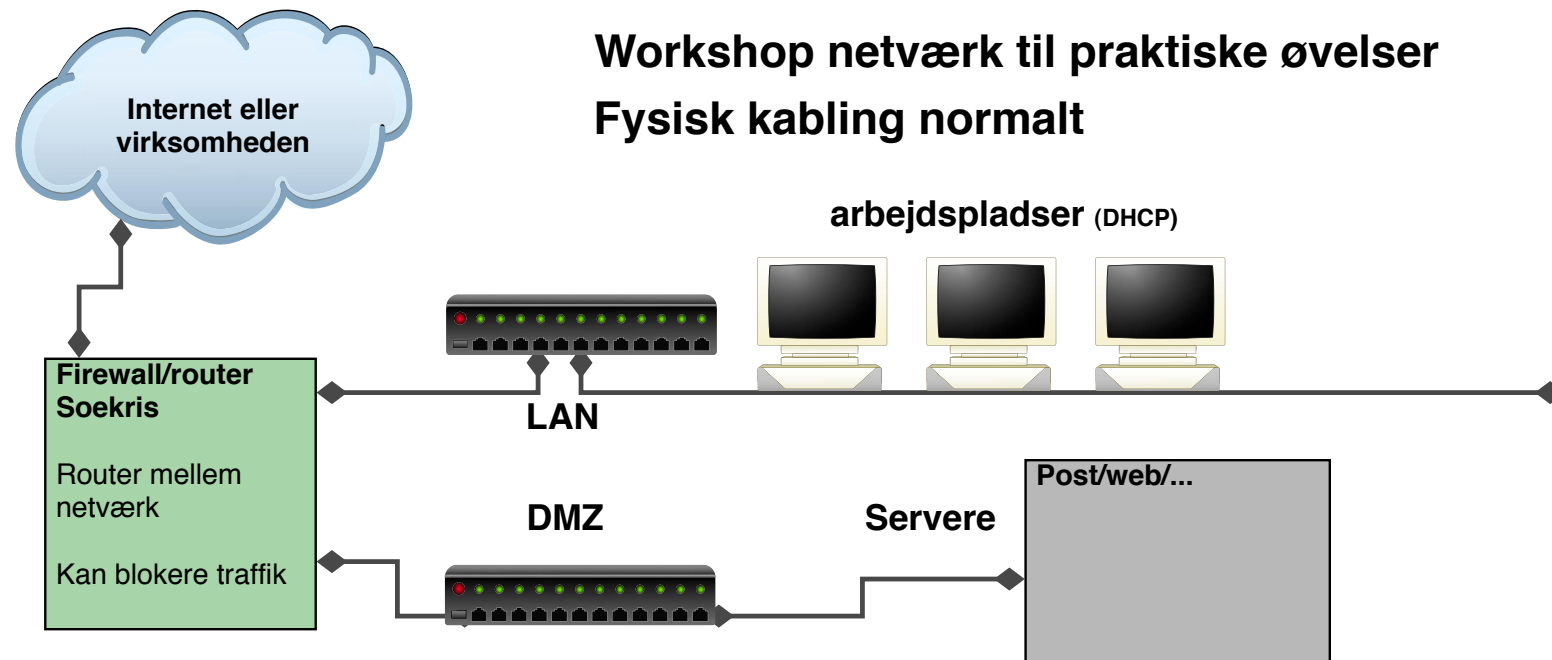
Free graphics by Lumen Design Studio

Dette materiale består af flere dele:

- Kursusmateriale - præsentationen til undervisning - dette sæt
- Øvelshæfte med øvelser

Hertil kommer diverse ressourcer fra internet og Kali Linux distribution

Bemærk: kursusmateriale er ikke en substitut for andet materiale, der er udeladt mange detaljer som forklares undervejs, eller kan slås op på internet



IP-baserede netværk - internet er overalt



At introducere IP netværk med IPv6

Kendskab til almindeligt brugte protokoller

- VLAN, DNS, ARP, NDP, TCP, UDP m.v.

Kendskab til almindelige værktøjer i disse miljøer

- ping, traceroute, Nping, iperf m.v.

NB: ja, jeg bruger en masse Unix og applikationer på Unix
**men de fleste af programmerne KAN installeres på Windows,
eller der kan findes alternativer der benytter samme protokoller!**

Dette er en workshop og fuldt udbytte kræver at deltagerne udfører praktiske øvelser

Kurset anvender Unix til øvelser, men forudgående Unix kendskab er ikke nødvendigt

De fleste øvelser kan udføres fra en Windows PC med virtuel maskine

Øvelserne foregår via

- Login til Unix maskinen
- Direkte fra jeres systemer Windows eller Linux boot CD/Virtuel maskine
- Via administrationsprogrammer, ofte webinterfaces

Der er opbygget et kursusnetværk med følgende primære systemer:

- Unix server Fiona med HTTP server og værktøjer
- Unix boot CD'er eller VMware images - jeres systemer
- WLAN "kamenet" med kode "henrik42"

På Fiona er oprettet kursusbrugere

- kursus1, kursus2, kursus3, ... kodeordet er **kursus**

Login: **kursus1**

Password: **kursus**

Det er en fordel at benytte hver sin bruger, så man kan gemme scripts

På de resterende systemer kan benyttes brugeren **kursus**

Login: **kursus**

Password: **kursus42** el **kursus**

The most advanced penetration testing distribution, ever.

From the creators of BackTrack comes Kali Linux, the most advanced and versatile penetration testing distribution ever created. BackTrack has grown far beyond its humble roots as a live CD and has now become a full-fledged operating system. With all this buzz, you might be asking yourself: - [What's new ?](#)



KALI LINUX
"the quieter you become, the more you are able to hear"

**PENETRATION TESTING,
REDEFINED.**

A Project By Offensive Security

BackTrack <http://www.backtrack-linux.org>

Kali <http://www.kali.org/>

Jeg benytter en del værktøjer i min dagligdag:

- Nmap, Nping - tester porte, godt til firewall admins <http://nmap.org>
- Metasploit Framework gratis på <http://www.metasploit.com/>
- Wireshark avanceret netværkssniffer - <http://http://www.wireshark.org/>
- Burpsuite <http://portswigger.net/burp/>
- Skipfish <http://code.google.com/p/skipfish/>
- OpenBSD operativsystem med fokus på sikkerhed <http://www.openbsd.org>

Vi når ikke at gennemgå alle disse

Der er mere end 300 værktøjer til rådighed på Kali/BackTrack

Stop - tid til check

Er alle kommet

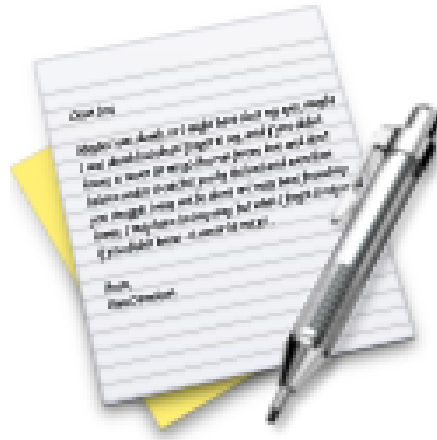
Har alle en PC med

Har alle et kabel eller trådløst netkort som virker

Der findes et trådløst netværk ved navn **kamenet** kode **henrik42**

Mangler der strømkabler

Mangler noget af ovenstående, sæt nogen igang med at finde det :-)



Vi laver nu øvelsen

Putty installation - Secure Shell login

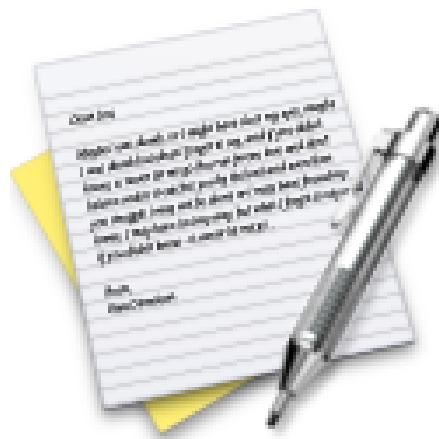
som er øvelse **1** fra øvelseshæftet.



Vi laver nu øvelsen

WinSCP installation - Secure Copy

som er øvelse 2 fra øvelseshæftet.



Vi laver nu øvelsen

Login på Unix systemerne

som er øvelse **3** fra øvelseshæftet.

Da Unix indgår er her et lille *cheat sheet* til Unix

- DOS/Windows kommando - tilsvarende Unix, og forklaring
- `dir` - `ls` - står for list files, viser filnavne
- `del` - `rm` - står for remove, sletter filer
- `cd` - `cd` - change directory, skifter katalog
- `type` - `cat` - concatenate, viser indholdet af tekstfiler
- `more` - `less` - viser tekstfiler en side af gangen
- `attrib` - `chmod` - change mode, ændrer rettighederne på filer

Prøv bare:

- **`ls`** list, eller long listing med **`ls -l`**
- **`cat /etc/hosts`** viser hosts filen
- **`chmod +x head.sh`** - sæt execute bit på en fil så den kan udføres som et program med kommandoen `./head.sh`

Opstart - hvad er IP og TCP/IP repetition, TCP, UDP, Subnets og CIDR

Basale værktøjer traceroute, ping, dig, host

Wireshark sniffer

SSL Secure Sockets Layer

VLAN 802.1q

Målinger nping, iperf, apache benchmark (ab)

Tuning og performancemålinger

Plus diverse webinterfaces og administrationsværktøjer

Adresseplaner og Infrastruktur i praksis - netværksdesign

Afslutning og opsummering på workshop

IPv4 Address Report

This report generated at 24-Jan-2012 07:59 UTC.

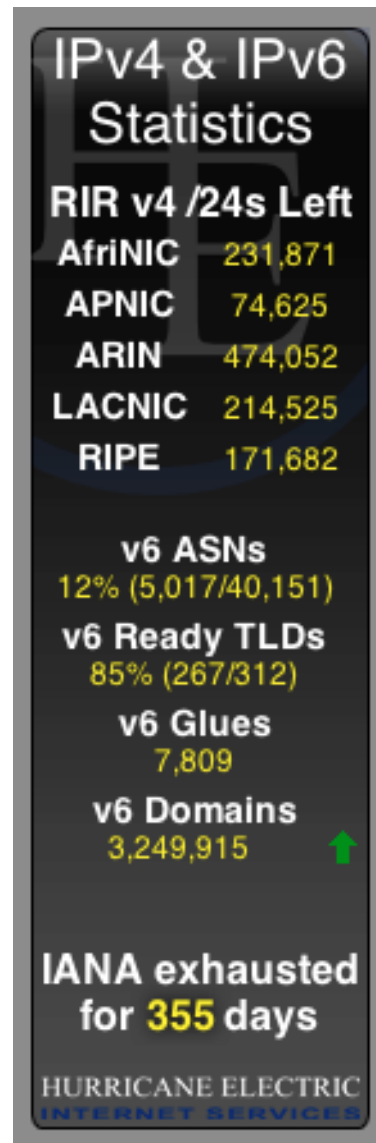
IANA Unallocated Address Pool
Exhaustion:

03-Feb-2011

Projected RIR Address Pool Exhaustion
Dates:

RIR	Projected Exhaustion Date	Remaining Addresses in RIR Pool (/8s)
APNIC:	19-Apr-2011	1.1990
RIPENCC:	27-Jul-2012	3.1711
ARIN:	19-Jul-2013	5.6671
LACNIC:	29-Jan-2014	3.8810
AFRINIC:	20-Oct-2014	4.3524

Kilde: <http://www.potaroo.net/tools/ipv4/>





An important consideration is that IPv6 is quite likely to be already running on the enterprise network, whether that implementation was planned or not. Some important characteristics of IPv6 include:

- IPv6 has a mechanism to automatically assign addresses so that end systems can easily establish communications.
- IPv6 has several mechanisms available to ease the integration of the protocol into the network.
- Automatic tunneling mechanisms can take advantage of the underlying IPv4 network and connect it to the IPv6 Internet.

Kilde:

http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6553/white_paper_c11-629391.html



For an IPv4 enterprise network, the existence of an IPv6 overlay network has several of implications:

- The IPv4 firewalls can be bypassed by the IPv6 traffic, and leave the security door wide open.
- Intrusion detection mechanisms not expecting IPv6 traffic may be confused and allow intrusion
- In some cases (for example, with the IPv6 transition technology known as 6to4), an internal PC can communicate directly with another internal PC and evade all intrusion protection and detection systems (IPS/IDS). Botnet command and control channels are known to use these kind of tunnels.

Kilde:

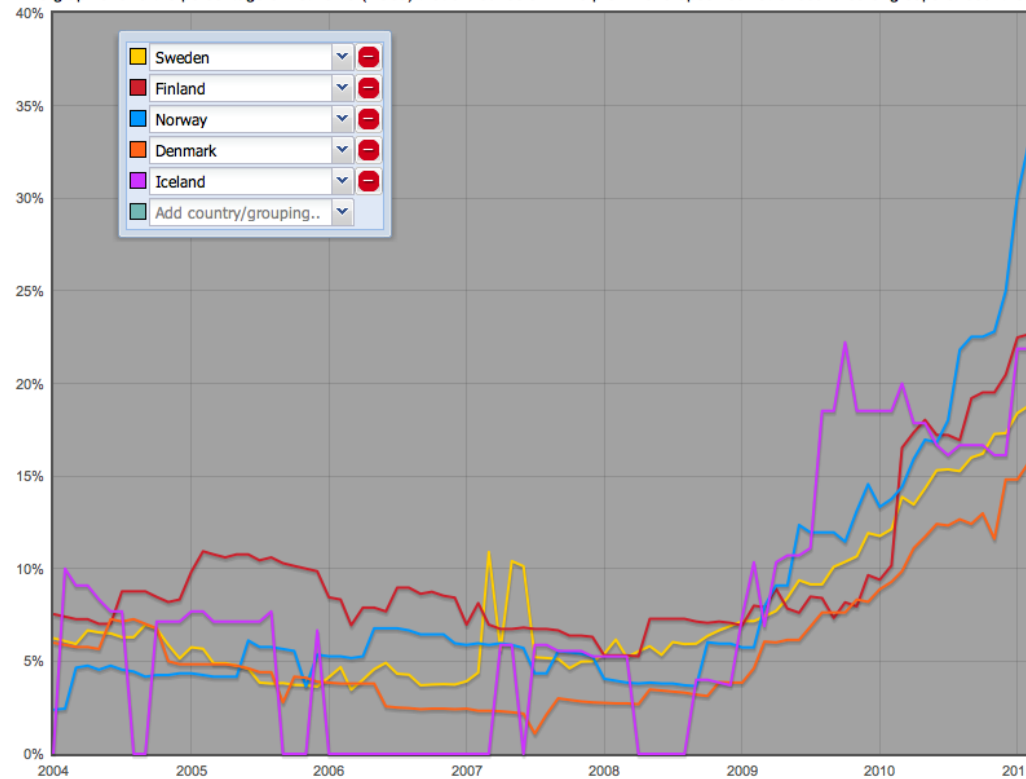
http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6553/white_paper_c11-629391.html



IPv6 Enabled Networks

permalink: <http://v6asns.ripe.net/v/6?s=SE;s=FI;s=NO;s=DK;s=IS>

This graph shows the percentage of networks (ASes) that announce an IPv6 prefix for a specified list of countries or groups of countries



http://v6asns.ripe.net/v/6?s=_ALL;s=DK;s=SE;s=NO;s=NL

Metasploit

Why Security Assessments Must Cover IPv6, Even In IPv4 Networks

Posted by [Christian Kirsch](#) in [Metasploit](#) on Mar 7, 2012 1:21:56 PM

What's your company doing to prepare for IPv6? Probably not an awful lot. While 10% of the world's top websites now offer IPv6 services, most companies haven't formulated an IPv6 strategy for the network. However, the issue is that most devices you have rolled out in the past 5 years have been IPv6-ready, if not IPv6-enabled. Windows 7 and Windows Server 2008 actually use IPv6 link-local addresses by default. Also think about all the other clients, servers, appliances, routers, and mobile devices you've added to your network in recent years. If you're honest, how do you know that your network is not vulnerable to IPv6 attacks right now?

That's why even if you haven't set up an IPv6 network internally yet, you should test for IPv6 vulnerabilities. Here are some common security issues that you may find:

- **Misconfiguration:** Not actively planning for IPv6 can introduce dangerous misconfiguration, such as a firewall that has filters set up for IPv4 traffic but accepts all IPv6 traffic. One organization we audited left zone transfers on their DNS server open for IPv6, but blocked for IPv4
- **Uneven features:** Many systems vendors are having to retrofit IPv6 into their products. Because Rome wasn't built in a day, IPv6 features often lag behind for a while. This uneven feature support for IPv6 can lead to security issues.
- **No IPv6 defenses:** Some defense mechanisms, such as older IPS systems, may simply be blind to IPv6 traffic, letting it pass through without scrutiny.

Metasploit can now conduct penetration tests on IPv6 networks to uncover these security issues, enabling you to find these issues:

IPv6-ENABLED SYSTEMS QUADRUPLED OVER THE LAST 3 YEARS
Source: Geoff Huston, RIPE, Rapid7

Year	Event
1992	ICM call for white papers
1996	RFC 1883 released, defining IPv6
2000	Microsoft releases IPv6 technology preview
2005	Linux IPv6 no longer "experimental"
2006	IPv6 is less than 1% of Internet traffic
2011	All major operating systems support IPv6
2012	Metasploit updated to identify IPv6 security risks

Kilde:

<https://community.rapid7.com/community/metasploit/blog/2012/03/07/>

NIST

**National Institute of
Standards and Technology**
U.S. Department of Commerce

Special Publication 800-119

Guidelines for the Secure Deployment of IPv6

SP 800-119 Dec. 2010 *Guidelines for the Secure Deployment of IPv6*
God introduktion til IPv6 og sikkerhed i forbindelse med IPv6

<http://csrc.nist.gov/publications/PubsSPs.html>

www.solidonetworks.com

hlik@solidonetworks.com

Really how to use IPv6?

Get IPv6 address and routing

Add AAAA (quad A) records to your DNS

Done

www.solidonetworks.com

WWW	IN A	91.102.95.20
	IN AAAA	2a02:9d0:10::9

- An almost unlimited scalability with a very large IPv6 address space (2^{128} addresses), enabling IP addresses to each and every device.
- Address self-configuration mechanisms, easing the deployment.
- Improved security and authentication features, such as mandatory IPSec capacities and the possibility to use of the address space to include encryption keys.
- Peer-to-peer connectivity, solving the NAT barrier with specific and permanent IP addresses for any device and/or user of the Internet.
- Mobility features, enabling a seamless connexion when moving from one access point to another access point on the Internet.
- Multi cast and any cast functionalities.
- IPv6 will provide an easier remote interaction with each and every device with a **direct integration to the Internet**. In other words, IPv6 will make possible to move from a network of servers, to a network of things.

Business case for IPv6 is **continuity**

Partial quote from <http://www.smartipv6building.org/index.php/en/ipv6-potential>

IPv6: Internet redesigned? - no!

Preserve the good stuff

back to basics, internet as it used to be!

route sharing - connection rely on end points, not intermediary NAT boxes

end-to-end transparency - you have an address and I have an address

Wants: bandwidth +10G, low latency/predictable latency, Quality of Service, Security

IPv6 is evolution, not revolution

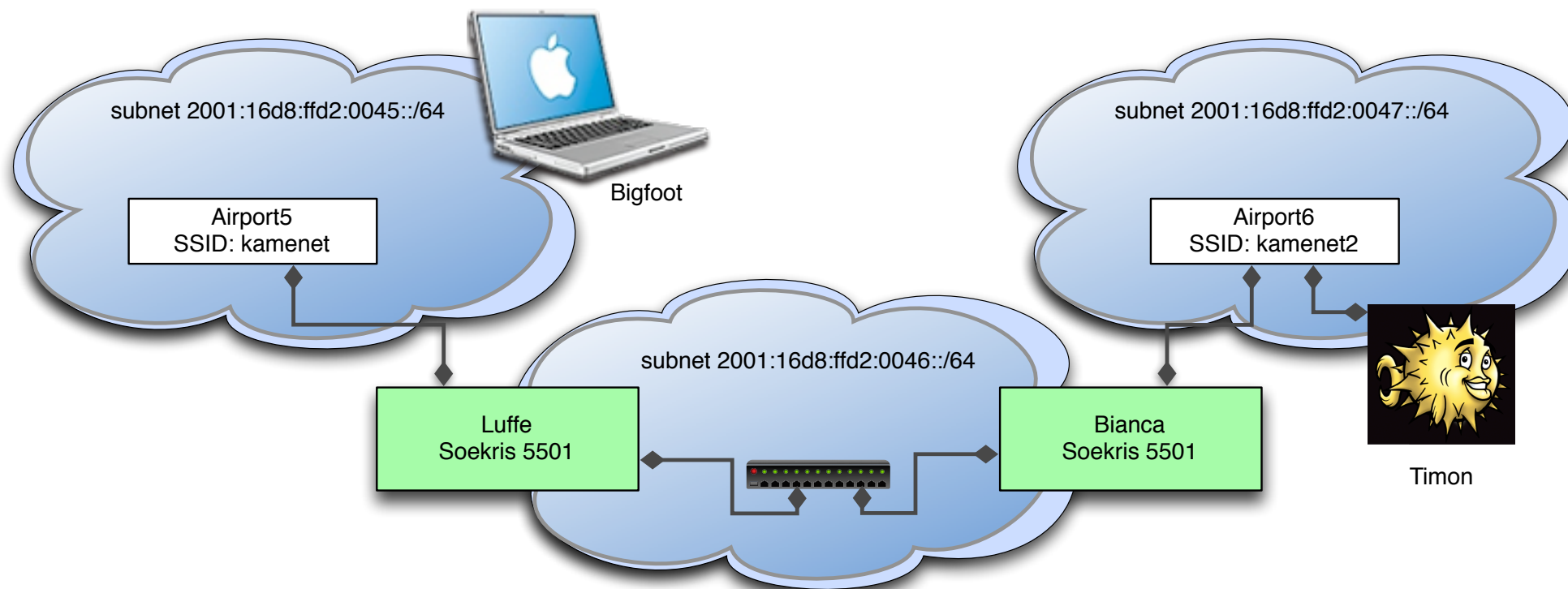
Note: IPv6 was not designed to solve all problems, so don't expect it to!

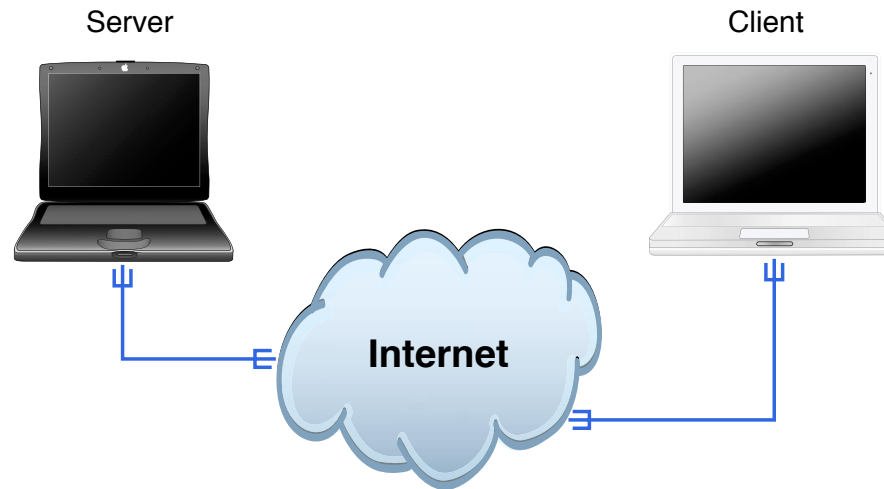
Use ping/ping6 and traceroute to test connectivity

Try in your browser:

- <http://www.kame.net> Dancing turtle
- <http://www.ripe.net> RIPE, look for address up right corner
- <http://loopsofzen.co.uk/> Play a game
- <https://www.sixxs.net/> Apply for IPv6 tunnel

Done 😊





Klienter og servere

Rødder i akademiske miljøer

Protokoller der er op til 20 år gamle

Meget lidt kryptering, mest på http til brug ved e-handel

Kurset omhandler udelukkende netværk baseret på IP protokollerne

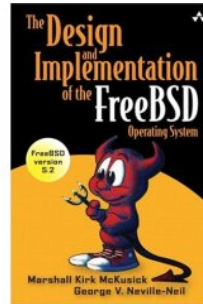
Kommunikation mellem mennesker!

Baseret på TCP/IP

- best effort
- packet switching (IPv6 kalder det packets, ikke datagram)
- forbindelsesorienteret, *connection-oriented*
- forbindelsesløs, *connection-less*

RFC-1958:

A good analogy for the development of the Internet is that of constantly renewing the individual streets and buildings of a city, rather than razing the city and rebuilding it. The architectural principles therefore aim to provide a framework for creating cooperation and standards, as a small "spanning set" of rules that generates a large, varied and evolving space of technology.



På Berkeley Universitetet blev der udviklet en del på Unix og det har givet anledning til en hel gren kaldet BSD Unix, BSD står for Berkeley Software Distribution

BSD Unix har blandt andet resulteret i virtual memory management og en masse TCP/IP relaterede applikationer

Specielt har BSD TCP/IP kernefunktionalitet været genbrugt mange steder

Tilsvarende genbruges KAME IPv6 implementationen mange steder

<http://en.wikipedia.org/wiki/BSD>



`http://www.kame.net`

- Er idag at betragte som en reference implementation
- i stil med BSD fra Berkeley var det
- KAME har været på forkant med implementation af draft dokumenter
- KAME er inkluderet i OpenBSD, NetBSD, FreeBSD og BSD/OS - har været det siden version 2.7, 1.5, 4.0 og 4.2
- Projektet er afsluttet, men nye projekter fortsætter i WIDE regi `http://www.wide.ad.jp/`
- Der er udkommet to bøger som i detaljer gennemgår IPv6 protokollerne i KAME

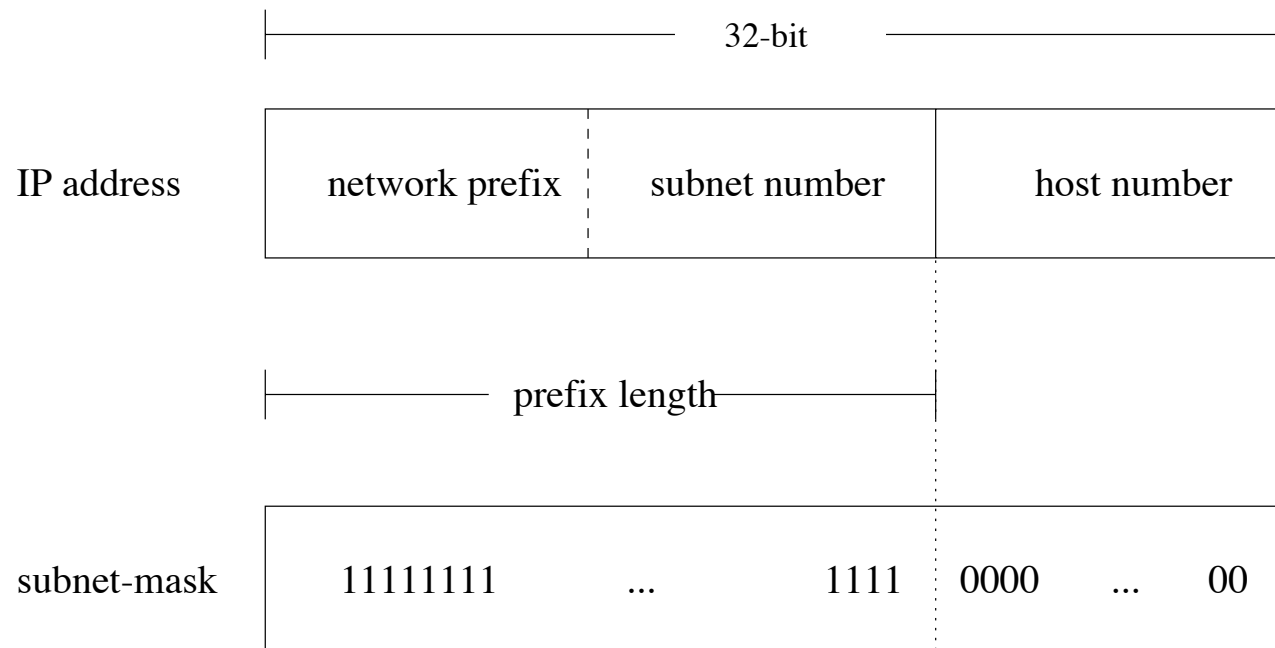
80'erne IP/TCP starten af 80'erne

90'erne IP version 6 udarbejdes

- IPv6 ikke brugt i Europa og US
- IPv6 er ekstremt vigtigt i Asien
- historisk få adresser tildelt til 3.verdenslande
- Større Universiteter i USA har ofte større allokering end Kina!

1991 WWW "opfindes" af Tim Berners-Lee hos CERN

E-mail var hovedparten af trafik - siden overtog web/http førstepladsen



Hvad kendetegner internet idag

Der er et fælles adresserum baseret på 32-bit adresser

En IP-adresse kunne være 10.0.0.1

```
hlk@bigfoot:hlk$ ipconvert.pl 127.0.0.1
Adressen er: 127.0.0.1
Adressen er: 2130706433
hlk@bigfoot:hlk$ ping 2130706433
PING 2130706433 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.135 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.144 ms
```

IP-adresser skrives typisk som decimaltal adskilt af punktum

Kaldes **dot notation**: 10.1.2.3

Kan også skrive som oktal eller heksadecimale tal

IP-adresse: 127.0.0.1

Heltal: 2130706433

Binary: 111111100000000000000000000000000000000001

IP-adresser kan også konverteres til bits

Computeren regner binært, vi bruger dot-notationen

Tidligere benyttede man klasseinddelingen af IP-adresser: A, B, C, D og E

Desværre var denne opdeling ufleksibel:

- A-klasse kunne potentielt indeholde 16 millioner hosts
- B-klasse kunne potentielt indeholder omkring 65.000 hosts
- C-klasse kunne indeholde omkring 250 hosts

Derfor bad de fleste om adresser i B-klasser - så de var ved at løbe tør!

D-klasse benyttes til multicast

E-klasse er blot reserveret

Se evt. http://en.wikipedia.org/wiki/Classful_network

Classfull routing		Classless routing (CIDR)	
4 Class C networks	Inherent subnet mask	Supernet	Subnet mask
192.0.08.0	255.255.255.0	192.0.08.0 Base network/prefix 192.0.8.0/	255.255.252.0 (252d=11111100b)
192.0.09.0	255.255.255.0		
192.0.10.0	255.255.255.0		
192.0.11.0	255.255.255.0		

Subnetmasker var oprindeligt indforstået

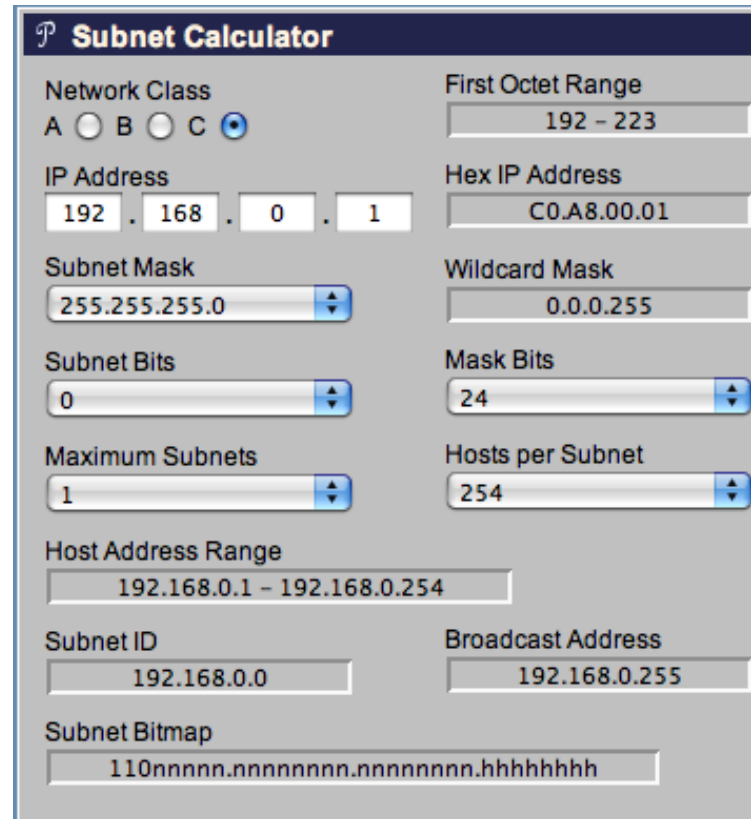
Dernæst var det noget man brugte til at opdele sit A, B eller C net med

Ved at tildele flere C-klasser kunne man spare de resterende B-klasser - men det betød en routing table explosion

Idag er subnetmaske en sammenhængende række 1-bit der angiver størrelse på nettet

10.0.0.0/24 betyder netværket 10.0.0.0 med subnetmaske 255.255.255.0

Nogle få steder kaldes det tillige supernet, supernetting



The screenshot shows a web-based 'Subnet Calculator' interface. It features a dark blue header with a 'P' icon and the title 'Subnet Calculator'. The interface is organized into two columns of input fields and output displays. On the left, there are radio buttons for 'Network Class' (A, B, C), with 'C' selected. Below this are input fields for 'IP Address' (192.168.0.1), 'Subnet Mask' (255.255.255.0), 'Subnet Bits' (0), and 'Maximum Subnets' (1). On the right, there are input fields for 'First Octet Range' (192 - 223), 'Hex IP Address' (C0.A8.00.01), 'Wildcard Mask' (0.0.0.255), 'Mask Bits' (24), and 'Hosts per Subnet' (254). At the bottom, there are two rows of output fields: 'Host Address Range' (192.168.0.1 - 192.168.0.254), 'Subnet ID' (192.168.0.0), 'Broadcast Address' (192.168.0.255), and 'Subnet Bitmap' (110nnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh). The 'n' characters represent network bits and 'h' characters represent host bits.

Der findes programmer som kan hjælpe med at udregne subnetmasker til IPv4

Screenshot fra <http://www.subnet-calculator.com/>

Der findes et antal adresserum som alle må benytte frit:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Address Allocation for Private Internets RFC-1918 adresserne!

NB: man må ikke sende pakker ud på internet med disse som afsender, giver ikke mening

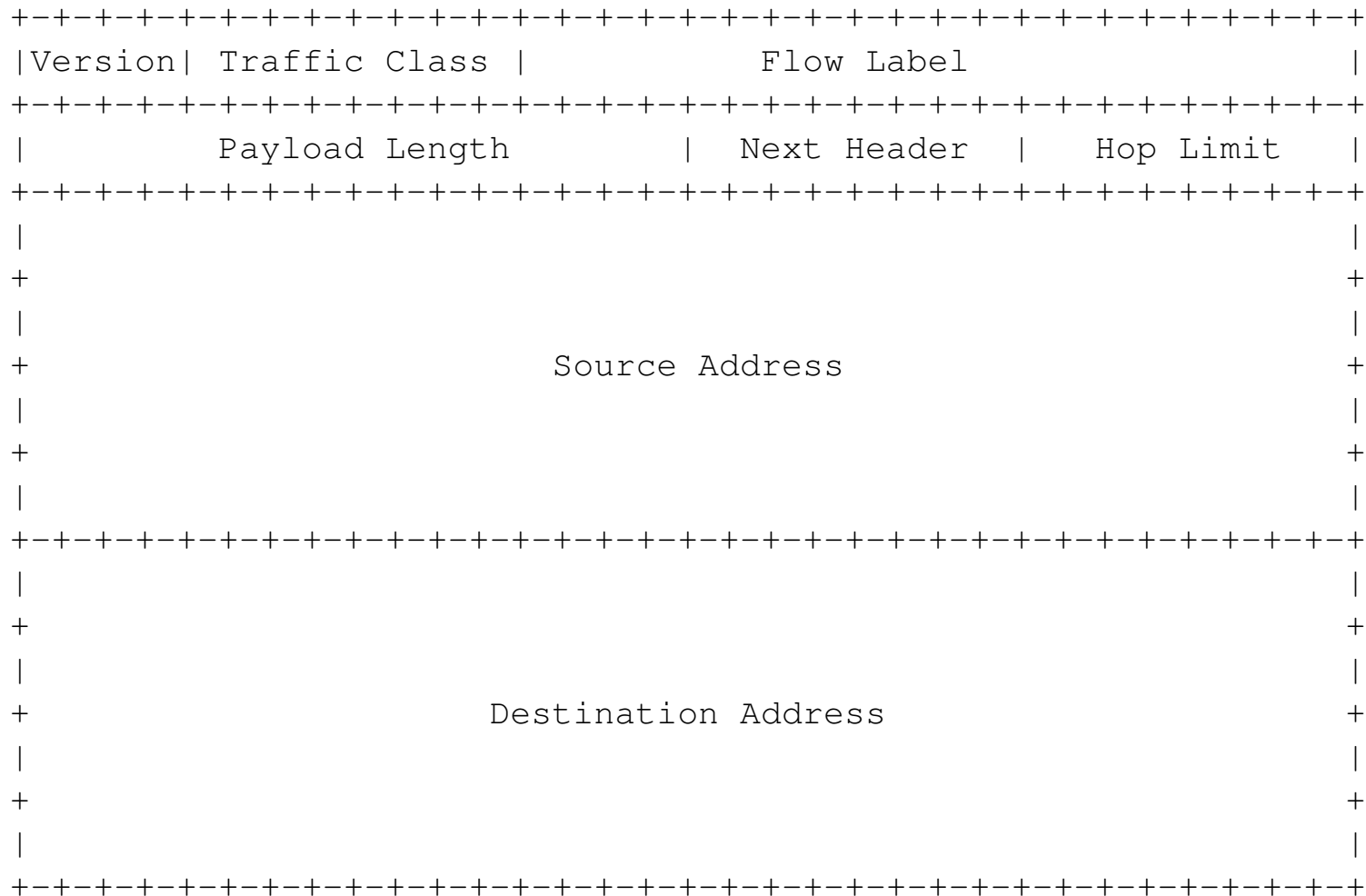
- Altid 32-bit adresser
- Skrives typisk med 4 decimaltal dot notation 10.1.2.3
- Netværk angives med CIDR Classless Inter-Domain Routing RFC-1519
- CIDR notation 10.0.0.0/8 - fremfor 10.0.0.0 med subnet maske 255.0.0.0
- Specielle adresser
 - 127.0.0.1 localhost/loopback
 - 0.0.0.0 default route
- RFC-1918 angiver private adresser som alle kan bruge

subnet prefix	interface identifier
---------------	----------------------

2001:16d8:ff00:012f:0000:0000:0000:0002
2001:16d8:ff00:12f::2

- 128-bit adresser, subnet prefix næsten altid 64-bit
- skrives i grupper af 4 hexcifre ad gangen adskilt af kolon :
- foranstillede 0 i en gruppe kan udelades, en række 0 kan erstattes med ::
- dvs 0:0:0:0:0:0:0:0 er det samme som
0000:0000:0000:0000:0000:0000:0000:0000
- Dvs min webservers IPv6 adresse kan skrives som: 2001:16d8:ff00:12f::2
- Specielle adresser: ::1 localhost/loopback og :: default route
- Læs mere i RFC-3513

IPv6 header - RFC-2460



Addresses are always 128-bit identifiers for interfaces and sets of interfaces

Unicast: An identifier for a **single interface**.

A packet sent to a unicast address is delivered to the interface identified by that address.

Anycast: An identifier for a **set of interfaces** (typically belonging to different nodes).

A packet sent to an anycast address is **delivered to one** of the interfaces identified by that address (the "nearest" one, according to the routing protocols' measure of distance).

Multicast: An identifier for a **set of interfaces** (typically belonging to different nodes).

A packet sent to a multicast address is **delivered to all interfaces identified by that address**.

OSI Reference
Model

Application
Presentation
Session
Transport
Network
Link
Physical

Internet protocol suite

Applications HTTP, SMTP, FTP, SNMP,	NFS
	XDR
	RPC
TCP UDP	
IPv4	IPv6 ICMPv6 ICMP
ARP RARP	
MAC	
Ethernet token-ring ATM ...	

Bemærk hvilket netværk vi bruger idag

Primære server fiona har IP-adressen 10.0.45.36

Primære router conhome har IP-adressen 10.0.45.2 (og flere andre)

Hvis du kender til IP i forvejen så udforsk gerne på egen hånd netværket

Det er tilladt at logge ind på alle systemer, undtagen Henrik's laptop :-)

Det er forbudt at ændre IP-konfiguration og passwords

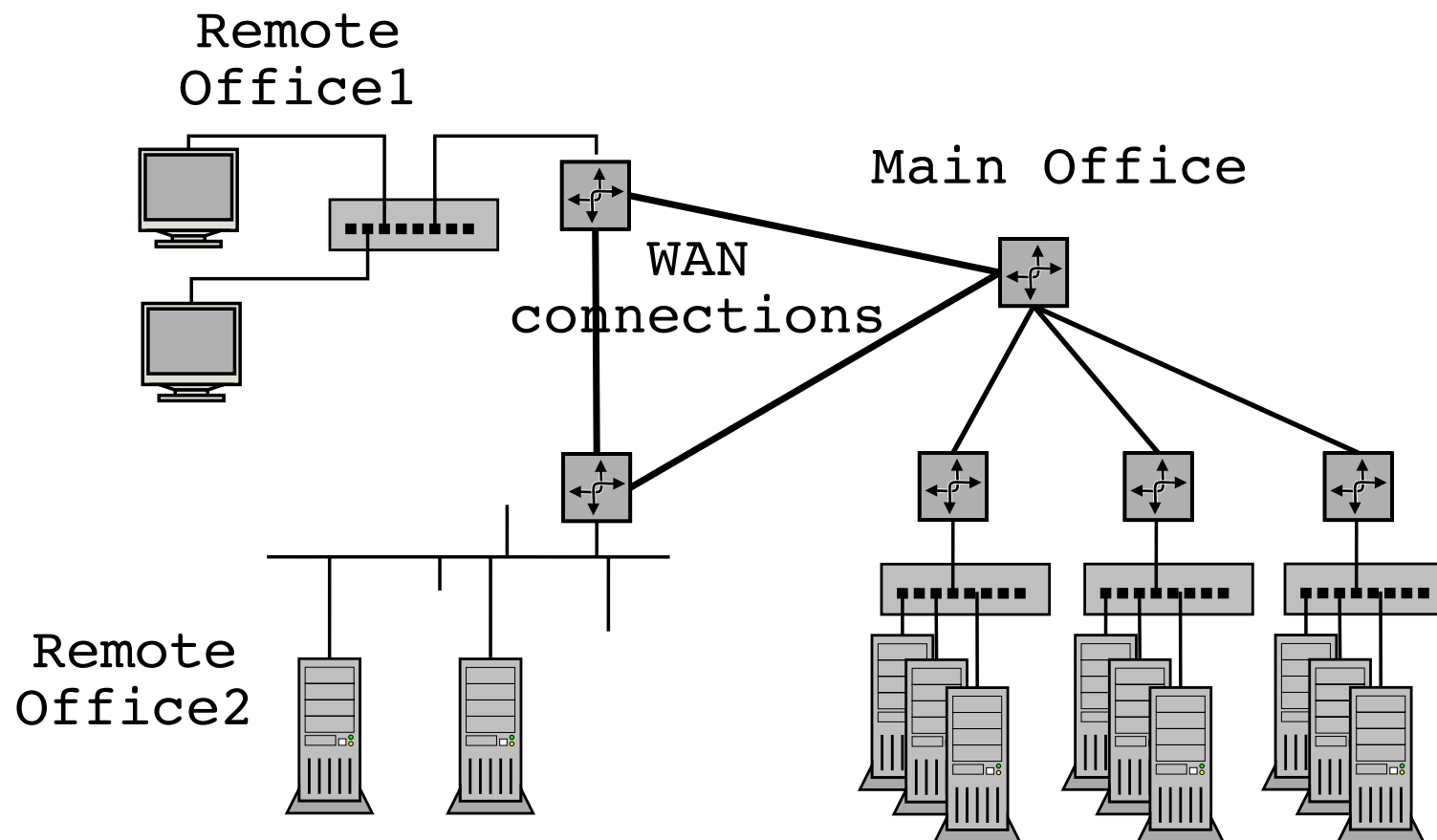
Nu burde I kunne forbinde jer til netværket fysisk, check med `ping 10.0.45.2`

Det er nok at en PC i hver gruppe er på kursusnetværket

Pause for dem hvor det virker, mens vi ordner resten



Et typisk 802.11 Access-Point (AP) der har Wireless og Ethernet stik/switch



Fysisk er der en begrænsning for hvor lange ledningerne må være

Ethernet er broadcast teknologi, hvor data sendes ud på et delt medie - Æteren

Broadcast giver en grænse for udbredningen vs hastighed

Ved hjælp af en bro kan man forbinde to netværkssegmenter på layer-2

Broen kopierer data mellem de to segmenter

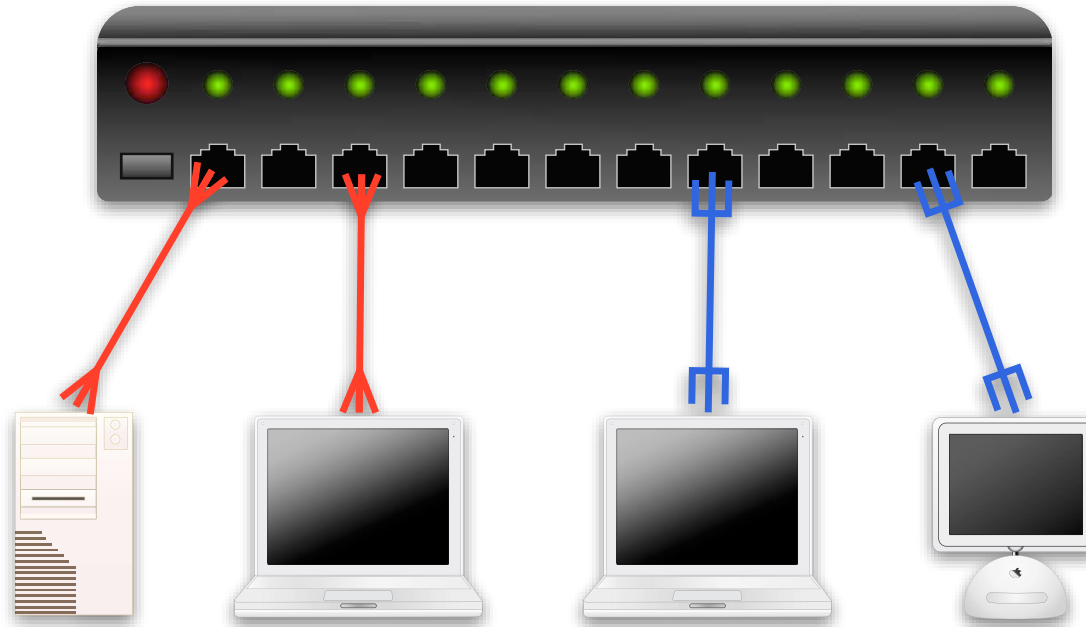
Virker som en forstærker på signalet, men mere intelligent

Den intelligente bro kender MAC adresserne på hver side

Broen kopierer kun hvis afsender og modtager er på hver sin side

Kilde: For mere information søg efter Aloha-net

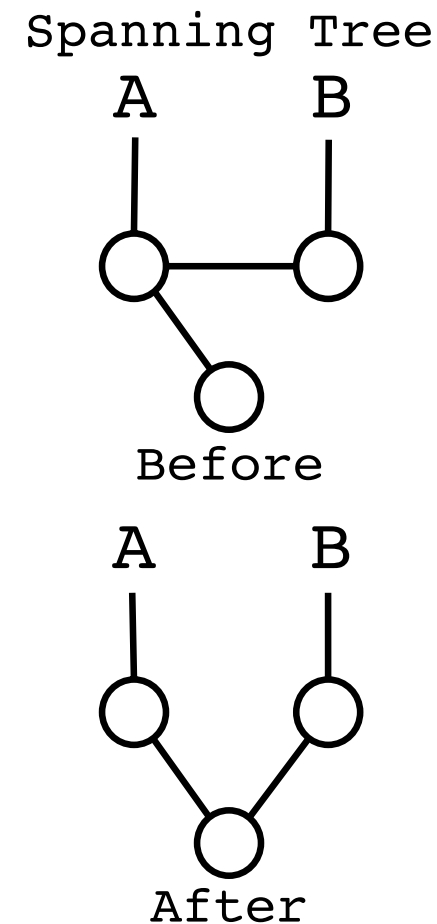
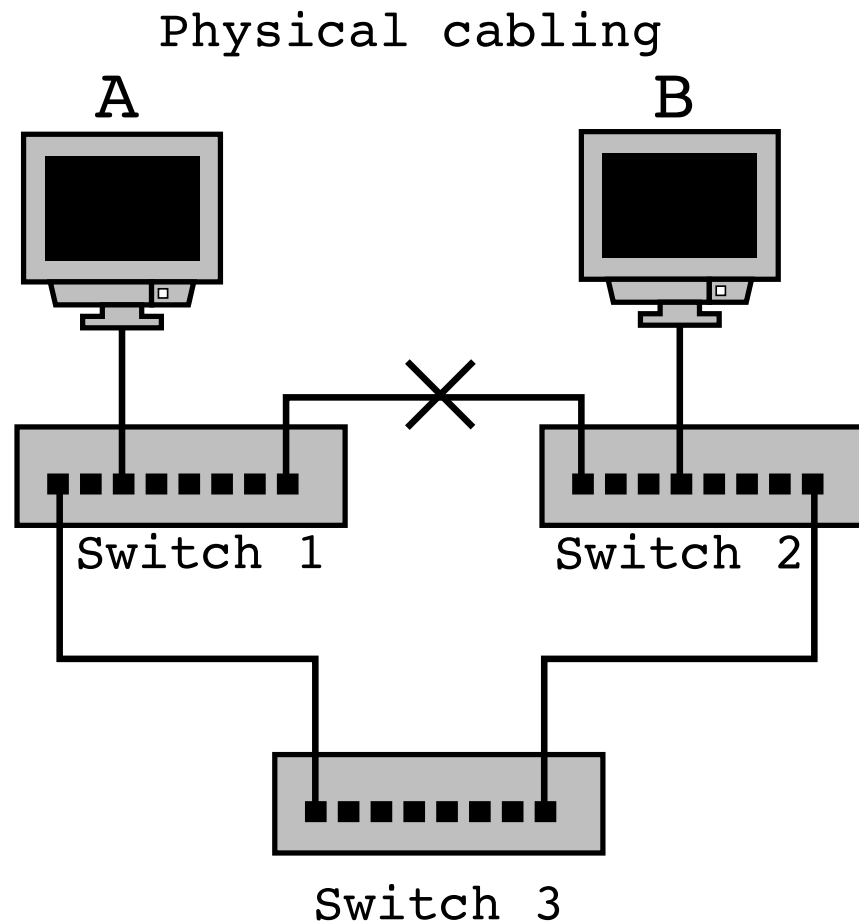
<http://en.wikipedia.org/wiki/ALOHAnet>



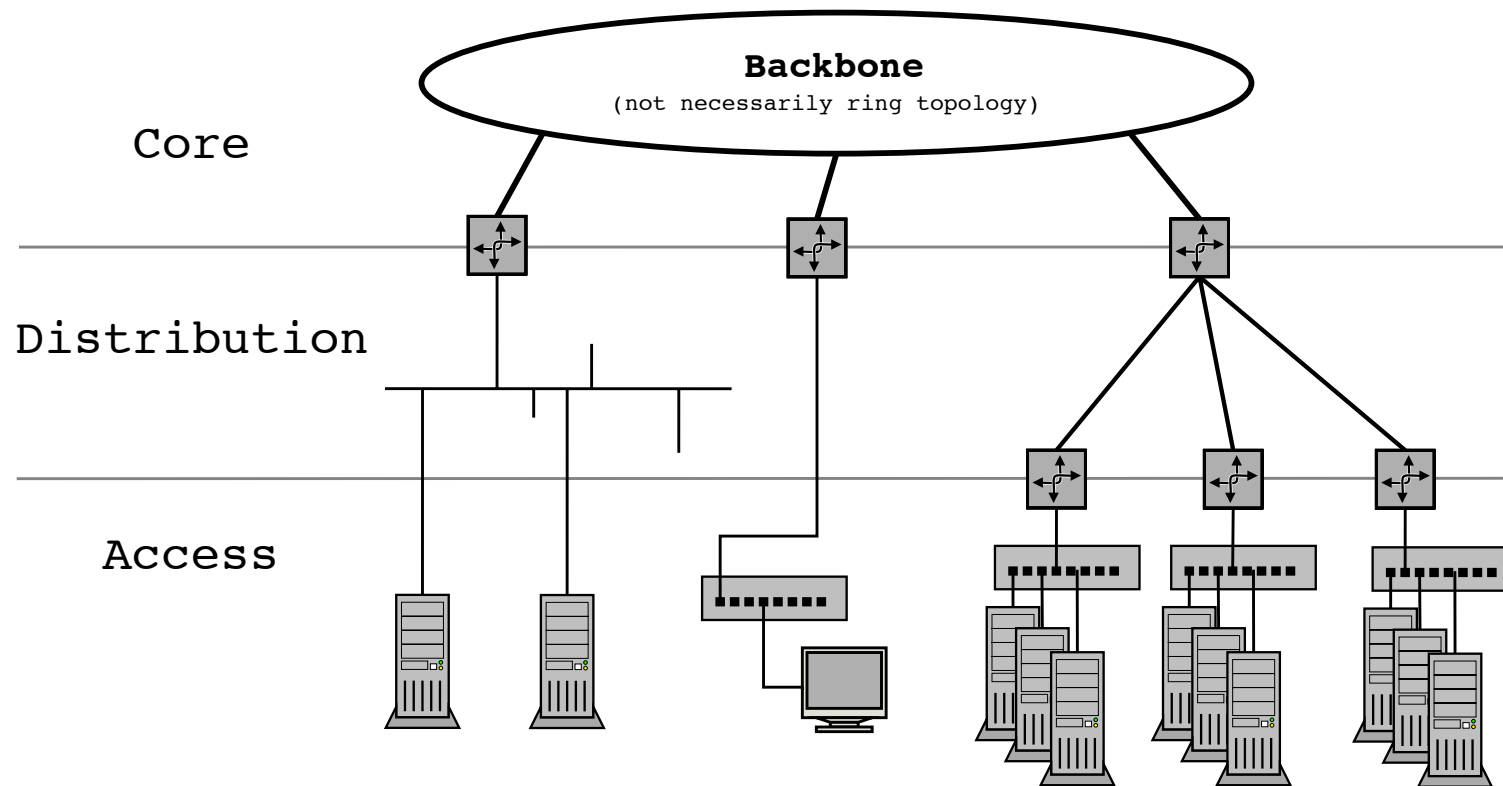
Ved at fortsætte udviklingen kunne man samle broer til en switch

En switch idag kan sende og modtage på flere porte samtidig, og med full-duplex

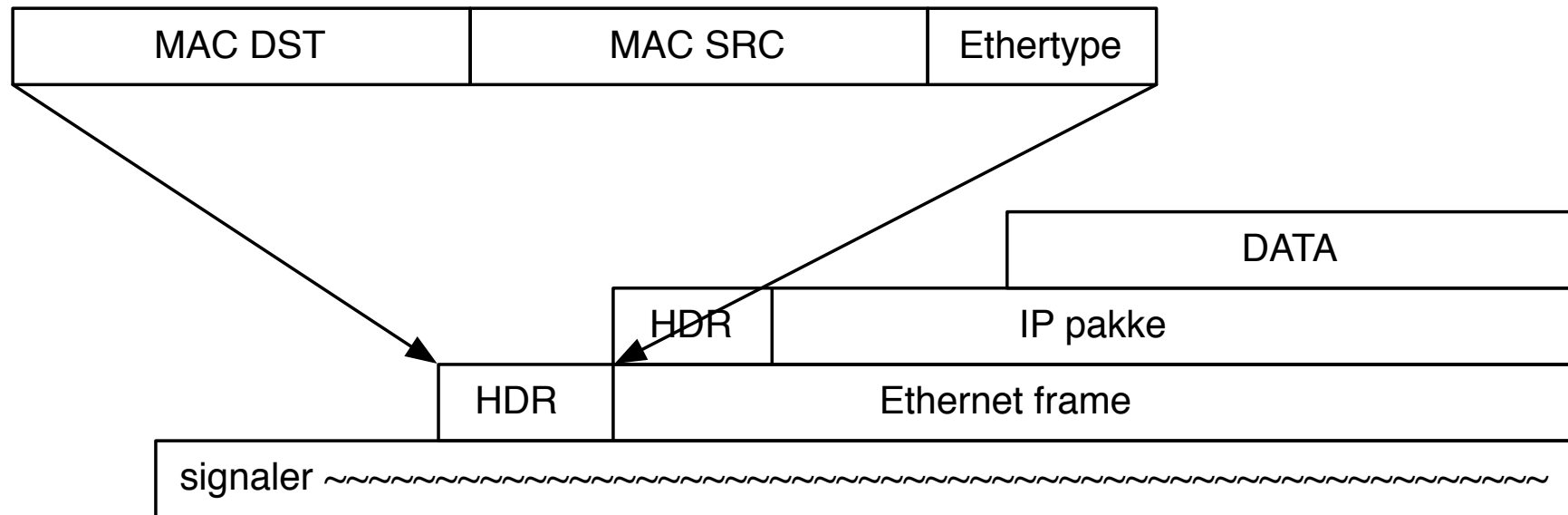
Bemærk performance begrænses af backplane i switchen



Se mere i bogen af Radia Perlman, *Interconnections: Bridges, Routers, Switches, and Internetworking Protocols*



Det er ikke altid man har præcis denne opdeling, men den er ofte brugt



Ser vi data som en datastrøm er pakkerne blot et mønster lagt henover data

Netværksteknologien definerer start og slut på en frame

Fra et lavere niveau modtager vi en pakke, eksempelvis 1500-bytes fra Ethernet driver

```
hlk$ ping 10.0.42.1
```

```
PING 10.0.42.1 (10.0.42.1): 56 data bytes
```

```
64 bytes from 10.0.42.1: icmp_seq=0 ttl=255 time=0.994 ms
```

```
64 bytes from 10.0.42.1: icmp_seq=1 ttl=255 time=1.060 ms
```

```
^C
```

```
--- 10.0.42.1 ping statistics ---
```

```
2 packets transmitted, 2 packets received, 0.0% packet loss
```

```
round-trip min/avg/max/stddev = 0.994/1.027/1.060/0.033 ms
```

```
hlk$ arp -an
```

```
? (10.0.42.1) at 0:0:24:ce:7c:9c on en1 ifscope [ethernet]
```

ARP cache kan vises med kommandoen `arp -an`

`-a` viser alle `-n` viser kun adresserne numerisk

ARP cache er dynamisk og adresser fjernes automatisk efter 5-20 minutter hvis de ikke bruges mere

Læs mere med `man 4 arp`

Routerne understøtter ofte Proxy ARP

Med Proxy ARP svarer de for en adresse bagved routeren

Derved kan man få trafik nemt igennem fra internet til adresser

Det er smart i visse situationer hvor en subnetting vil spilde for mange adresser

Hvis man kun har få adresser er subnetting måske heller ikke muligt

http://en.wikipedia.org/wiki/Proxy_ARP


```
$ ping6 -w -I en1 ff02::1
PING6(72=40+8+24 bytes) fe80::223:6cff:fe9a:f52c%en1 --> ff02::1
30 bytes from fe80::223:6cff:fe9a:f52c%en1: bigfoot
36 bytes from fe80::216:cbff:feac:1d9f%en1: mike.kramse.dk.
38 bytes from fe80::200:aaff:feab:9f06%en1: xrx0000aaab9f06
34 bytes from fe80::20d:93ff:fe4d:55fe%en1: harry.local
36 bytes from fe80::200:24ff:fec8:b24c%en1: kris.kramse.dk.
31 bytes from fe80::21b:63ff:fef5:38df%en1: airport5
32 bytes from fe80::216:cbff:fec4:403a%en1: main-base
44 bytes from fe80::217:f2ff:fee4:2156%en1: Base Station Koekken
35 bytes from fe80::21e:c2ff:feac:cd17%en1: arnold.local
```

```
hlk@bigfoot:basic-ipv6-new$ arp -an
? (10.0.42.1) at 0:0:24:c8:b2:4c on en1 [ethernet]
? (10.0.42.2) at 0:c0:b7:6c:19:b on en1 [ethernet]
hlk@bigfoot:basic-ipv6-new$ ndp -an
Neighbor                               Linklayer Address  Netif  Expire      St  Flgs  Prbs
::1                                   (incomplete)      lo0    permanent  R
2001:16d8:ffd2:cf0f:21c:b3ff:fec4:e1b6 0:1c:b3:c4:e1:b6  en1    permanent  R
fe80::1%lo0                          (incomplete)      lo0    permanent  R
fe80::200:24ff:fec8:b24c%en1 0:0:24:c8:b2:4c  en1    8h54m51s   S   R
fe80::21c:b3ff:fec4:e1b6%en1  0:1c:b3:c4:e1:b6  en1    permanent  R
```

NDP og ARP har samme funktion, binder IP og MAC sammen

ARP Address Resolution Protocol

IP og ICMP Internet Control Message Protocol

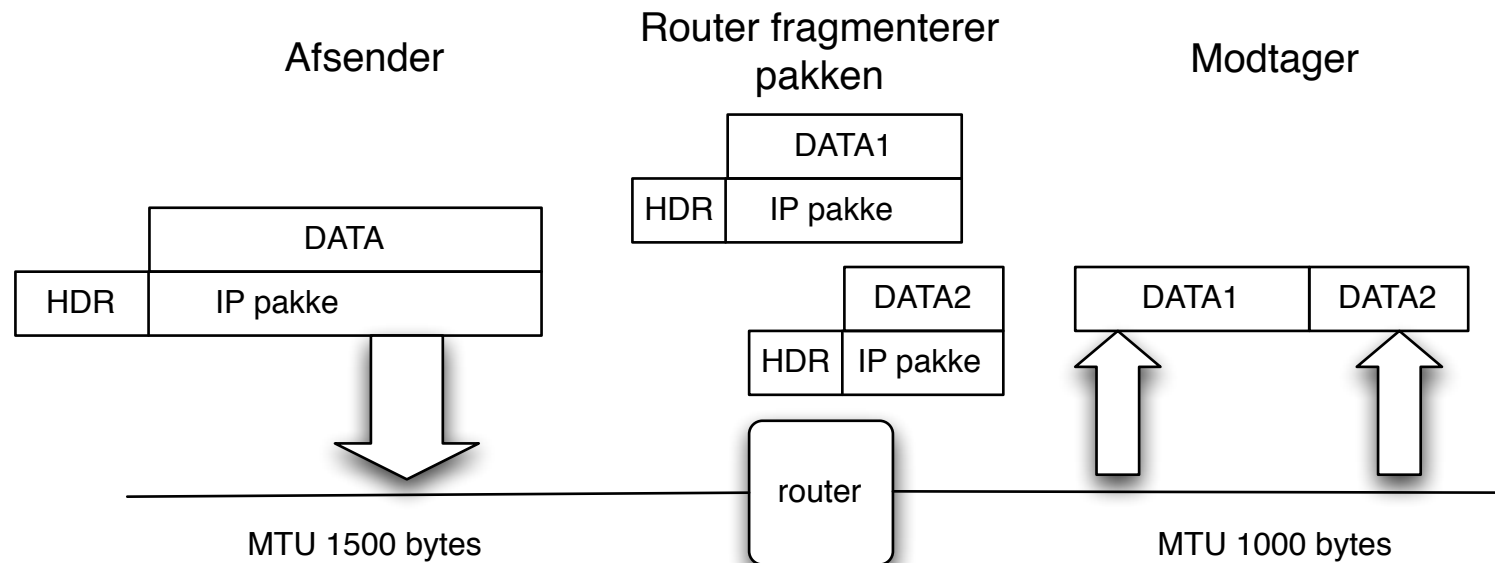
UDP User Datagram Protocol

TCP Transmission Control Protocol

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

Ovenstående er omtrent minimumskrav for at komme på internet



Hidtil har vi antaget at der blev brugt Ethernet med pakkestørrelse på 1500 bytes

Pakkestørrelsen kaldes MTU Maximum Transmission Unit

Skal der sendes mere data opdeles i pakker af denne størrelse, fra afsender

Men hvad hvis en router på vejen ikke bruger 1500 bytes, men kun 1000

Kontrolprotokol og fejlmeldinger

Nogle af de mest almindelige beskedtyper

- echo
- netmask
- info

Bruges generelt til *signalering*

Defineret i RFC-792

NB: nogle firewall-administratorer blokerer alt ICMP - det er forkert!

Type

- 0 = net unreachable;
- 1 = host unreachable;
- 2 = protocol unreachable;
- 3 = port unreachable;
- 4 = fragmentation needed and DF set;
- 5 = source route failed.

Ved at fjerne ALT ICMP fra et net fjerner man nødvendig funktionalitet!

Tillad ICMP types:

- 3 Destination Unreachable
- 4 Source Quench Message
- 11 Time Exceeded
- 12 Parameter Problem Message

ICMP - Internet Control Message Protocol

Benyttes til fejlbeskeder og til diagnosticering af forbindelser

ping programmet virker ved hjælp af ICMP ECHO request og forventer ICMP ECHO reply

```
$ ping 192.168.1.1
```

```
PING 192.168.1.1 (192.168.1.1): 56 data bytes  
64 bytes from 192.168.1.1: icmp_seq=0 ttl=150 time=8.849 ms  
64 bytes from 192.168.1.1: icmp_seq=1 ttl=150 time=0.588 ms  
64 bytes from 192.168.1.1: icmp_seq=2 ttl=150 time=0.553 ms
```

ping eller ping6

Nogle systemer vælger at ping kommandoen kan ping'e både IPv4 og Ipv6

Andre vælger at `ping` kun benyttes til IPv4, mens IPv6 ping kaldes for `ping6`

Læg også mærke til jargonen *at pinge*

traceroute programmet virker ved hjælp af TTL

levetiden for en pakke tælles ned i hver router på vejen og ved at sætte denne lavt opnår man at pakken *timer ud* - besked fra hver router på vejen

default er UDP pakker, men på Unix systemer er der ofte mulighed for at bruge ICMP

```
$ traceroute 217.157.20.129
```

```
traceroute to 217.157.20.129 (217.157.20.129),
```

```
30 hops max, 40 byte packets
```

```
1  safri (10.0.0.11)  3.577 ms  0.565 ms  0.323 ms
```

```
2  router (217.157.20.129)  1.481 ms  1.374 ms  1.261 ms
```

Husk at på Windows hedder kommandoen tracert

```
# tcpdump -i en0 host 217.157.20.129 or host 10.0.0.11
tcpdump: listening on en0
23:23:30.426342 10.0.0.200.33849 > router.33435: udp 12 [ttl 1]
23:23:30.426742 safri > 10.0.0.200: icmp: time exceeded in-transit
23:23:30.436069 10.0.0.200.33849 > router.33436: udp 12 [ttl 1]
23:23:30.436357 safri > 10.0.0.200: icmp: time exceeded in-transit
23:23:30.437117 10.0.0.200.33849 > router.33437: udp 12 [ttl 1]
23:23:30.437383 safri > 10.0.0.200: icmp: time exceeded in-transit
23:23:30.437574 10.0.0.200.33849 > router.33438: udp 12
23:23:30.438946 router > 10.0.0.200: icmp: router udp port 33438 unreachable
23:23:30.451319 10.0.0.200.33849 > router.33439: udp 12
23:23:30.452569 router > 10.0.0.200: icmp: router udp port 33439 unreachable
23:23:30.452813 10.0.0.200.33849 > router.33440: udp 12
23:23:30.454023 router > 10.0.0.200: icmp: router udp port 33440 unreachable
23:23:31.379102 10.0.0.200.49214 > safri.domain: 6646+ PTR?

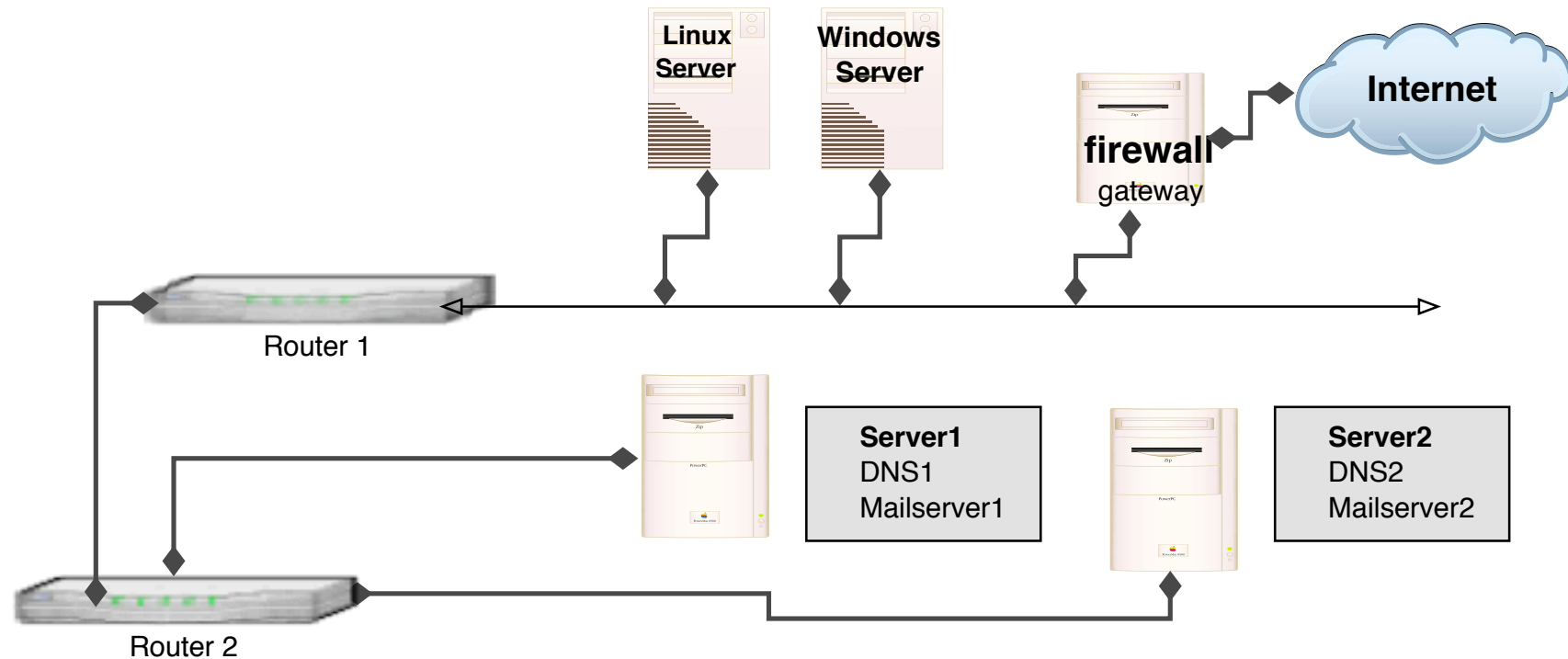
200.0.0.10.in-addr.arpa. (41)
23:23:31.380410 safri.domain > 10.0.0.200.49214: 6646 NXDomain* 0/1/0 (93)
14 packets received by filter
0 packets dropped by kernel
```

Diagnosticering af netværksproblemer - formålet med traceroute

Indblik i netværkets opbygning!

Svar fra hosts - en modtaget pakke fremfor et *sort hul*

Traceroute er ikke et angreb - det er også vigtigt at kunne genkende normal trafik!



Ved brug af traceroute og tilsvarende programmer kan man ofte udlede topologien i det netværk man undersøger

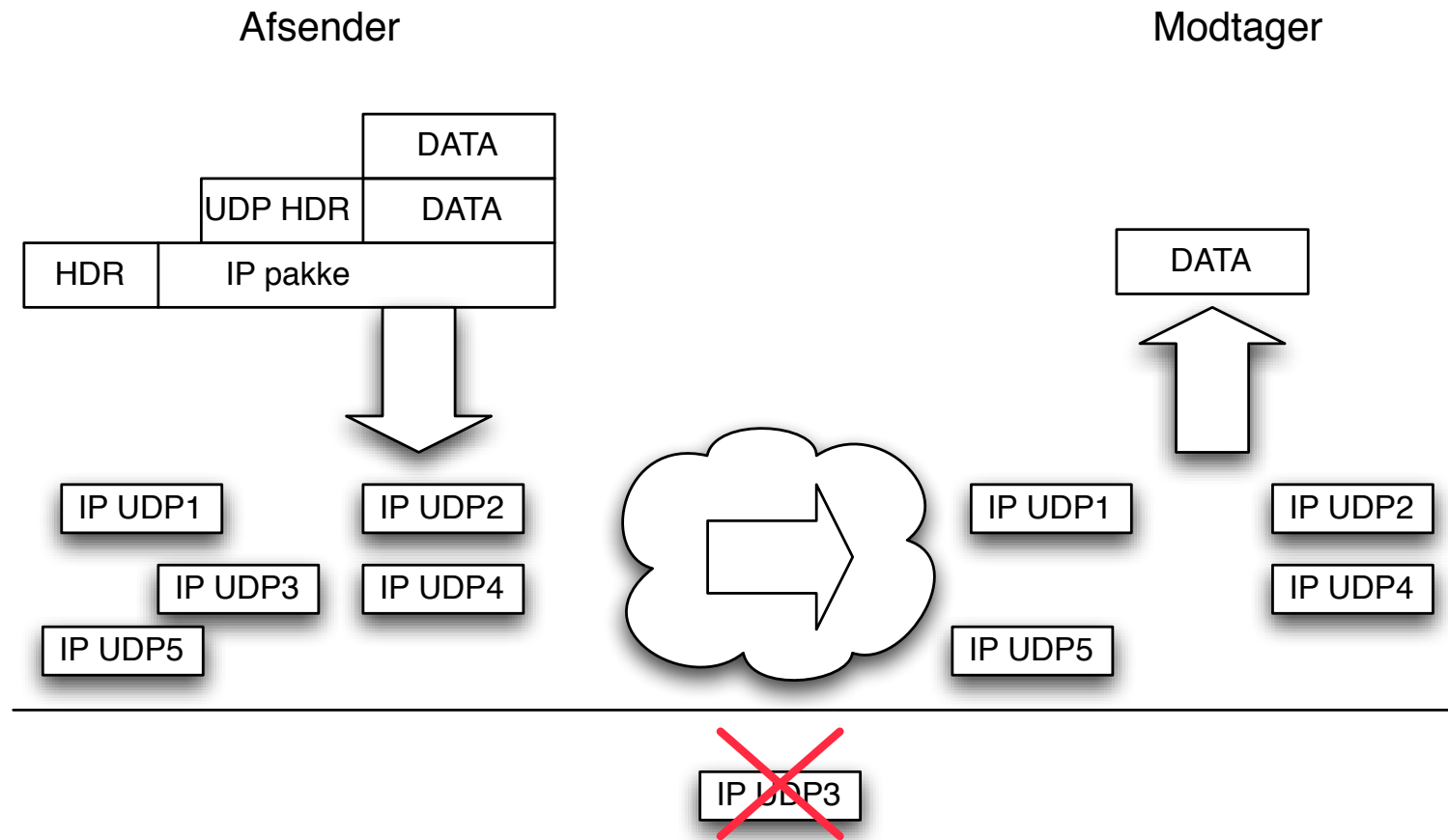
mtr My traceroute - grafisk <http://www.bitwizard.nl/mtr/>

lft - *layer four trace* benytter TCP SYN og FIN prober

trace ved hjælp af TCP og andre protokoller findes

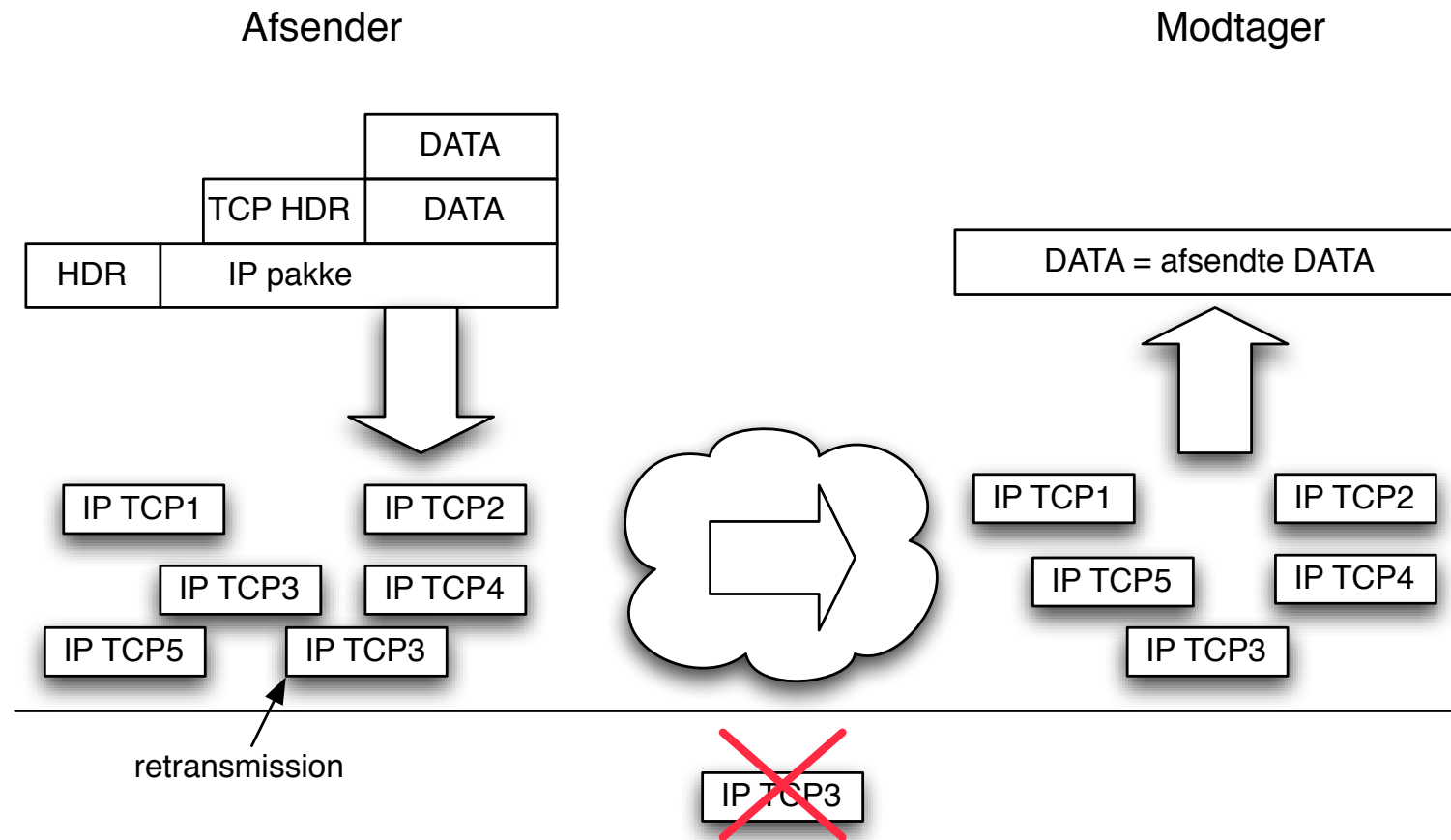
paratrace - *Parasitic Traceroute via Established TCP Flows and IPID Hopcount*

Der findes webservices hvor man kan trace fra, eksempelvis: <http://www.sampade.org>



Forbindelsesløs RFC-768, *connection-less* - der kan tabes pakker

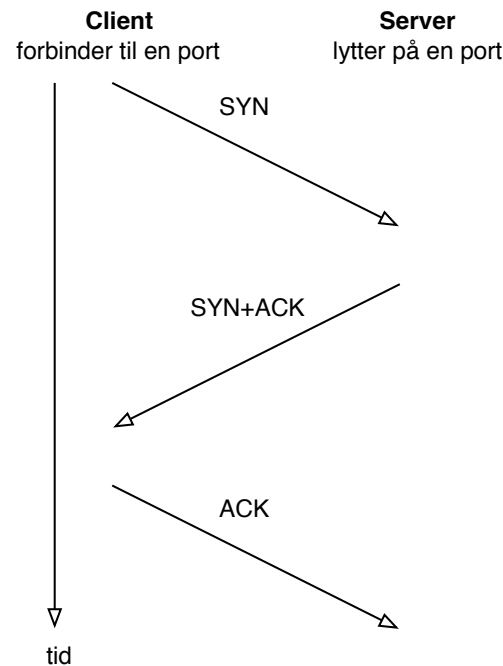
Kan benyttes til multicast/broadcast - flere modtagere



Forbindelsesorienteret RFC-791 September 1981, *connection-oriented*

Enten overføres data eller man får fejlmeddelelse

TCP three way handshake



- **TCP SYN half-open scans**
- Tidligere loggede systemer kun når der var etableret en fuld TCP forbindelse - dette kan/kunne udnyttes til *stealth*-scans
- Hvis en maskine modtager mange SYN pakker kan dette fylde tabellen over connections op - og derved afholde nye forbindelser fra at blive oprette - **SYN-flooding**



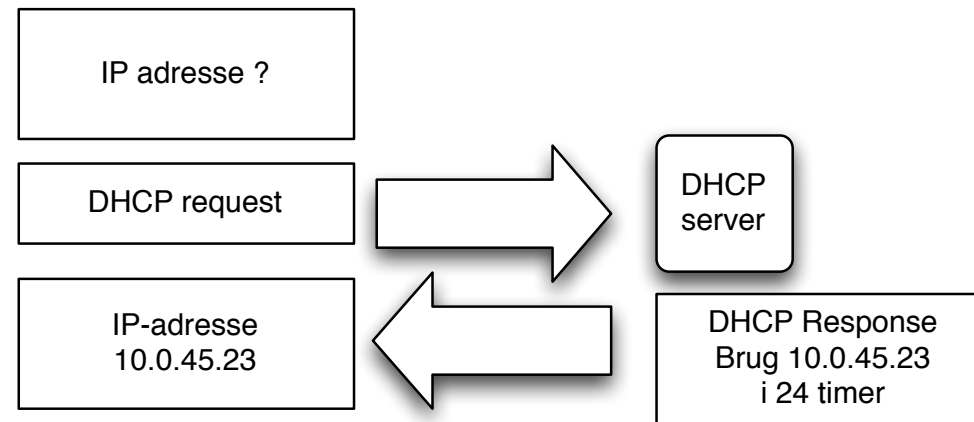
IANA vedligeholder en liste over magiske konstanter i IP

De har lister med hvilke protokoller har hvilke protokol ID m.v.

En liste af interesse er port numre, hvor et par eksempler er:

- Port 25 SMTP Simple Mail Transfer Protocol
- Port 53 DNS Domain Name System
- Port 67/68 DHCP
- Port 80 HTTP Hyper Text Transfer Protocol over TLS/SSL
- Port 443 HTTP over TLS/SSL

Se flere på `http://www.iana.org`



Hvordan får man information om default gateway

Man sender et DHCP request og modtager et svar fra en DHCP server

Dynamisk konfiguration af klienter fra en centralt konfigureret server

Bruges til IP adresser og meget mere

http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

```
/etc/rtadvd.conf:
en0:
    :addrs#1:addr="2001:1448:81:b00f::":prefixlen#64:
en1:
    :addrs#1:addr="2001:1448:81:beef::":prefixlen#64:

root# /usr/sbin/rtadvd -Df en0 en1
root# sysctl -w net.inet6.ip6.forwarding=1
net.inet6.ip6.forwarding: 0 -> 1
```

Stateless autoconfiguration er en stor ting i IPv6

Kommandoen starter den i debug-mode og i forgrunden
- normalt vil man starte den fra et script

Typisk skal forwarding aktiveres, som vist med BSD sysctl kommando

NB: de fleste clients vil idag implementere IPv6 privacy addresses

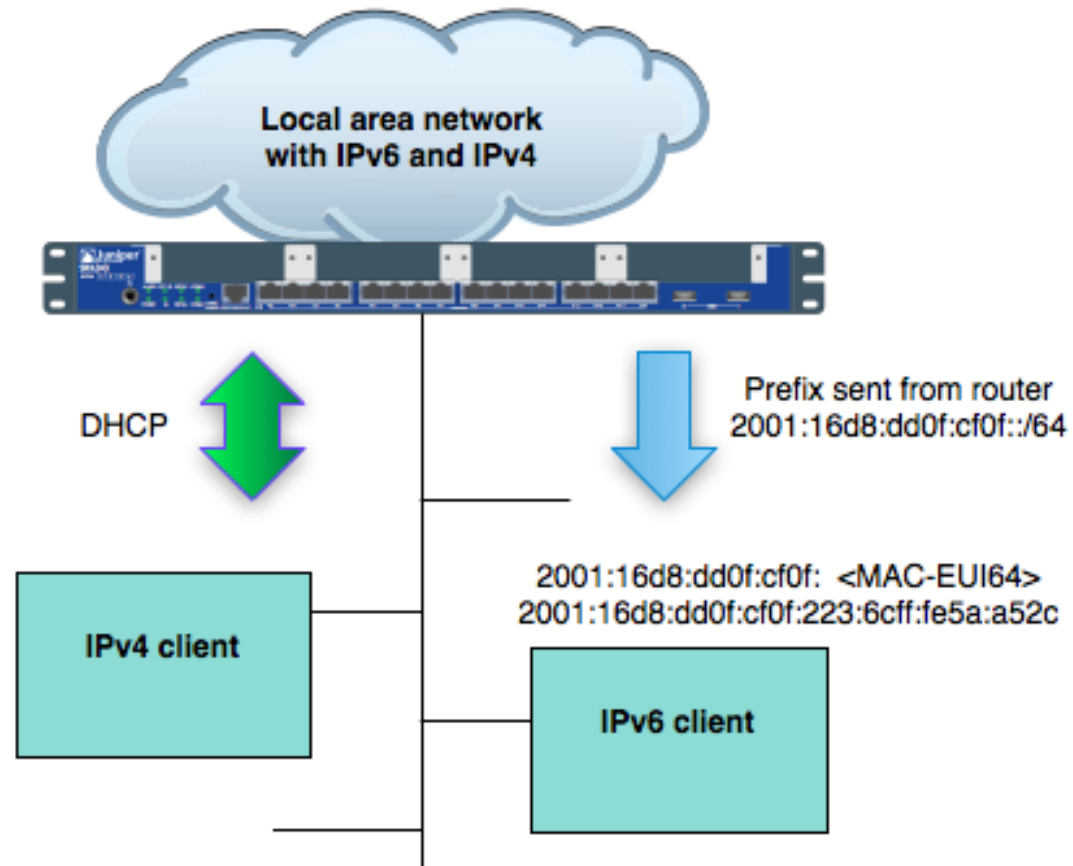
Modified EUI-64 format-based interface identifiers

```
ifconfig en1
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:23:6c:9a:f5:2c
        00-23-6c-ff-fe-9a-f5-2c 48-bit MAC stretched to become EUI-64
        02-23-6c-ff-fe-9a-f5-2c inverting the "u" bit (universal/local bit)
        fe80:: + 0223:6cff:fe9a:f52c add link-local prefix
    inet6 fe80::223:6cff:fe9a:f52c%en1 prefixlen 64 scopeid 0x6
```

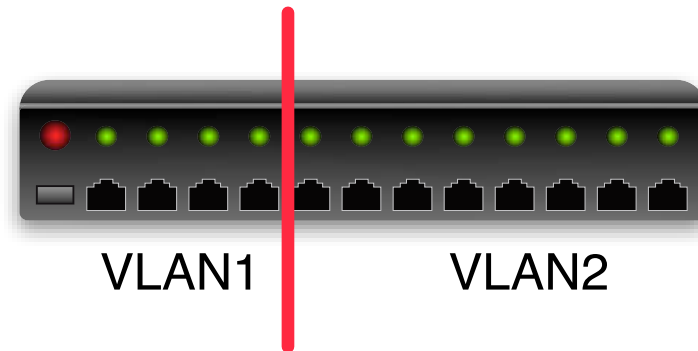
DHCPv6 is available, but **stateless autoconfiguration** is king

Routers announce subnet prefix via **router advertisements**

Individual nodes then combine this with their EUI64 identifier



Portbased VLAN



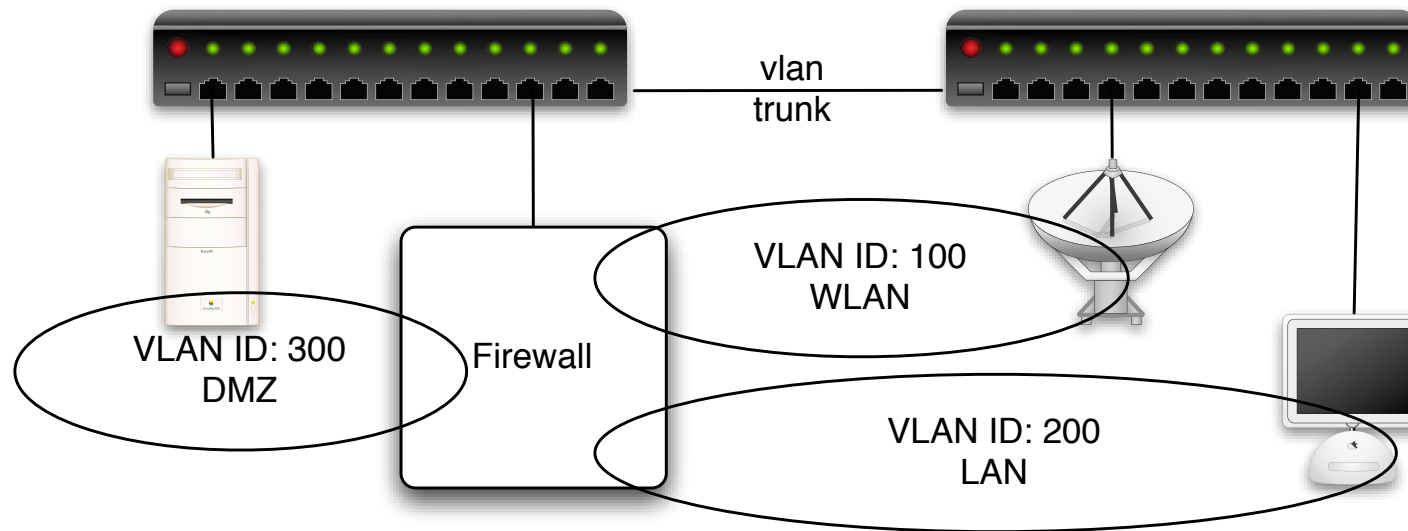
Nogle switche tillader at man opdeler portene

Denne opdeling kaldes VLAN og portbaseret er det mest simple

Port 1-4 er et LAN

De resterende er et andet LAN

Data skal omkring en firewall eller en router for at krydse fra VLAN1 til VLAN2



Nogle switche tillader konfiguration med 802.1q VLAN tagging på Ethernet niveau

Data skal omkring en firewall eller en router for at krydse fra VLAN1 til VLAN2

VLAN trunking giver mulighed for at dele VLANs ud på flere switches

Der findes administrationsværktøjer der letter dette arbejde: OpenNAC FreeNAC, Cisco VMPS

Stop - vi gennemgår og tester vores setup

Vi gennemgår hvordan vores setup ser ud

Vi laver traceroute før og efter:

Vi fjerner en ledning *link down*

Vi stopper en router og ser de annoncerede netværk forsvinder

Vi booter en router og ser de annoncerede netværk igen

Stop - vi ser i fællesskab på admin interfaces

Vi prøver lige at se på diverse interfaces sammen

Huskeliste til Henrik:

- Airport Extreme access-point
- Juniper SRX router/firewall
- OpenBSD router
- Linux Ubuntu config

```
ifconfig en0 10.0.42.1 netmask 255.255.255.0  
route add default gw 10.0.42.1
```

konfiguration af interfaces og netværk på Unix foregår med:

`ifconfig`, `route` **og** `netstat`

- ofte pakket ind i konfigurationsmenuer m.v.

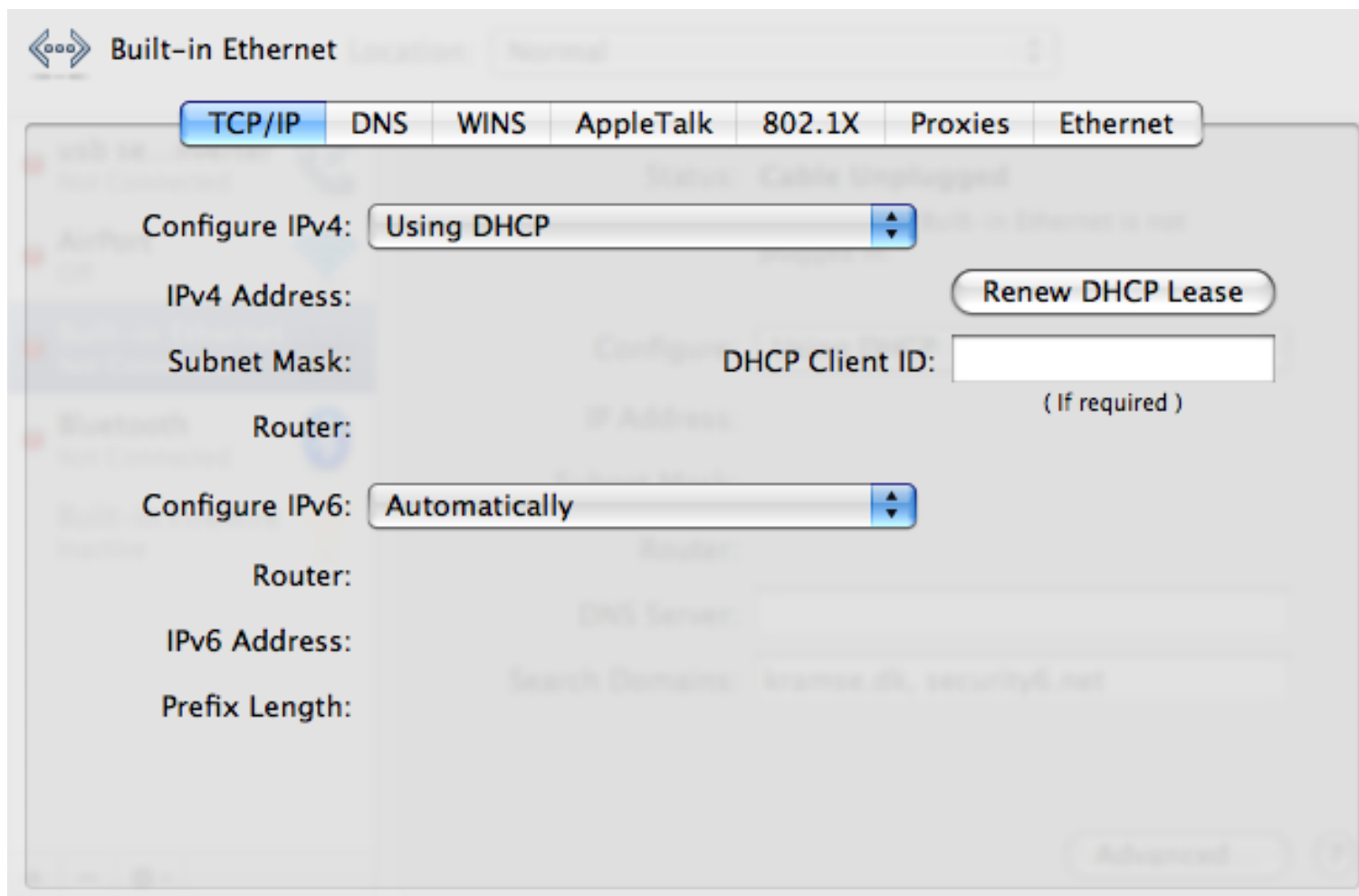
fejlsøgning foregår typisk med `ping` **og** `traceroute`

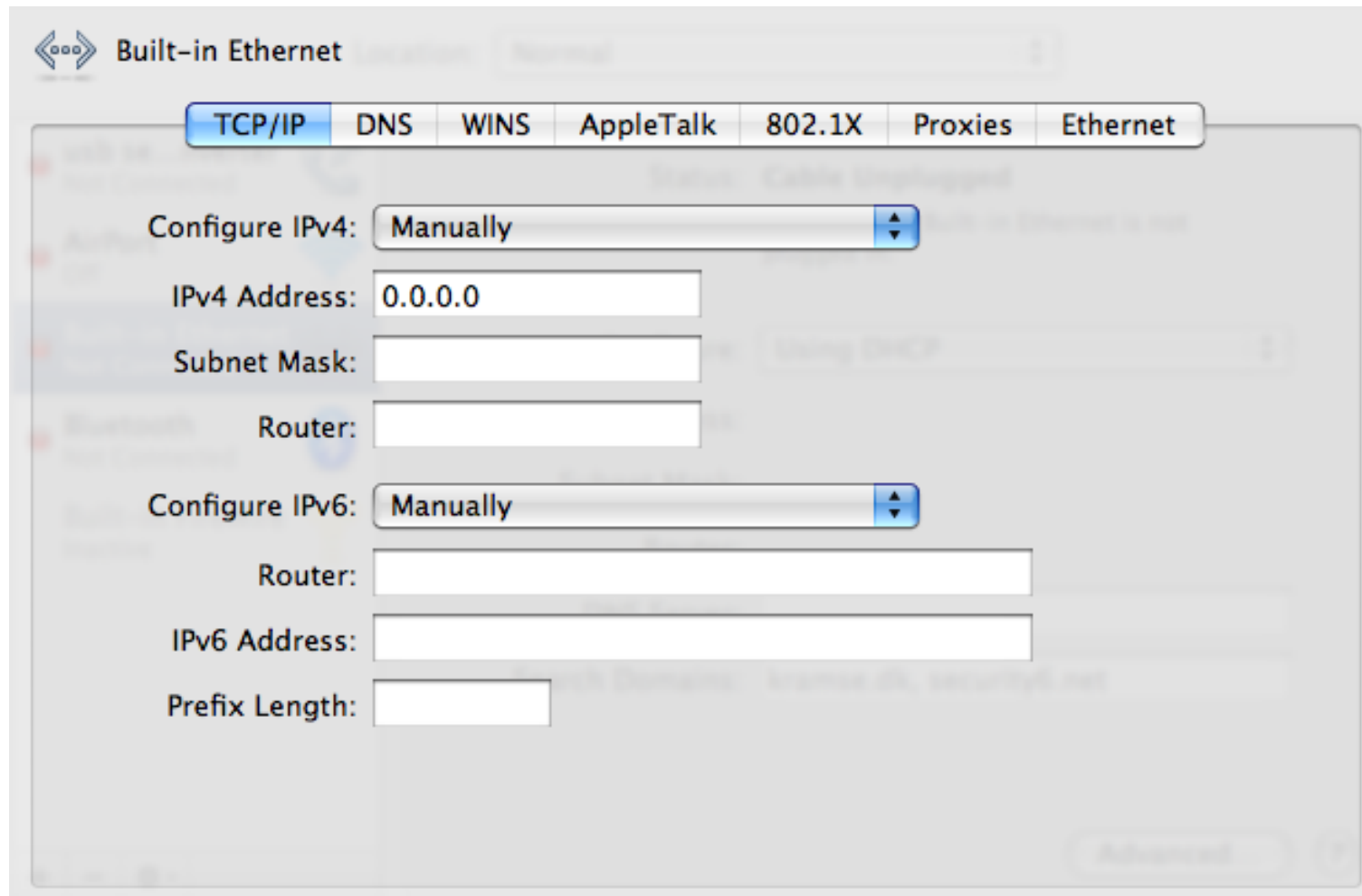
På Microsoft Windows benyttes ikke `ifconfig`
men kommandoerne `ipconfig` **og** `ipv6`

Netværkskonfiguration på OpenBSD:

To netkort med fast IPv6 og DHCP på ydersiden

```
# cat /etc/hostname.em0
inet 10.0.42.1 255.255.255.0
inet6 2a02:2190:f000:cf0f::1
# cat hostname.em3
dhcp
inet6 2a02:2190:1004:da::2 64
!route add -inet6 default 2a02:2190:1004:da::1
# cat /etc/resolv.conf
domain security6.net
lookup file bind
nameserver 212.242.40.3
nameserver 212.242.40.51
```





```
hlk@bigfoot:hlk$ ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:0a:95:db:c8:b0
    media: autoselect (none) status: inactive
    supported media: none autoselect 10baseT/UTP <half-duplex> 10baseT/UTP
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:0d:93:86:7c:3f
    media: autoselect (<unknown type>) status: inactive
    supported media: autoselect
```

ifconfig output er næsten ens på tværs af Unix

ARP Address Resolution Protocol

IP og ICMP Internet Control Message Protocol

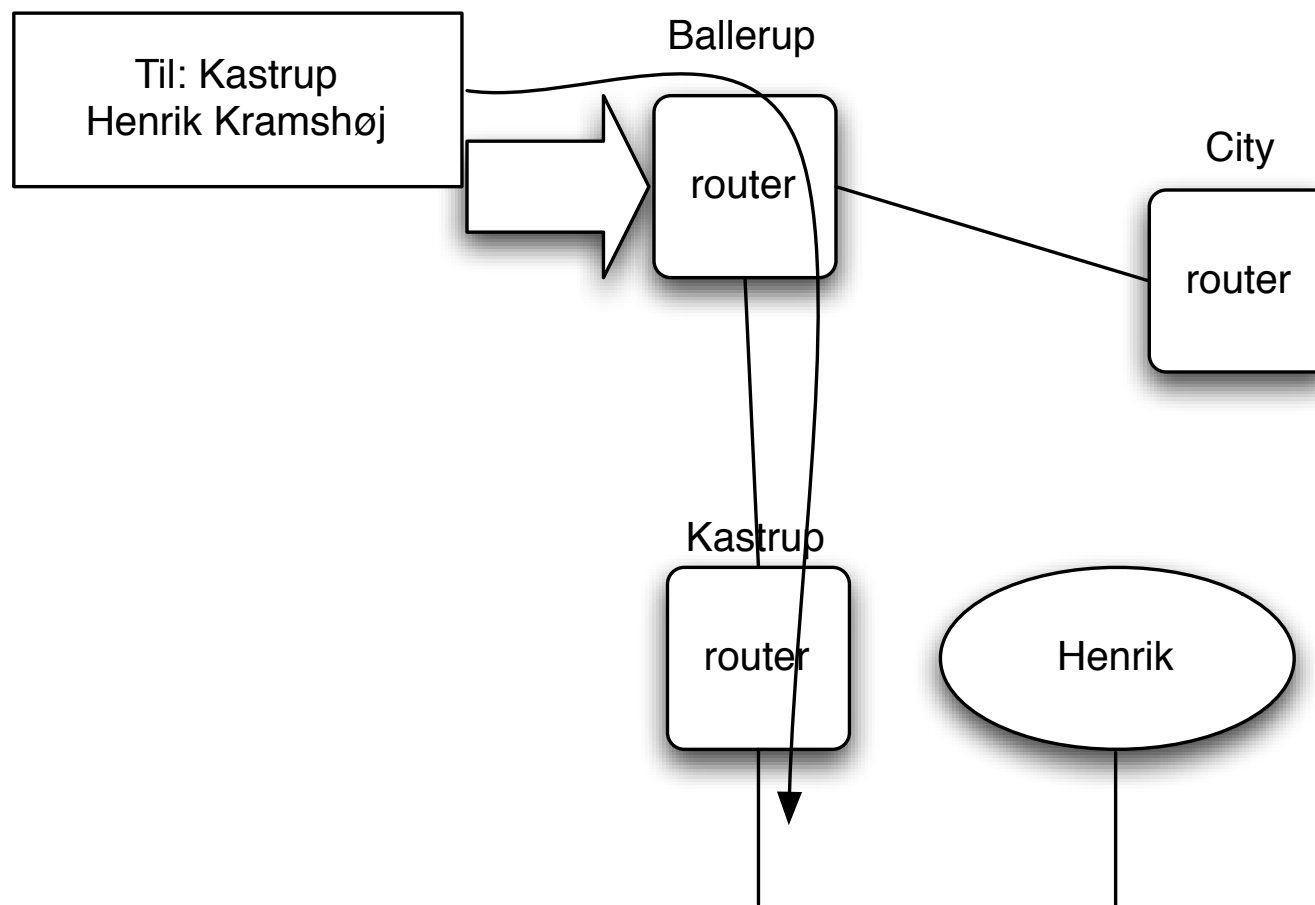
UDP User Datagram Protocol

TCP Transmission Control Protocol

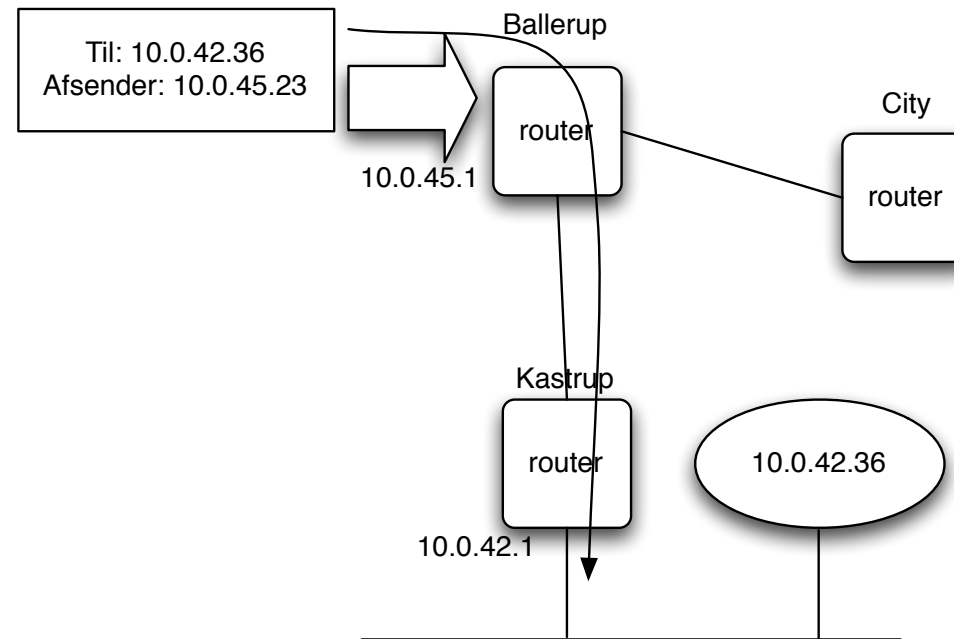
DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

Ovenstående er omtrent minimumskrav for at komme på internet



Hvordan kommer pakkerne frem til modtageren

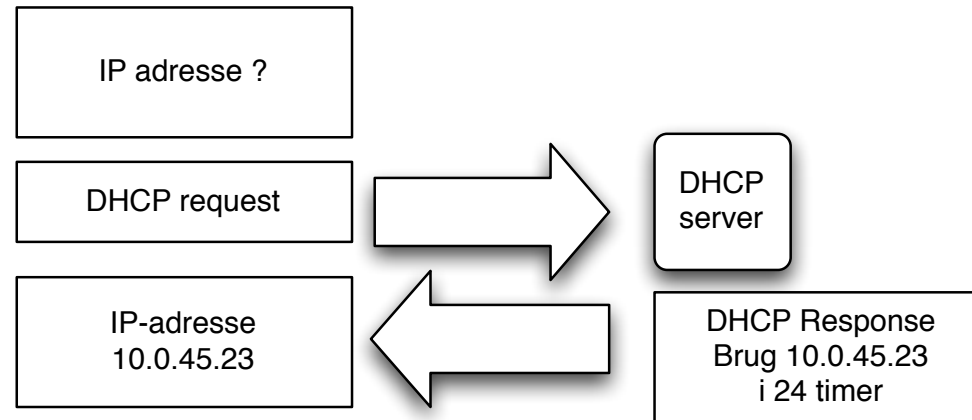


IP routing er nemt - **longest match vinder altid**

En host kender en default gateway i nærheden

En router har en eller flere upstream routere, få adresser den sender videre til

Core internet har default free zone, kender *alle netværk*



Hvordan får man information om default gateway

Man sender et DHCP request og modtager et svar fra en DHCP server

Dynamisk konfiguration af klienter fra en centralt konfigureret server

Bruges til IP adresser og meget mere

```
/etc/rtadvd.conf:
en0:
    :addrs#1:addr="2001:1448:81:b00f::":prefixlen#64:
en1:
    :addrs#1:addr="2001:1448:81:beef::":prefixlen#64:

root# /usr/sbin/rtadvd -Df en0 en1
root# sysctl -w net.inet6.ip6.forwarding=1
net.inet6.ip6.forwarding: 0 -> 1
```

Stateless autoconfiguration er en stor ting i IPv6

Kommandoen starter den i debug-mode og i forgrunden
- normalt vil man starte den fra et script

Typisk skal forwarding aktiveres, som vist med BSD sysctl kommando

routing table - tabel over netværkshort og tilhørende adresser

default gateway - den adresse hvortil man sender *non-local* pakker
kalderes også default route, gateway of last resort

routing styres enten manuelt - opdatering af route tabellen, eller konfiguration af adresser og subnet maske på netkort

eller automatisk ved brug af routing protocols - interne og eksterne route protokoller

de lidt ældre routing protokoller har ingen sikkerhedsmekanismer

IP benytter longest match i routing tabeller!

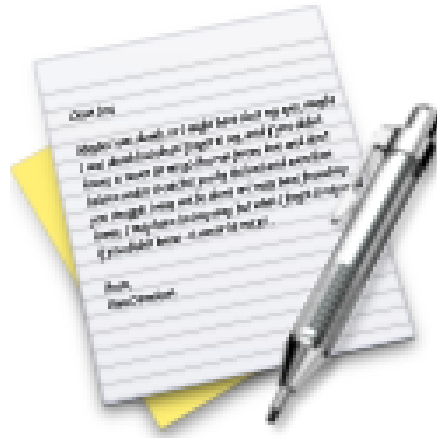
Den mest specifikke route gælder for forward af en pakke!

```
$ netstat -rn
Routing tables
```

Internet:

Destination	Gateway	Flags	Refs	Use	Netif
default	10.0.0.1	UGSc	23	7	en0
10/24	link#4	UCS	1	0	en0
10.0.0.1	0:0:24:c1:58:ac	UHLW	24	18	en0
10.0.0.33	127.0.0.1	UHS	0	1	lo0
10.0.0.63	127.0.0.1	UHS	0	0	lo0
127	127.0.0.1	UCS	0	0	lo0
127.0.0.1	127.0.0.1	UH	4	7581	lo0
169.254	link#4	UCS	0	0	en0

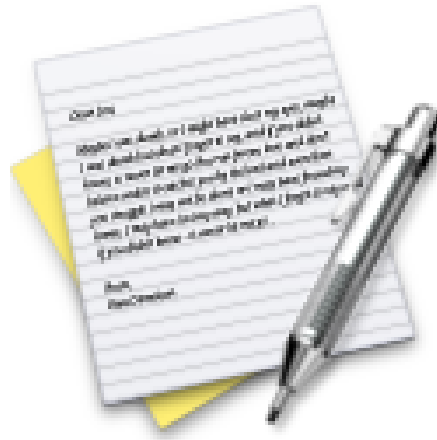
Start med kun at se på Destination, Gateway og Netinterface



Vi laver nu øvelsen

Netværksinformation: ifconfig/ipconfig

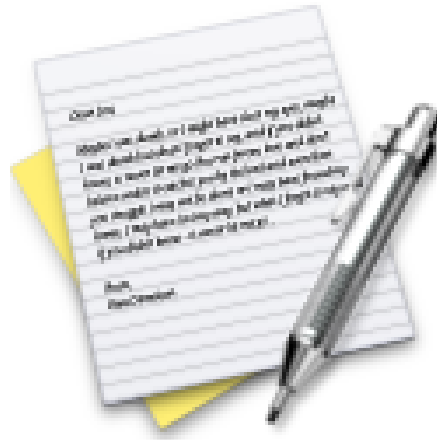
som er øvelse 4 fra øvelseshæftet.



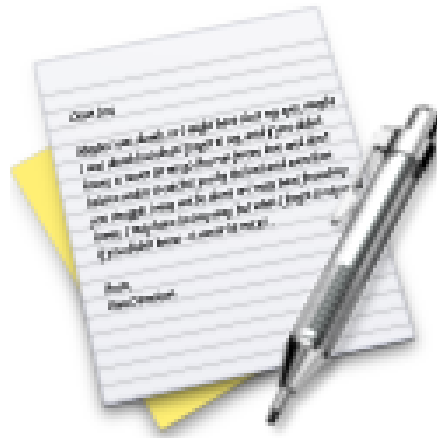
Vi laver nu øvelsen

Netværksinformation: netstat

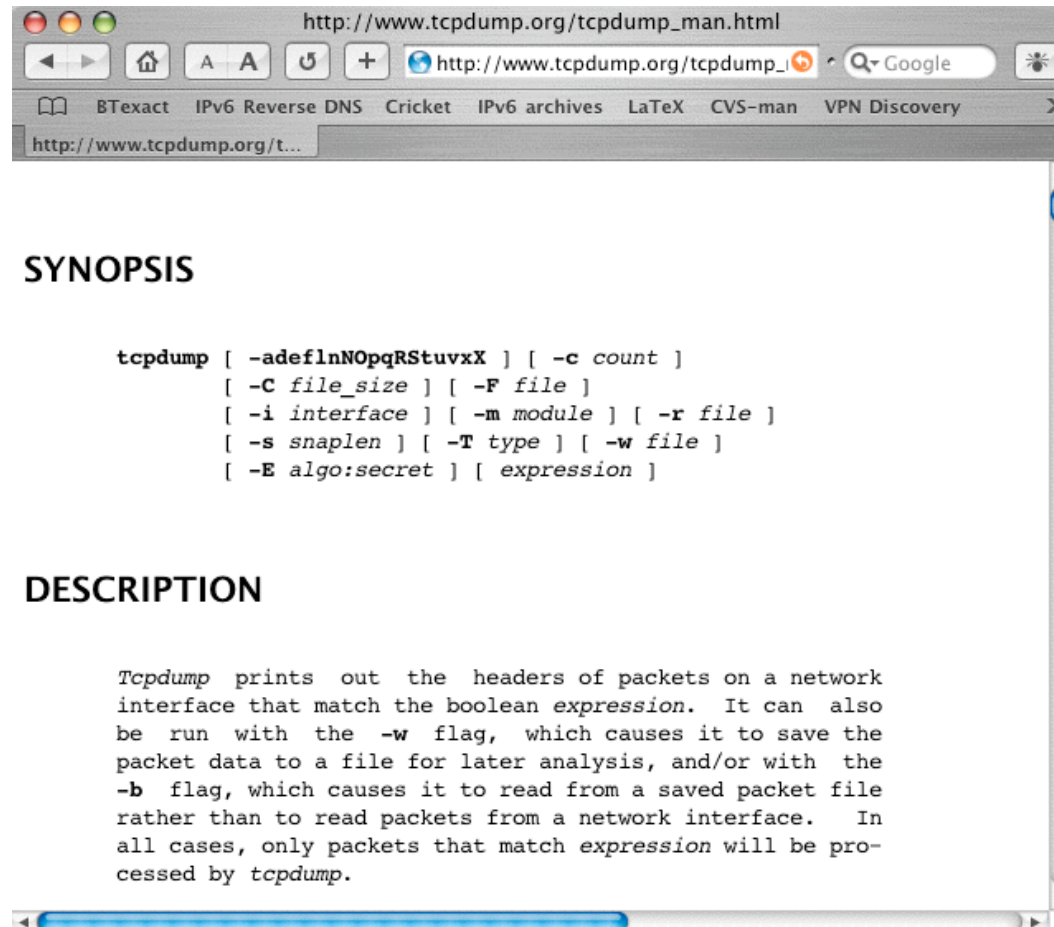
som er øvelse 5 fra øvelseshæftet.



Vi laver nu øvelsen
ping og traceroute
som er øvelse **6** fra øvelseshæftet.



Vi laver nu øvelsen
ping6 og traceroute6
som er øvelse **7** fra øvelseshæftet.



<http://www.tcpdump.org> - både til Windows og Unix

- tekstmode
- kan gemme netværkspakker i filer
- kan læse netværkspakker fra filer
- er de-facto standarden for at gemme netværksdata i filer

```
[root@otto hlk]# tcpdump -i en0
tcpdump: listening on en0
13:29:39.947037 fe80::210:a7ff:fe0b:8a5c > ff02::1: icmp6: router advertisement
13:29:40.442920 10.0.0.200.49165 > dns1.cybercity.dk.domain: 1189+[|domain]
13:29:40.487150 dns1.cybercity.dk.domain > 10.0.0.200.49165: 1189 NXDomain*+[|domain]
13:29:40.514494 10.0.0.200.49165 > dns1.cybercity.dk.domain: 24765+[|domain]
13:29:40.563788 dns1.cybercity.dk.domain > 10.0.0.200.49165: 24765 NXDomain*+[|domain]
13:29:40.602892 10.0.0.200.49165 > dns1.cybercity.dk.domain: 36485+[|domain]
13:29:40.648288 dns1.cybercity.dk.domain > 10.0.0.200.49165: 36485 NXDomain*+[|domain]
13:29:40.650596 10.0.0.200.49165 > dns1.cybercity.dk.domain: 4101+[|domain]
13:29:40.694868 dns1.cybercity.dk.domain > 10.0.0.200.49165: 4101 NXDomain*+[|domain]
13:29:40.805160 10.0.0.200 > mail: icmp: echo request
13:29:40.805670 mail > 10.0.0.200: icmp: echo reply
...
```

filtre til husbehov

- type - host, net og port
- src pakker med afsender IP eller afsender port
- dst pakker med modtager IP eller modtager port
- host - afsender eller modtager
- proto - protokol: ether, fddi, tr, ip, ip6, arp, rarp, decnet, tcp og udp

IP adresser kan angives som dotted-decimal eller navne

porte kan angives med numre eller navne

komplekse udtryk opbygges med logisk and, or, not

Host 10.1.2.3

Alle pakker hvor afsender eller modtager er 10.1.2.3


host 10.2.3.4 and not host 10.3.4.5

Alle pakker til/fra 10.2.3.4 undtagen dem til/fra 10.3.4.5

- meget praktisk hvis man er logget ind på 10.2.3.4 via netværk fra 10.3.4.5

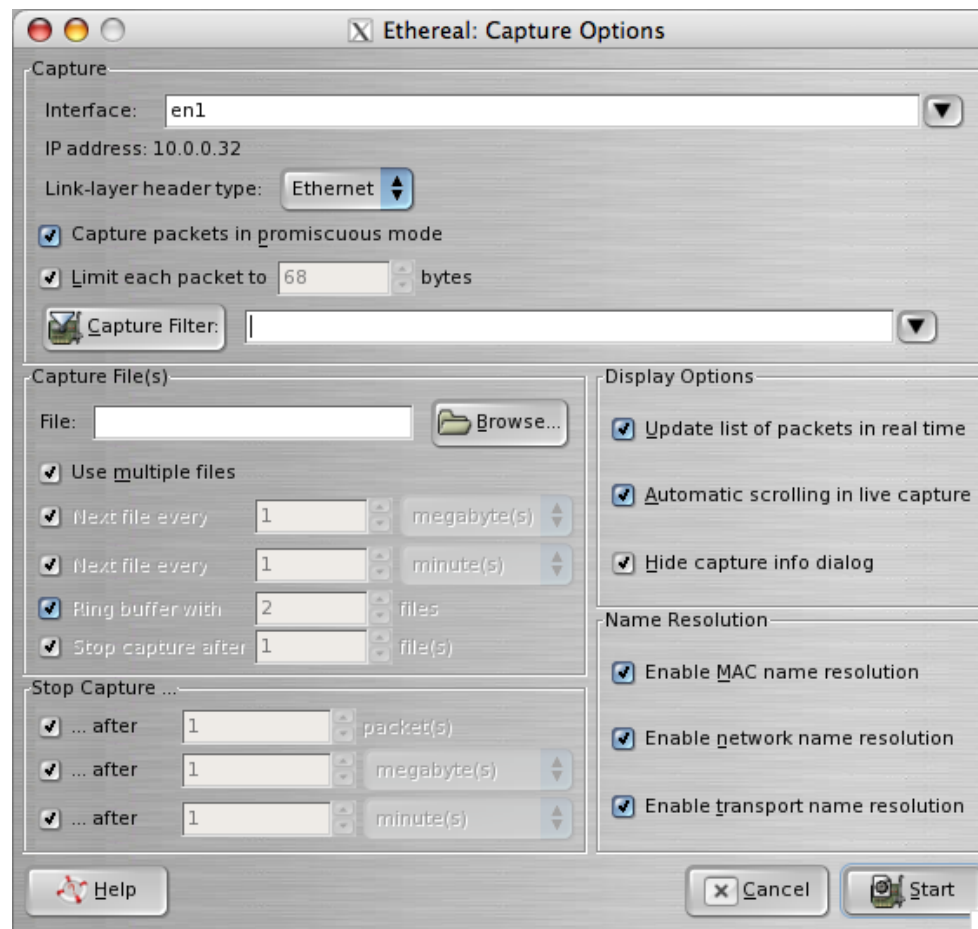
host foo and not port ftp and not port ftp-data

trafik til/fra maskine *foo* undtagen hvis det er FTP trafik

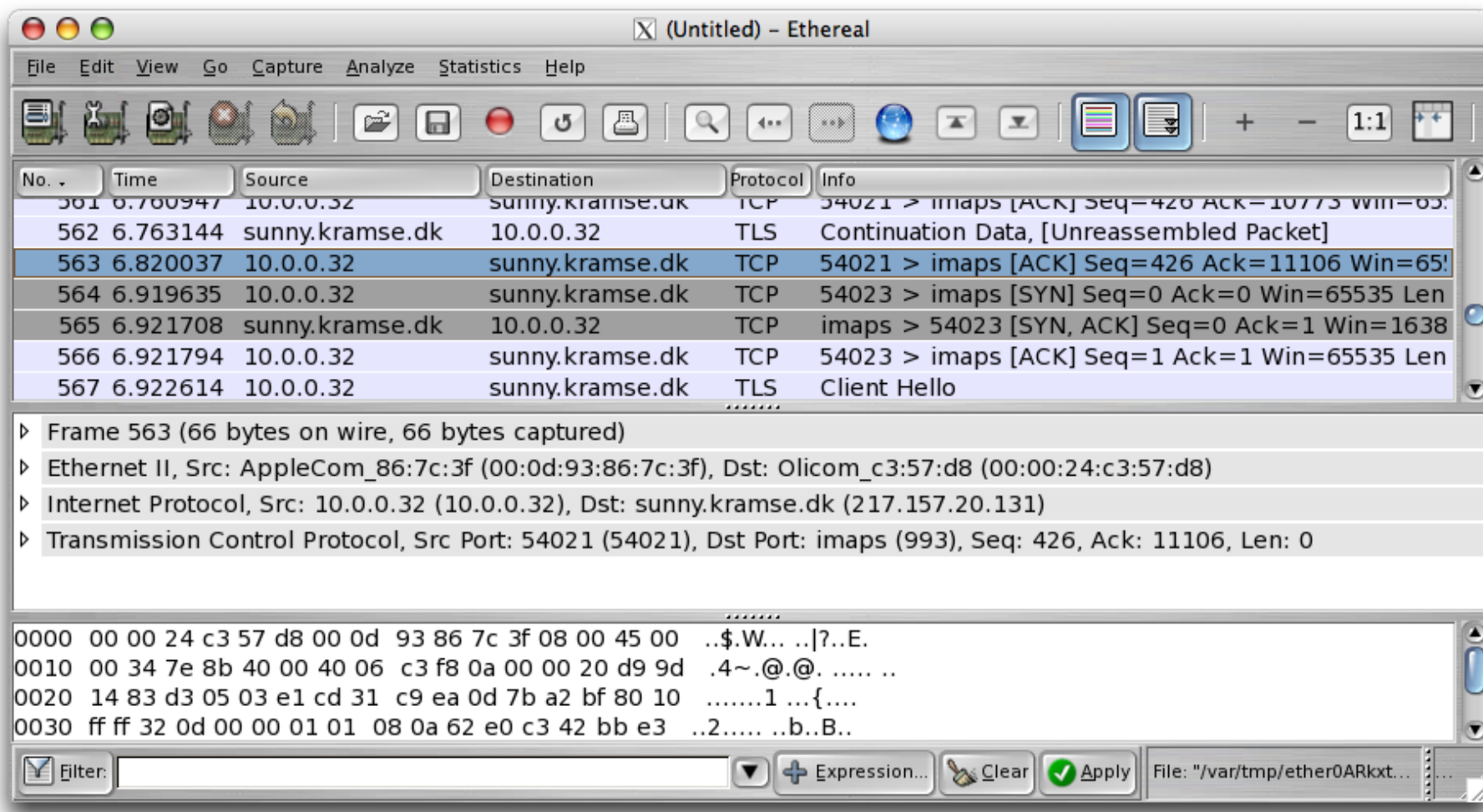


The screenshot shows the Wireshark website homepage. At the top is a blue banner with the 'WIRESHARK' logo and a shark illustration. Below the banner is a navigation bar with links: HOME, ABOUT, WHAT'S NEW, DOWNLOAD, and FAQ. The main content area is divided into several sections. On the left is a sidebar with links under 'Get It' (Download), 'Get Help' (FAQs, Documentation, Mailing Lists, Wiki, Bug tracker), 'Develop' (Developer Info), and 'Products' (AirPcap, Network Toolkit, OEM WinPcap). The central part features an article titled 'Sniffing Problems A Mile Away' about the name change from Ethereal to Wireshark, accompanied by a screenshot of the Wireshark interface. Below this is a 'News' section announcing 'Wireshark 0.99.3 Released' on August 23, 2006, detailing security fixes. To the right of the article is a 'Download Now' box for version 0.99.3, and below that is a Q&A section with the question 'How do I capture 802.11 traffic on Windows?' and the answer 'AirPcap'.

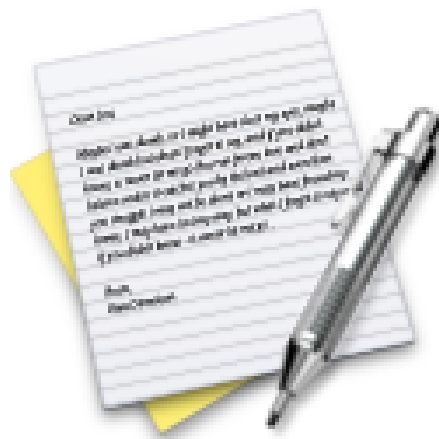
<http://www.wireshark.org>
både til Windows og Unix, tidligere kendt som Ethereal



Man starter med Capture - Options



Læg mærke til filtermulighederne



Vi laver nu øvelsen

Wireshark netværksniffer

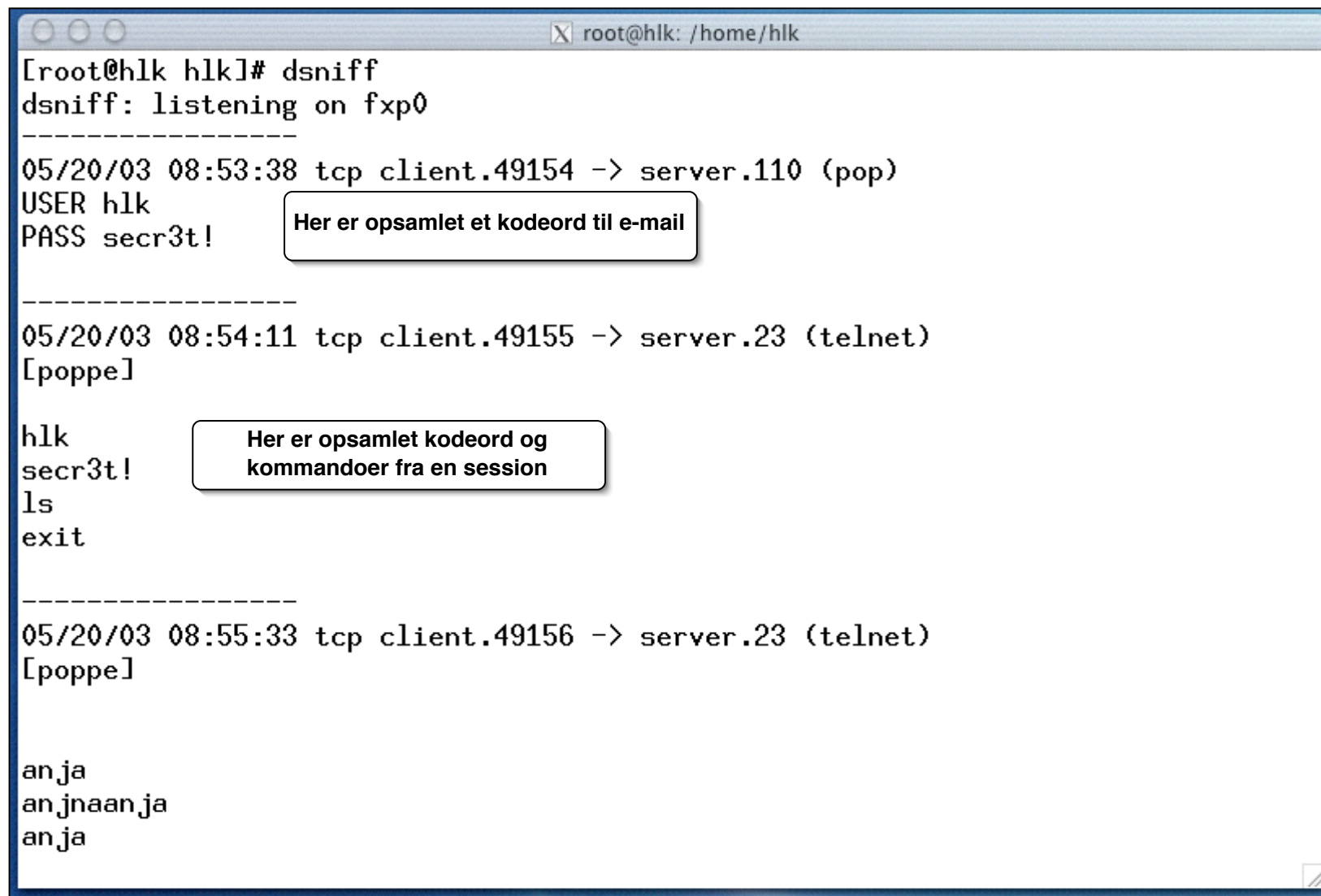
som er øvelse 8 fra øvelseshæftet.

en sniffer til mange usikre protokoller

inkluderer **arpspoof**

Lavet af Dug Song, dugsong@monkey.org

dsniff is a password sniffer which handles FTP, Telnet, SMTP, HTTP, POP, poppass, NNTP, IMAP, SNMP, LDAP, Rlogin, RIP, OSPF, PPTP, MS-CHAP, NFS, VRRP, YP/NIS, SOCKS, X11, CVS, IRC, AIM, ICQ, Napster, PostgreSQL, Meeting Maker, Citrix ICA, Symantec pcAnywhere, NAI Sniffer, Microsoft SMB, Oracle SQL*Net, Sybase and Microsoft SQL protocols.



```
root@h1k: /home/h1k
[root@h1k h1k]# dsniff
dsniff: listening on fxp0
-----
05/20/03 08:53:38 tcp client.49154 -> server.110 (pop)
USER h1k
PASS secr3t!
-----
05/20/03 08:54:11 tcp client.49155 -> server.23 (telnet)
[poppe]
h1k
secr3t!
ls
exit
-----
05/20/03 08:55:33 tcp client.49156 -> server.23 (telnet)
[poppe]
an ja
an jna an ja
an ja
```

Her er opsamlet et kodeord til e-mail

Her er opsamlet kodeord og kommandoer fra en session



Chaosreader Report

Created at: Sun Nov 16 21:04:18 2003, Type: snoop

[Image Report](#) - Click here for a report on captured images.

[GET/POST Report](#) (Empty) - Click here for a report on HTTP GETs and POSTs.

[HTTP Proxy Log](#) - Click here for a generated proxy style HTTP log.

TCP/UDP/... Sessions

1.	Sun Nov 16 20:38:22 2003	30 s	192.168.1.3:1368 <-> 192.77.84.99:80	web	383 bytes	• as_html
2.	Sun Nov 16 20:38:22 2003	29 s	192.168.1.3:1366 <-> 192.77.84.99:80	web	381 bytes	• as_html

Med adgang til et netværksdump kan man læse det med chaosreader

Output er HTML med oversigter over sessioner, billeder fra datastrømmen osv.

<http://chaosreader.sourceforge.net/>

Kryptering af e-mail

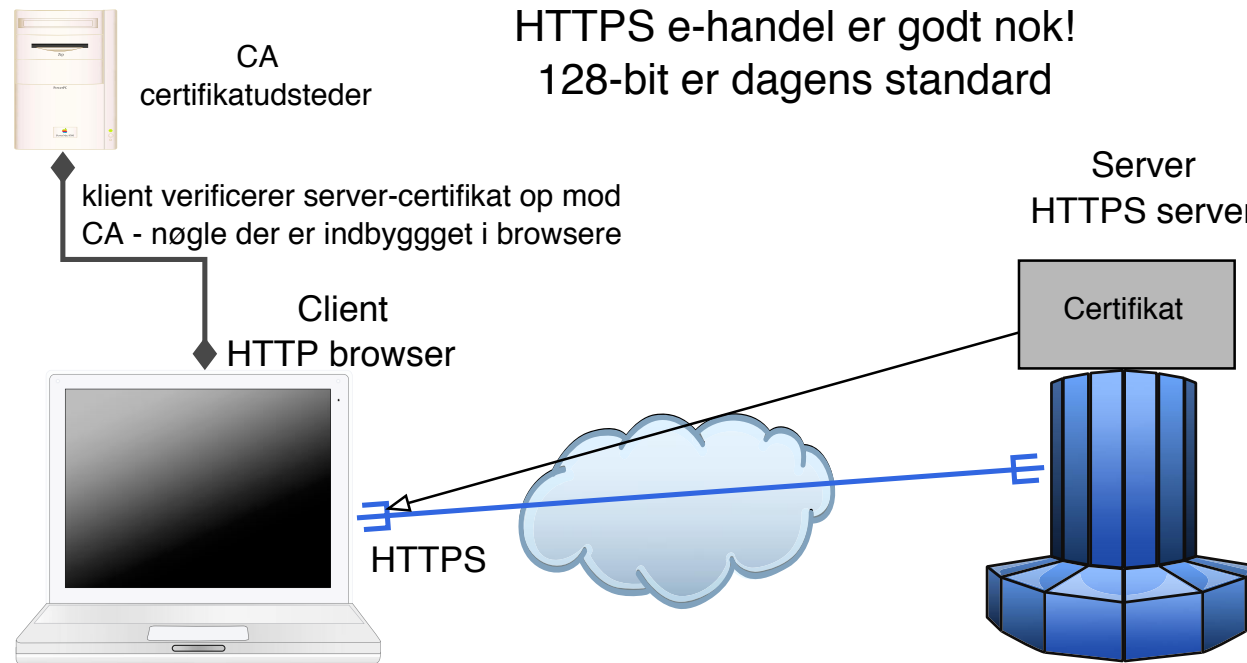
- Pretty Good Privacy - Phil Zimmermann
- GNU Privacy Guard - Open Source implementation af OpenPGP
- OpenPGP = mail sikkerhed, OpenPGP RFC-2440, PGP/MIME RFC 3156)

Kryptering af sessioner SSL/TLS

- Secure Sockets Layer SSL / Transport Layer Services TLS
- krypterer data der sendes mellem webservere og klienter
- SSL kan bruges generelt til mange typer sessioner, eksempelvis POP3S, IMAPS, SSH m.fl.

Kryptering af netværkstrafik - Virtual Private Networks VPN

- IPsec IP Security Framework, se også L2TP
- PPTP Point to Point Tunneling Protocol - dårlig og usikker, brug den ikke mere!
- OpenVPN m.fl.



Oprindeligt udviklet af Netscape Communications Inc.

Secure Sockets Layer SSL er idag blevet adopteret af IETF og kaldes derfor også for Transport Layer Security TLS TLS er baseret på SSL Version 3.0

RFC-2246 The TLS Protocol Version 1.0 fra Januar 1999

Check with your system administrator before changing any of the advanced options below:

IMAP Path Prefix:

Port: ☒ Use SSL

Authentication:

Mange protokoller findes i udgaver hvor der benyttes SSL

HTTPS vs HTTP

IMAPS, POP3S, osv.

Bemærk: nogle protokoller benytter to porte IMAP 143/tcp vs IMAPS 993/tcp

Andre benytter den samme port men en kommando som starter:

SMTP STARTTLS RFC-3207



Hvad er Secure Shell SSH?

Oprindeligt udviklet af Tatu Ylönen i Finland,
se <http://www.ssh.com>

SSH afløser en række protokoller som er usikre:

- Telnet til terminal adgang
- r* programmerne, rsh, rcp, rlogin, ...
- FTP med brugerid/password

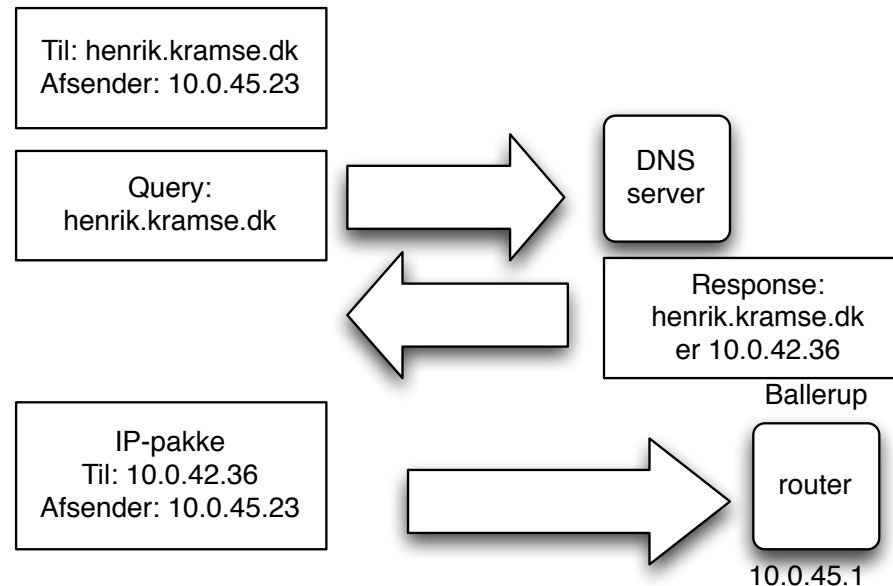
kommandoerne er:

- ssh - Secure Shell
- scp - Secure Copy
- sftp - secure FTP

Husk: SSH er både navnet på protokollerne - version 1 og 2 samt programmet `ssh` til at logge ind på andre systemer

SSH tillader også port-forward, tunnel til usikre protokoller, eksempelvis X protokollen til Unix grafiske vinduer

NB: Man bør idag bruge SSH protokol version 2!



Gennem DHCP får man typisk også information om DNS servere

En DNS server kan slå navne, domæner og adresser op

Foregår via query og response med datatyper kaldet resource records

DNS er en distribueret database, så opslag kan resultere i flere opslag

består af resource records med en type:

- adresser A-records
- IPv6 adresser AAAA-records
- autoritative navneservere NS-records
- post, mail-exchanger MX-records
- flere andre: md , mf , cname , soa , mb , mg , mr , null , wks , ptr , hinfo , minfo , mx

ns1	IN	A	217.157.20.130
	IN	AAAA	2001:618:433::1
www	IN	A	217.157.20.131
	IN	AAAA	2001:618:433::14
	IN	MX	10 mail.security6.net.
	IN	MX	20 mail2.security6.net.

Husk også DANE DNS-Based Authentication of Named Entities RFC 6698

`/etc/resolv.conf`

NB: denne fil kan hedde noget andet på Unix varianter!

eksempelvis `/etc/netsvc.conf`

typisk indhold er domænenavn og IP-adresser for navneservere

```
domain solido.net
nameserver 212.242.40.3
nameserver 212.242.40.51
```

Berkeley Internet Name Daemon server

BIND fra Internet Systems Consortium

konfigureres gennem `named.conf`

det anbefales at bruge BIND version 9 eller senere

- *DNS and BIND*, Paul Albitz & Cricket Liu, O'Reilly, 4th edition Maj 2001
- *DNS and BIND cookbook*, Cricket Liu, O'Reilly, 4th edition Oktober 2002

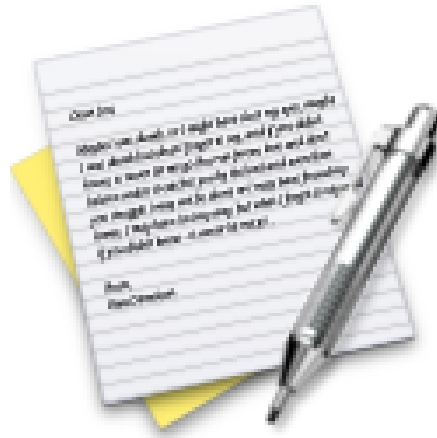
Kilde: `http://www.isc.org`

```
acl internals { 127.0.0.1; ::1; 10.0.0.0/24; };
options {
    // the random device depends on the OS !
    random-device "/dev/random"; directory "/namedb";
    listen-on-v6 any;
    port 53; version "Dont know"; allow-query { any; };
};
view "internal" {
    match-clients { internals; }; recursion yes;
    zone "." {
        type hint;    file "root.cache"; };
    // localhost forward lookup
    zone "localhost." {
        type master; file "internal/db.localhost";    };
    // localhost reverse lookup from IPv4 address
    zone "0.0.127.in-addr.arpa" {
        type master; file "internal/db.127.0.0"; notify no;    };
    ...
}
```

Unbound DNS server <http://www.nlnetlabs.nl/projects/unbound/>

NSD <http://www.nlnetlabs.nl/projects/nsd/>

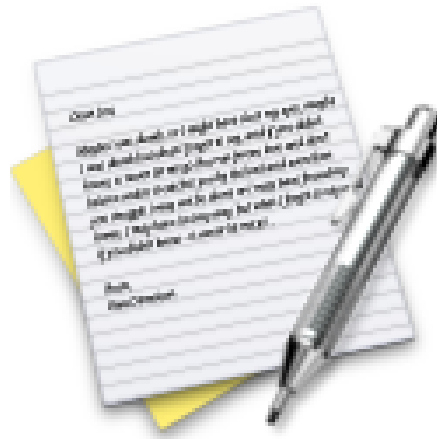
OpenDNSSEC <http://www.opendnssec.org/>



Vi laver nu øvelsen

DNS og navneopslag

som er øvelse 9 fra øvelseshæftet.



Vi laver nu øvelsen

DNS og navneopslag - IPv6

som er øvelse **10** fra øvelseshæftet.

Små DNS tools bind-version - Shell script

```
#!/bin/sh
# Try to get version info from BIND server
PROGRAM=`basename $0`
. `dirname $0`/functions.sh
if [ $# -ne 1 ]; then
    echo "get name server version, need a target! "
    echo "Usage: $0 target"
    echo "example $0 10.1.2.3"
    exit 0
fi
TARGET=$1
# using dig
start_time
dig @$1 version.bind chaos txt
echo Authors BIND er i versionerne 9.1 og 9.2 - måske ...
dig @$1 authors.bind chaos txt
stop_time

http://www.kramse.dk/files/tools/dns/bind-version
```

Små DNS tools dns-timecheck - Perl script

```
#!/usr/bin/perl
# modified from original by Henrik Kramshøj, hlk@kramse.dk
# 2004-08-19
#
# Original from: http://www.rfc.se/fpdns/timecheck.html
use Net::DNS;

my $resolver = Net::DNS::Resolver->new;
$resolver->nameservers($ARGV[0]);

my $query = Net::DNS::Packet->new;
$query->sign_tsig("n", "test");

my $response = $resolver->send($query);
foreach my $rr ($response->additional)
    print "localtime vs nameserver $ARGV[0] time difference: ";
    print $rr->time_signed - time() if $rr->type eq "TSIG";
```

<http://www.kramse.dk/files/tools/dns/dns-timecheck>

RFC-2142 Mailbox Names for Common Services, Roles and Functions

Du BØR konfigurere dit domæne til at modtage post for følgende adresser:

- postmaster@domæne.dk
- abuse@domæne.dk
- webmaster@domæne.dk, evt. www@domæne.dk

Du gør det nemmere at rapportere problemer med dit netværk og services

MAILBOX -----	AREA -----	USAGE -----
ABUSE	Customer Relations	Inappropriate public behaviour
NOC	Network Operations	Network infrastructure
SECURITY	Network Security	Security bulletins or queries
...		
MAILBOX -----	SERVICE -----	SPECIFICATIONS -----
POSTMASTER	SMTP	[RFC821], [RFC822]
HOSTMASTER	DNS	[RFC1033-RFC1035]
USENET	NNTP	[RFC977]
NEWS	NNTP	Synonym for USENET
WEBMASTER	HTTP	[RFC 2068]
WWW	HTTP	Synonym for WEBMASTER
UUCP	UUCP	[RFC976]
FTP	FTP	[RFC959]

Kilde: RFC-2142 Mailbox Names for Common Services, Roles and Functions. D. Crocker. May 1997

IP adresserne administreres i dagligdagen af et antal Internet registries, hvor de største er:

- RIPE (Réseaux IP Européens) <http://ripe.net>
- ARIN American Registry for Internet Numbers <http://www.arin.net>
- Asia Pacific Network Information Center <http://www.apnic.net>
- LACNIC (Regional Latin-American and Caribbean IP Address Registry) - Latin America and some Caribbean Islands

disse fire kaldes for Regional Internet Registries (RIRs) i modsætning til Local Internet Registries (LIRs) og National Internet Registry (NIR)

NTP opsætning

foregår typisk i `/etc/ntp.conf` eller `/etc/ntpd.conf`

det vigtigste er navnet på den server man vil bruge som tidskilde

Brug enten en NTP server hos din udbyder eller en fra `http://www.pool.ntp.org/`

Eksempelvis:

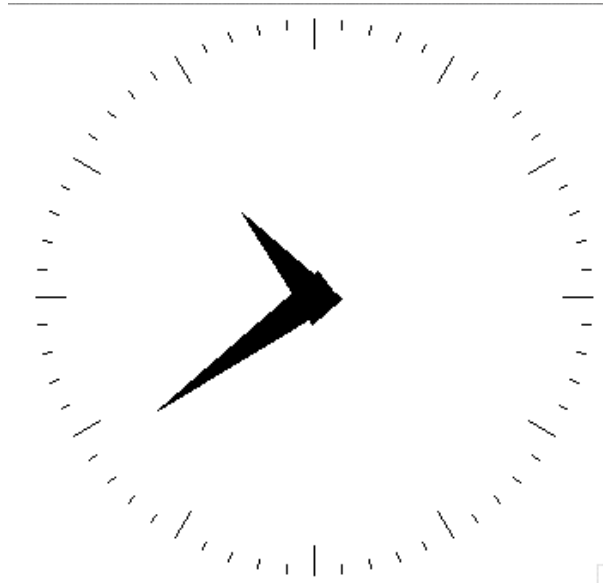
```
server ntp.cybercity.dk
```

```
server 0.dk.pool.ntp.org
```

```
server 0.europe.pool.ntp.org
```

```
server 3.europe.pool.ntp.org
```

What time is it?



Hvad er klokken?

Hvad betydning har det for sikkerheden?

Brug NTP Network Time Protocol på produktionssystemer

What time is it? - spørg ICMP

ICMP timestamp option - request/reply

hvad er klokken på en server

Slayer icmpush - er installeret på server

viser tidstempel

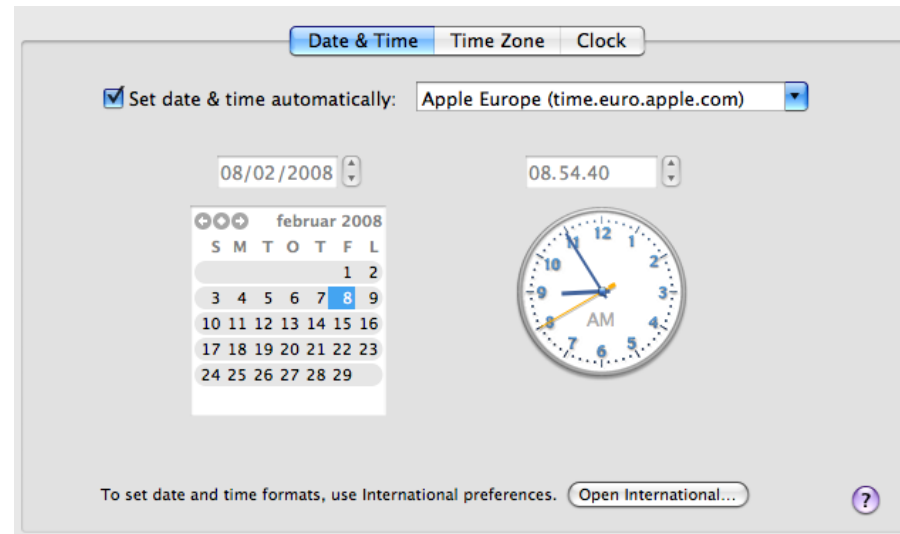
```
# icmpush -v -tstamp 10.0.0.12
```

```
ICMP Timestamp Request packet sent to 10.0.0.12 (10.0.0.12)
```

```
Receiving ICMP replies ...
```

```
fischer          -> 21:27:17
```

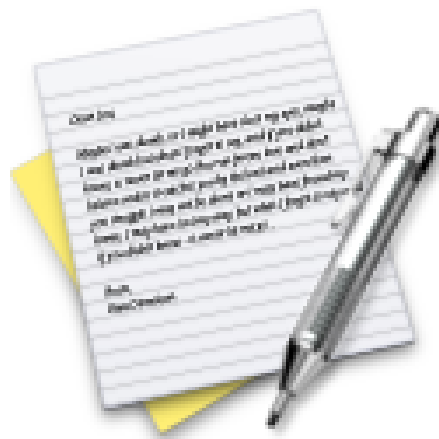
```
icmpush: Program finished OK
```



Vi har en masse udstyr, de meste kan NTP, men hvordan

Vi gennemgår, eller I undersøger selv:

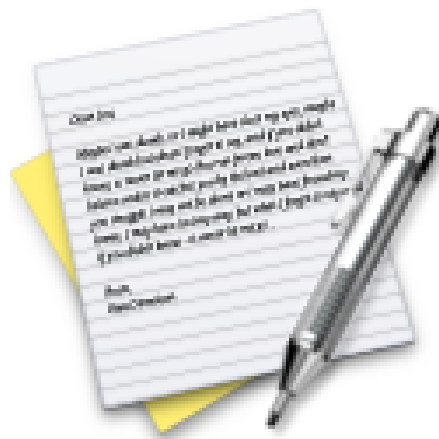
- Airport
- Switche (managed)
- Mac OS X
- OpenBSD - `check man rdate` og `man ntpd`



Vi laver nu øvelsen

Opslag i whois databaser

som er øvelse **11** fra øvelseshæftet.



Vi laver nu øvelsen

Test din forbindelse test-ipv6.com

som er øvelse **12** fra øvelseshæftet.



Vi laver nu øvelsen

Test din forbindelse for DNS problemer

som er øvelse **13** fra øvelseshæftet.



IP har eksisteret mange år

Vi har udskiftet langsomme forbindelser med hurtige forbindelser

Vi har udskiftet langsomme MHz maskiner med Quad-core GHz maskiner

IP var tidligere meget konservativt, for ikke at overbelaste modtageren

Billedet er en HP arbejdsstation med 19" skærm og en 60MHz HP PA-RISC processor

Der er visse indstillinger som tidligere var standard, de bør idag slås fra
En del er allerede tunet i nyere versioner af IP-stakkene, men check lige

Ideer til ting som skal slås fra:

- broadcast ICMP, undgå smurfing
- Source routing, kan måske omgå firewalls og filtre

Ideer til ting som skal slås til/ændres:

- Bufferstørrelser - hvorfor have en buffer på 65535 bytes på en maskine med 32GB ram?
- Nye funktioner som RFC-1323 TCP Extensions for High Performance

Det anbefales at finde leverandørens vejledning til hvad der kan tunes

```
# tuning
net.inet.tcp.recvspace=65535
net.inet.tcp.sendspace=65535
net.inet.udp.recvspace=65535
net.inet.udp.sendspace=32768
# postgresql tuning
kern.seminfo.semmni=256
kern.seminfo.semmns=2048
kern.shminfo.shmmax=50331648
```

På mange Unix varianter findes et specielt tuningsprogram, sysctl

Findes blandt andet på alle BSD'erne: FreeBSD, OpenBSD, NetBSD og Darwin/OSX

Ændringerne skrives ind i filen `/etc/sysctl.conf`

På Linux erstatter det til dels konfiguration med echo

```
echo 1 > /proc/net/ip/forwarding
```

På AIX benyttes kommandoen `network options no`

Hvad er flaskehalsen for programmet?

I/O bundet - en enkelt disk eller flere

CPU bundet - regnekraften

Netværket - 10Mbit half-duplex adapter

Memory - begynder systemet at *swappe* eller *thrash*

brug top og andre statistikprogrammer til at se disse data

Når der skal tunes er det altid nødvendigt med en baseline

Man kan ikke begynde at tune ud fra subjektive målinger

Det kører langsomt, Svartiden er for høj

Målinger der giver præcise tal er nødvendige, før og efter målinger!

Der findes et antal værktøjer til test, blandt andet Nping og Iperf

```
hlk@fluffy:hlk$ nping www.solidonetworks.com
Starting Nping 0.5.61TEST5 ( http://nmap.org/nping ) at 2012-03-15 07:06 CET
SENT (0.2006s) Starting TCP Handshake > www.solidonetworks.com:80 (91.102.95.20:80)
RECV (0.2198s) Handshake with www.solidonetworks.com:80 (91.102.95.20:80) completed
SENT (1.2030s) Starting TCP Handshake > www.solidonetworks.com:80 (91.102.95.20:80)
RECV (1.2213s) Handshake with www.solidonetworks.com:80 (91.102.95.20:80) completed
SENT (2.2056s) Starting TCP Handshake > www.solidonetworks.com:80 (91.102.95.20:80)
RECV (2.2260s) Handshake with www.solidonetworks.com:80 (91.102.95.20:80) completed
SENT (3.2076s) Starting TCP Handshake > www.solidonetworks.com:80 (91.102.95.20:80)
RECV (3.2286s) Handshake with www.solidonetworks.com:80 (91.102.95.20:80) completed
SENT (4.2098s) Starting TCP Handshake > www.solidonetworks.com:80 (91.102.95.20:80)
RECV (4.2301s) Handshake with www.solidonetworks.com:80 (91.102.95.20:80) completed

Max rtt: 21.040ms | Min rtt: 18.324ms | Avg rtt: 19.819ms
TCP connection attempts: 5 | Successful connections: 5 | Failed: 0 (0.00%)
Tx time: 4.01046s | Tx bytes/s: 99.74 | Tx pkts/s: 1.25
Rx time: 4.03064s | Rx bytes/s: 49.62 | Rx pkts/s: 1.24
Nping done: 1 IP address pinged in 4.23 seconds
```

God test af netværkslatency, samt TCP sockets på OS

```
hlk@fluffy:hlk$ iperf -s
```

```
-----  
Server listening on TCP port 5001  
TCP window size: 64.0 KByte (default)  
-----
```

```
[ 4] local 10.0.42.23 port 5001 connected with 10.0.42.67 port 51148  
[ 4]  0.0-10.2 sec  6.95 MBytes  5.71 Mbits/sec  
[ 4] local 10.0.42.23 port 5001 connected with 10.0.42.67 port 51149  
[ 4]  0.0-10.2 sec  7.02 MBytes  5.76 Mbits/sec
```

Ovenstående er set fra server, client kaldes med `iperf -c 10.0.42.23`

NB: brug i praksis altid mindst 60 sekunder, `iperf -i 5 -t 60 -c 10.0.42.23`

Stop - vi prøver i fællesskab Iperf

Vi prøver lige Iperf sammen

hvis alle prøver samtidig giver det stor variation i resultaterne

Telnet blev tidligere brugt til login og er en klartekst forbindelse over TCP

Telnet kan bruges til at teste forbindelsen til mange ældre serverprotokoller som benytter ASCII kommandoer

- `telnet mail.kramse.dk 25` laver en forbindelse til port 25/tcp
- `telnet www.kramse.dk 80` laver en forbindelse til port 80/tcp

Til krypterede forbindelser anbefales det at teste med openssl

- `openssl s_client -host www.kramse.dk -port 443`
laver en forbindelse til port 443/tcp med SSL
- `openssl s_client -host mail.kramse.dk -port 993`
laver en forbindelse til port 993/tcp med SSL

Med OpenSSL i client-mode kan services tilgås med samme tekstkommandoer som med telnet

UDP er lidt drilsk, for de fleste services er ikke *ASCII protokoller*

Der findes dog en række testprogrammer, a la ping

- nsping - name server ping
- dhcping - dhcp server ping
- ...

Derudover kan man bruge de sædvanlige programmer som `host` til navneopslag osv.

Til tider er det ikke båndbredden som sådan man vil måle

Specielt for routere er det vigtigt at de kan behandle mange pakker per sekund, pps

Til dette kan man lege med det indbyggede Ping program i flooding mode

Når programmet kaldes (som systemadministrator) med `ping -f server` vil den sende ping pakker så hurtigt som netkortet tillader

Programmer der kan teste pakker per sekund kaldes generelt for blaster tools


```
hlk@bigfoot:hlk$ ab -n 100 http://www.kramse.dk/  
This is ApacheBench, Version 2.0.41-dev <$Revision: 1.121.2.12 $> apache-2.0  
Copyright (c) 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/  
Copyright (c) 2006 The Apache Software Foundation, http://www.apache.org/  
  
Benchmarking www.kramse.dk (be patient)...  
...
```

Der findes specialiserede værktøjer til mange protokoller

Eksempelvis følger der et apache benchmark med Apache HTTPD serveren

Mange andre værktøjer til at simulere flere samtidige brugere

Apache Benchmark output - 1

```
Server Software:      Apache
Server Hostname:      www.kramse.dk
Server Port:          80

Document Path:        /
Document Length:       7547 bytes

Concurrency Level:     1
Time taken for tests:  13.84924 seconds
Complete requests:     100
Failed requests:        0
Write errors:           0
Total transferred:     778900 bytes
HTML transferred:      754700 bytes
Requests per second:   7.64 #/sec (mean)
Time per request:      130.849 ms (mean)
Time per request:      130.849 ms (mean, across all concurrent requests)
Transfer rate:         58.08 Kbytes/sec received
```

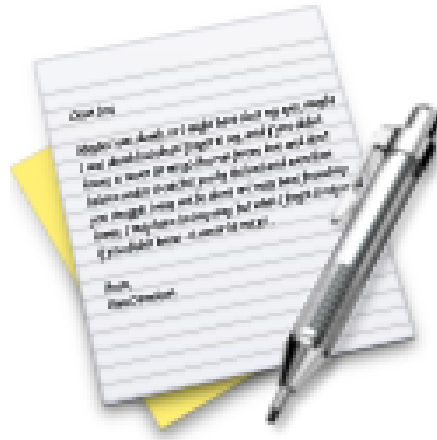
Connection Times (ms)

	min	mean+/-sd	median	max
Connect:	22	24 4.0	24	58
Processing:	96	105 33.0	99	421
Waiting:	63	71 32.7	65	386
Total:	119	130 33.5	124	446

Percentage of the requests served within a certain time (ms)

50%	124
66%	126
75%	128
80%	130
90%	143
95%	153
98%	189
99%	446
100%	446 (longest request)

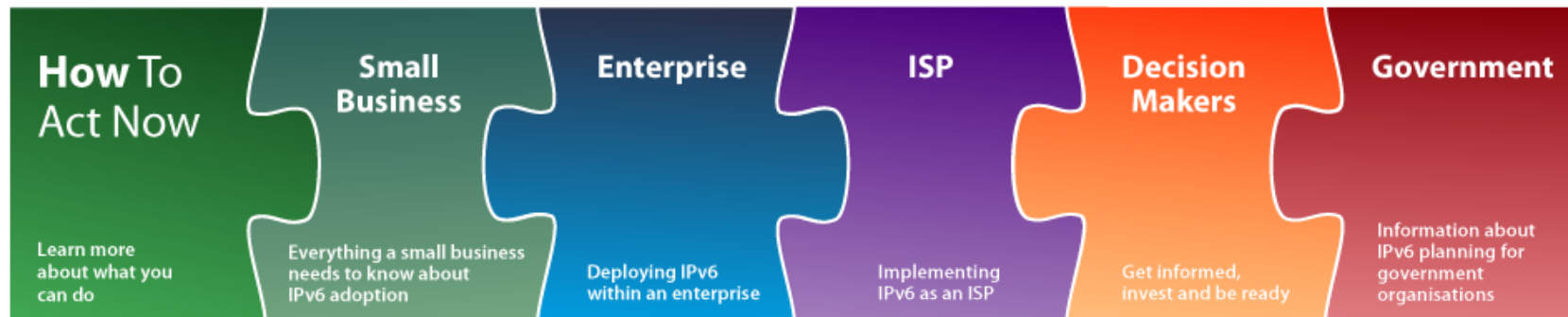
ab er godt til små test, i praksis er Tsung, Jmeter m.fl. bedre



Vi laver nu øvelsen

Performance tool - iperf

som er øvelse **14** fra øvelseshæftet.



<http://www.ipv6actnow.org>

Guidelines for the Secure Deployment of IPv6, SP800-119, NIST

<http://csrc.nist.gov/publications/nistpubs/800-119/sp800-119.pdf>

<http://www.ripe.net>

Preparing an IPv6 Addressing Plan Manual, Surfnet

This presentation ☺

Book: *IPv6 Network Administration* af David Malone og Niall Richard Murphy

Book: *The Second Internet: Reinventing Computer Networks with IPv6*, Lawrence E. Hughes, October 2010,

<http://www.secondinternet.org/>

Book: *Understanding IPv6: Your Essential Guide to IPv6 on Windows Networks* Joseph Davies Microsoft Press; 3 edition (June 27, 2012)

devices - what is a network device?

switches - Layer 2 does not matter much, management by RFC-1918 IPv4 is probably wise

routers - most important, connectivity MUST support IPv6. Check vendor home page - do NOT assume support is ready

Security devices: firewalls, IDS/IPS, VPN - critical and support in general poor. Some vendors such as Cisco ASA and Juniper SRX has good support

You have plenty!

Providers and LIRs will typically get /32

Providers will typically give organisations /48 or /56

Your /48 can be used for:

- 65536 subnets - all host subnets are /64
- Each subnet has 2^{64} addresses

Preparing an IPv6 Addressing Plan Manual

December 2010: Original text

March 2011: Translation provided by RIPE NCC

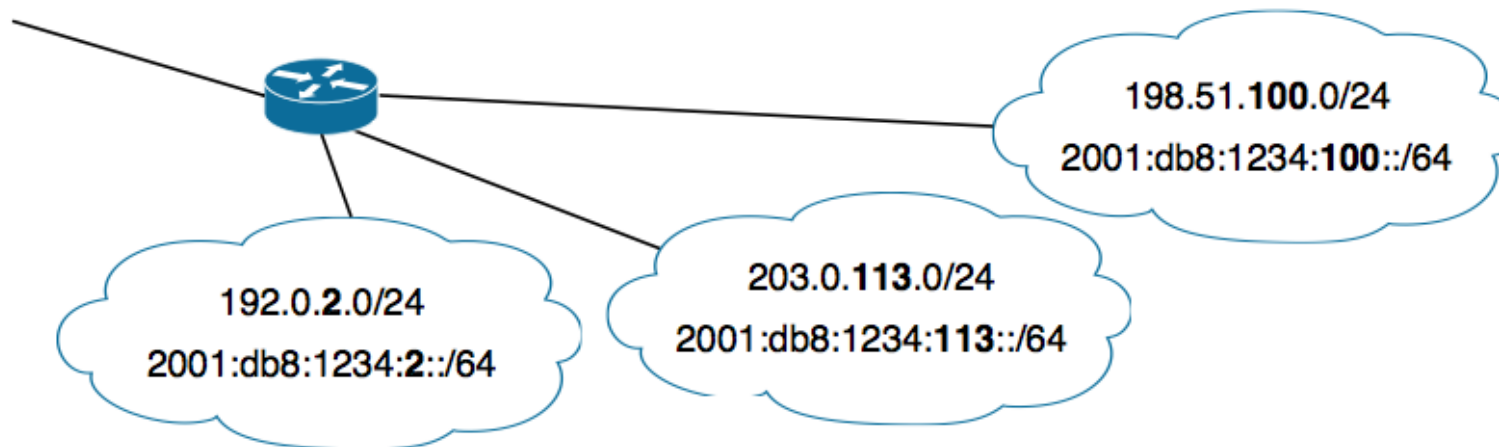


http://www.ripe.net/training/material/IPv6-for-LIRs-Training-Course/IPv6_addr_plan4.pdf

3.2 Direct Link Between IPv4 and IPv6 Addresses

If the existing IPv4 networks use only /24 subnets (for example, from 203.0.113.0 to 203.0.113.255), a direct link can be established between IPv4 addresses and the new IPv6 addresses. In this case, you can include the penultimate number of the IPv4 address (113 in 203.0.113.0/24, for example) in the IPv6 subnet. The IPv6 address will then be 2001:db8:1234:113::/64.

Such an IPv4-to-IPv6 transition could appear as follows:



Easy and coupled with VLAN IDs it will work 😊

Make sure you establish IPv6 in **production**

Enabling service on IPv6 without production - bad experience for users

Start by enabling your DNS servers for IPv6 - and DNSSEC - and DNS over TCP
Remember that your firewall might have problems with large DNS packets

Add a production IPv6 router - hardware device or generic server

Tunnels are OK, and SixXS consider their service production

Yes, IPv6 software has flaws, and Type 0 routing header is deprecated *RFC 5095 - Deprecation of Type 0 Routing Headers in IPv6*

NOTE: a lot of problems are similar to IPv4, mac overflow in IPv4 or IPv6 is quite similar IMHO

Attack toolkits, testing devices

`http://www.thc.org/thc-ipv6/`

`http://www.si6networks.com/research/tools.html`

`http://lists.si6networks.com/pipermail/ipv6hackers/`

IPv6 hackers meeting, video: `http://www.youtube.com/user/deploy360`

`http://www.insinuator.net/2013/08/ipv6-hackers-meeting-ietf-87-in-berlin-slides/`

A complete tool set to attack the inherent protocol weaknesses of IPV6 and ICMP6, and includes an easy to use packet factory library.

Last update 2012-01-15 - opdateres løbende

Current public version: v1.8 - CCC Camp release

<http://thc.org/thc-ipv6/>

- - parasite6: icmp neighbor solicitation/advertisement spoofer, puts you as man-in-the-middle, same as ARP mitm (and parasite)
- - alive6: an effective alive scanning, which will detect all systems listening to this address
- - dnsdict6: parallelized dns ipv6 dictionary bruteforcer
- - fake_router6: announce yourself as a router on the network, with the highest priority
- - redir6: redirect traffic to you intelligently (man-in-the-middle) with a clever icmp6 redirect spoofer
- - toobig6: mtu decreaser with the same intelligence as redir6
- - detect-new-ip6: detect new ip6 devices which join the network, you can run a script to automatically scan these systems etc.
- - dos-new-ip6: detect new ip6 devices and tell them that their chosen IP collides on the network (DOS).
- - trace6: very fast traceroute6 with supports ICMP6 echo request and TCP-SYN
- - flood_router6: flood a target with random router advertisements

- - flood_advertise6: flood a target with random neighbor advertisements
- - exploit6: known ipv6 vulnerabilities to test against a target
- - denial6: a collection of denial-of-service tests againsts a target
- - fuzz_ip6: fuzzer for ipv6
- - implementation6: performs various implementation checks on ipv6
- - implementation6d: listen daemon for implementation6 to check behind a fw
- - fake_mld6: announce yourself in a multicast group of your choice on the net
- - fake_mld26: same but for MLDv2
- - fake_mldrouter6: fake MLD router messages
- - fake_mipv6: steal a mobile IP to yours if IPSEC is not needed for authentication
- - fake_advertiser6: announce yourself on the network
- - smurf6: local smurfer
- - rsmurf6: remote smurfer, known to work only against linux at the moment
- - sendpees6: a tool by willdamn(ad)gmail.com, which generates a neighbor solicitation requests with a lot of CGAs (crypto stuff ;-) to keep the CPU busy. nice.

- - thcping6: sends a hand crafted ping6 packet
- and about 15 more tools for you to discover



Danish IPv6 Task Force

Danish IPv6 task force - unofficial <http://www.ipv6tf.dk>

Husk følgende:

- Unix og Linux er blot eksempler - navneservice eller HTTP server kører fint på Windows
- DNS er grundlaget for Internet, specielt for IPv6
- Sikkerheden på internet er generelt dårlig, brug SSL!
- Man skal *hærde* operativsystemer *før* man sætter dem på Internet
- God sikkerhed kommer fra langsigtede initiativer

Jeg håber I har lært en masse om netværk og kan bruge det i praksis :-)

Henrik Lund Kramshøj, internet samurai
hlk@solido.net

`http://www.solidonetworks.com`

I er altid velkomne til at sende spørgsmål på e-mail

- *Network Warrior, 2nd Edition Everything you need to know that wasn't on the CCNA exam* af Gary A. Donahue Publisher: O'Reilly Media Released: May 2011
- *IPv6 Network Administration* af David Malone og Niall Richard Murphy - god til real-life admins, typisk O'Reilly bog
- *Understanding IPv6: Your Essential Guide to IPv6 on Windows Networks* Joseph Davies Microsoft Press; 3 edition (June 27, 2012)

- Stevens, Comer klassiske bøger om TCP/IP
- KAME bøgerne om IPv6 protokollerne, meget detaljerede
- O'Reilly cookbooks: Cisco, BIND og Apache HTTPD m.fl.
- Cisco Press og website
- Juniper website specielt Day One booklets
- Firewall bøger Cheswick
- Der findes mange gode bøger om netværk

IPv6 Essentials af Silvia Hagen, O'Reilly 2nd edition (May 17, 2006) god reference om emnet - men ca. samme indhold som RFC'erne

IPv6 Core Protocols Implementation af Qing Li, Tatuya Jinmei og Keiichi Shima - detaljeret information om implementation

IPv6 Advanced Protocols Implementation af Qing Li, Jinmei Tatuya og Keiichi Shima
- flere andre