Welcome to

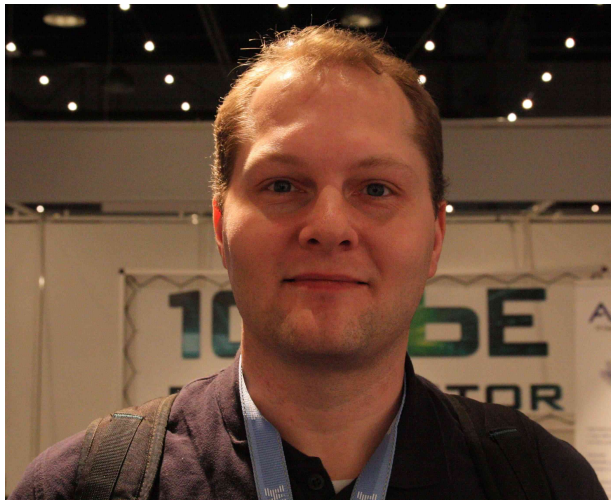
IPv6 Security Workshop

Marts 2012

Henrik Lund Kramshøj
hlk@solido.net

http://www.solidonetworks.com

Contact information



- Henrik Lund Kramshøj, IT-security and internet samurai
- Email: hlk@solido.net Mobile: +45 2026 6000
- Educated from the Computer Science Department at the University of Copenhagen, DIKU
- CISSP and CEH certified
- 2003 - 2010 Independent security consultant
- 2010 - owner and partner in Solido Networks ApS

Planen idag

Don’t Panic!

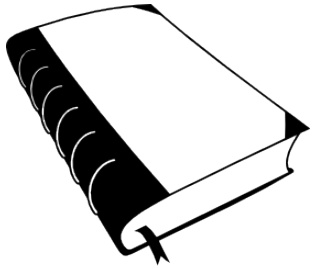
Kl 12-15

Mindre foredrag mere snak

Mindre enetale, mere foredrag 2.0 med socialt medie, informationsdeling og interaktion

Send gerne spørgsmål senere

Kursusmateriale



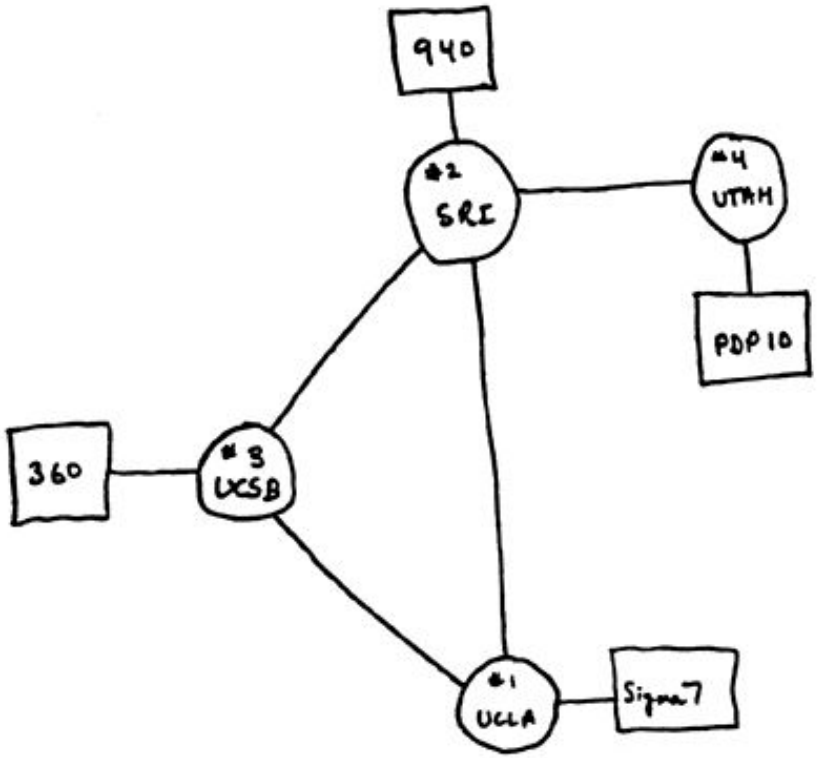
Dette materiale består af flere dele:

- Kursusmaterialet - præsentationen til undervisning - dette sæt
- Øvelseshæfte med øvelser

Hertil kommer diverse ressourcer fra internet

Boot DVD'er baseret på BackTrack Linux - image udleveres

Formål: netværkssikkerhed for TCP/IP netværk



TCP/IP-baserede netværk - internet er overalt

Formål: mere specifikt

At introducere TCP/IP version 6

Introducere specifikke sikkerhedsproblemer ved brug af IPv6

Forudsætninger

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

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

De fleste øvelser kan udføres fra Windows 7

Øvelserne foregår via

- Login til Unix maskinen
- Direkte fra jeres systemer Windows eller Linux boot DVD/Virtuel server

Kursusfaciliteter

Der er opbygget et kursusnetværk med følgende primære systemer:

- Unix server Fiona med HTTP server og værktøjer
- Unix boot DVD'er eller VMware images - jeres systemer

På Unix serveren tillades login diverse 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**

Hackerværktøjer

Der benyttes en del værktøjer:

- 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/`
- Apache Tomcat J2EE servlet container `http://tomcat.apache.org`
- OpenBSD operativsystem med fokus på sikkerhed `http://www.openbsd.org`

BackTrack 5 og sniffer programmer

Wireshark - `http://www.wireshark.org` avanceret netværkssniffer
bruger vi til at sniffe, vi bruger Wireshark til primære demo, nævner Ettercap osv.

BackTrack `http://www.backtrack-linux.org/` BackTrack er baseret på Linux og må kopieres frit :-)

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**

Mangler der strømkabler

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

Status idag på internet

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/`

Why IPv6

DDoS udviklingen, januar 2010 rapporten

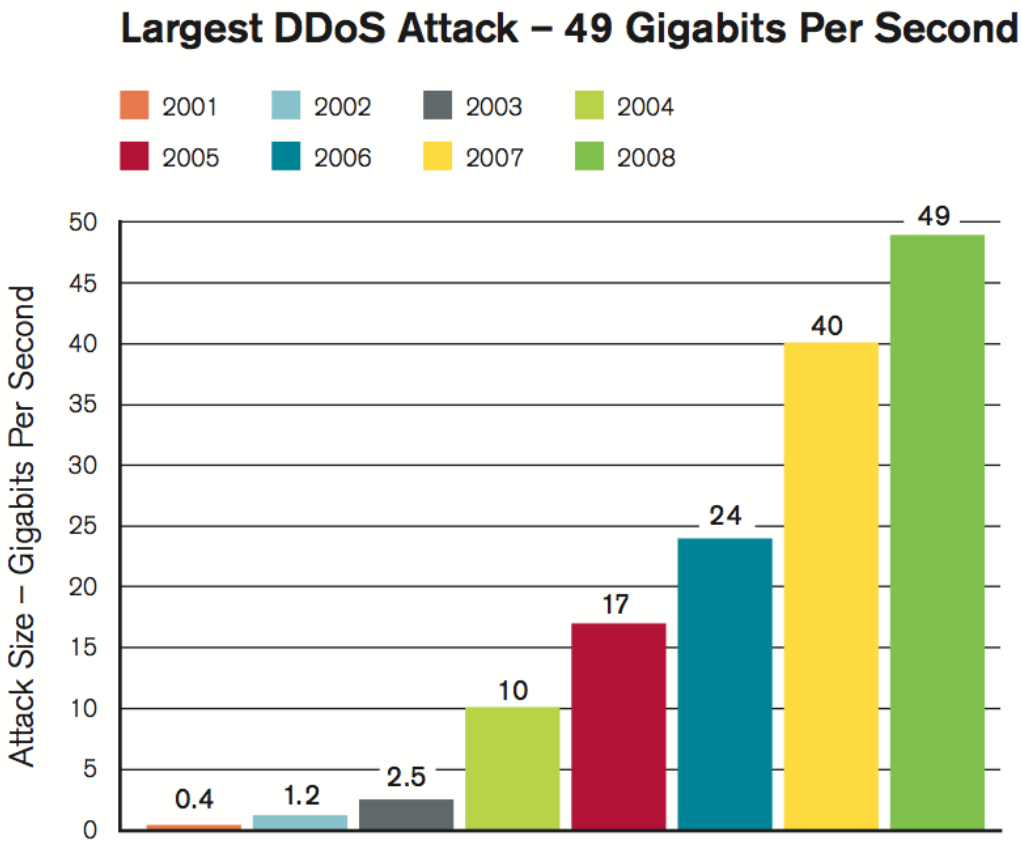


Figure 1: *Largest DDoS Attack – 49 Gigabits Per Second*

Source: Arbor Networks, Inc.

Kilde: `http://www.arbornetworks.com/report` 2009 rapporten

DDoS udviklingen, februar 2011

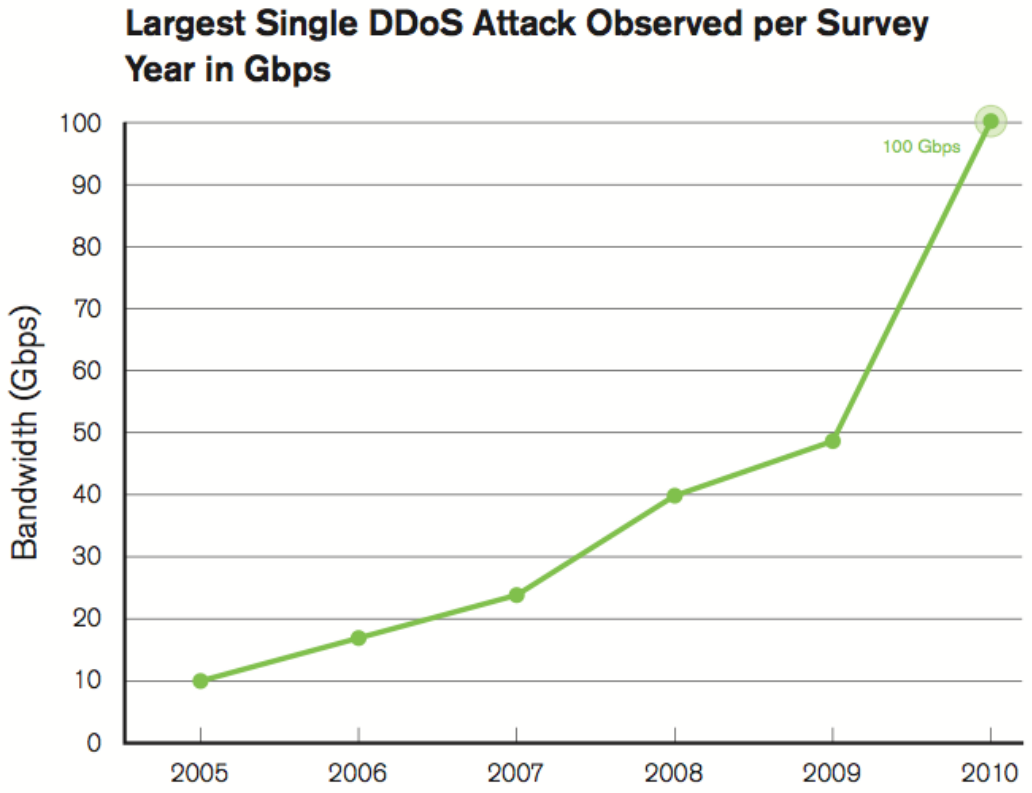


Figure 1
Source: Arbor Networks, Inc.

Kilde: `http://www.arbornetworks.com/report` 2010 rapporten

Key finding:
Application-Layer DDoS Attacks Are Increasing in Sophistication and Operational Impact
IDC and mobile/fixed wireless operators in particular are reporting significant outages, increased OPEX, customer churn and revenue loss due to application-layer DDoS attacks. These attacks are targeting both their customers and their own ancillary supporting services, such as DNS, Web portals, etc.
...
Conclusions
We further note that the fastest-growing category of ISPs-mobile and fixed wireless broadband operators-are also the least-prepared organizations in terms of network visibility, network control, and overall ability to successfully defend themselves and their customers against attack. These operators are balancing an overwhelming array of threats with conflicting budget pressures and business objectives.

Mere komplekse trusler, betyder det flere firewall?

Worldwide Infrastructure Security Report 2010 Volume V

- Application-Layer DDoS Attacks Are Increasing in Sophistication and Operational Impact
- Mobile/Fixed Wireless Operators Are Facing Serious Challenges to Maintaining Availability in the Face of Attacks
- Firewalls and IPS Devices Are Falling Short on DDoS Protection
- DNS Has Broadly Emerged as an Attack Target and Enabler
- **Lack of Visibility into and Control over IPv6 Traffic Is a Significant Challenge**
- Chronic Underfunding of Operational Security Teams
- Operators Continue to Express Low Confidence in the Efficacy of Law Enforcement
- Operators Have Little Confidence in Government Efforts to Protect Critical Infrastructure

Kilde: `http://www.arbornetworks.com/report`

Worldwide Infrastructure Security Report 2011 Volume VI

- Ideologically-Motivated ”Hactivism” and Vandalism Are the Most Readily-Identified DDoS Attack Motivations
- 10 Gbps and Larger Flood-Based DDoS Attacks Are the ”New Normal”
- Increased Sophistication and Complexity of Application-Layer (Layer 7) DDoS Attacks and Multi-Vector DDoS Attacks Are Becoming More Common
- Visibility and Security of Mobile and Fixed Wireless Networks Are an Ongoing Concern
- **First-Ever Reports of IPv6 DDoS Attacks ”in the Wild” on Production Networks**
- **Rarity of IPv6-Enabled Attacks Indicates Low IPv6 Market Penetration and Lack of Critical Mass**
- Stateful Firewalls, IPS and Load-Balancer Devices Continue to Fall Short on DDoS Protection Capabilities
- The Overwhelming Majority of Network Operators Do Not Engage Law Enforcement

Kilde: `http://www.arbornetworks.com/report`

IPv6 is coming

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`

Implications

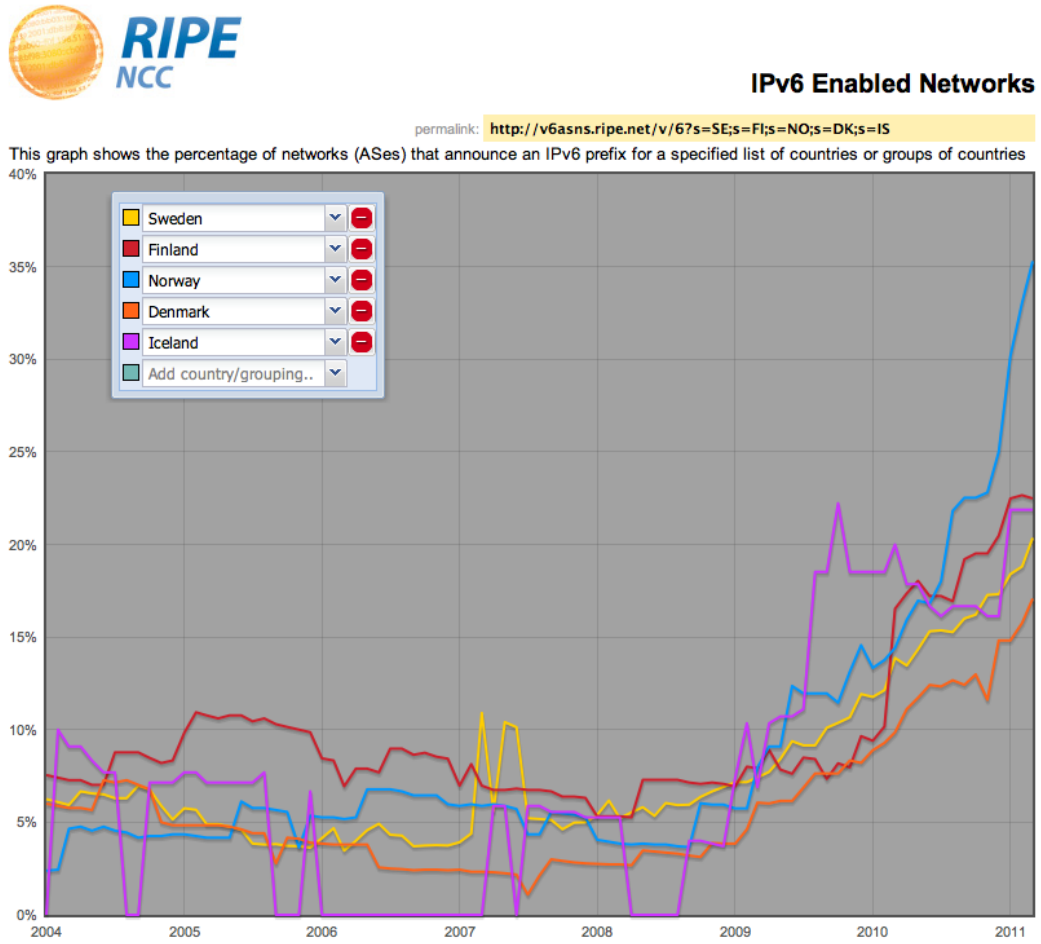
For an IPv4 enterprise network, the existence of an IPv6 overlay network has several of implications:

- 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 in the Nordic region



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

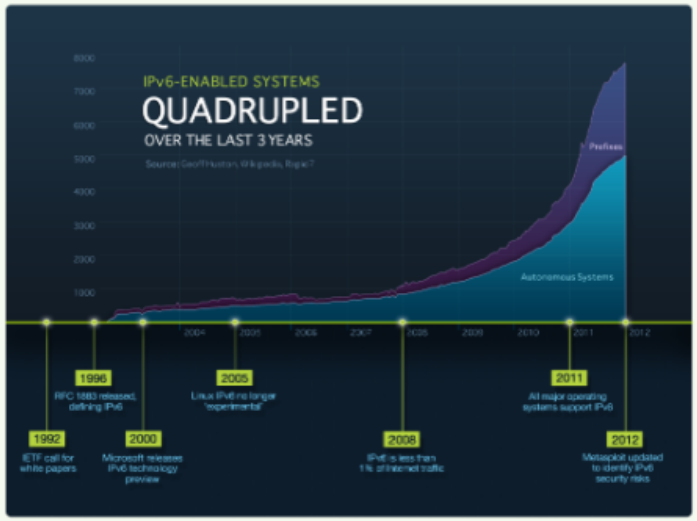
Metasploit IPv6

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:

Kilde:

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

New Loki - ikke alt er layer 7

SOLIDO NETWORKS

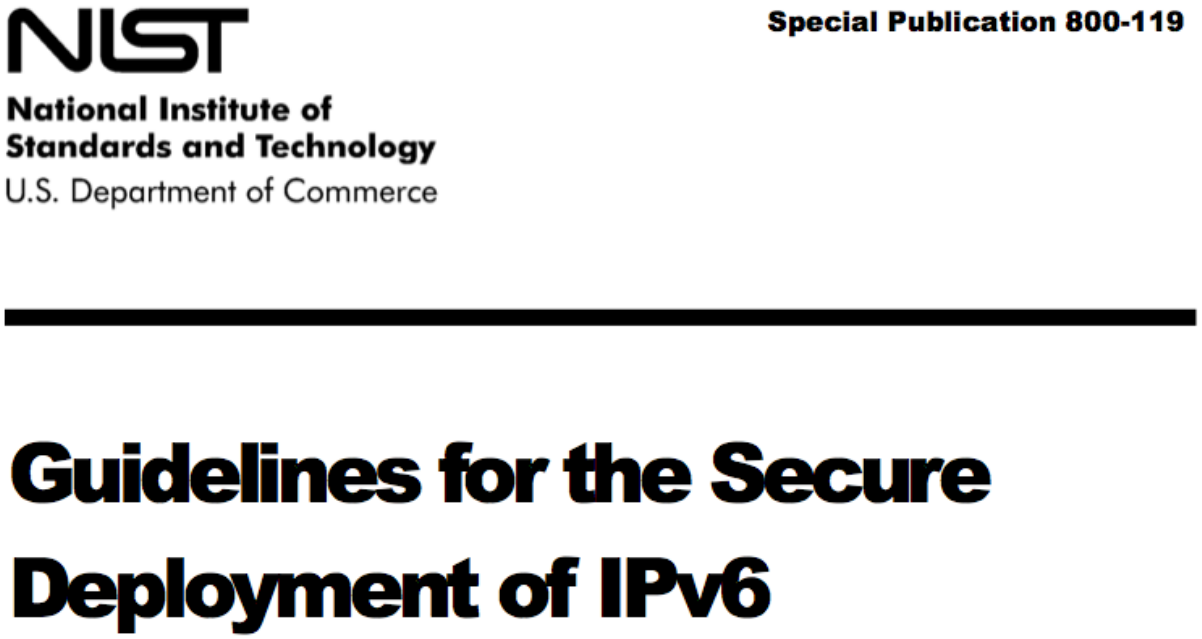
At the beginning LOKI was made to combine some stand-alone **command line tools**, like the `bgp_cli`, the `ospf_cli` or the `ldp_cli` and to give them a **user friendly, graphical interface**. In the meantime LOKI is more than just the combination of the single tools, it gave its modules the opportunity to base upon each other (like **combining ARP-spoofing from the ARP module with some man-in-the-middle actions, rewriting MPLS-labels for example)** and even inter operate with each other. (Fremhævning: HLK)

`https://www.c0decafe.de/loki.html` Loki

`http://www.packetstan.com/2011/02/running-loki-on-backtrack-4-r2.html`

”Yersinia is a network tool designed to take advantage of some weakeness in different network protocols. It pretends to be a solid framework for analyzing and testing the deployed networks and systems.” `http://www.yersinia.net/`

NIST Special Publication 800 series



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`

THC

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/`

THC IPv6 [0x03] The Included Tools

SOLIDO NETWORKS

- - parasite6: icmp neighbor solitication/advertisement spoofer, puts you as man-in-the-middle, same as ARP mitm (and parasite)
- - alive6: an effective alive scanng, which will detect all systems listening to this address
- - dnsdict6: parallized 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

www.solidonetworks.com

hlk@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
```

IPv6 Status Denmark

IT- og Telestyrelsen are becoming more active

Unofficial IPv6 task force at `http://www.ipv6tf.dk/`

Other initiatives `http://world-ipv6-day.dk/`

Major providers are ready on back bones

Internet Providers are increasingly becoming ready

Collect information about IPv6

Guidelines for the Secure Deployment of IPv6, SP800-119, NIST
`http://csrc.nist.gov/publications/nistpubs/800-119/sp800-119.pdf`

The Second Internet: Reinventing Computer Networks with IPv6, Lawrence E. Hughes, October 2010,
`http://www.secondinternet.org/`

IPv6 Network Administration af David Malone og Niall Richard Murphy

`http://www.ripe.net`

This presentation ☺

Allocating IPv6 addresses

SOLIDO NETWORKS

You have plenty!

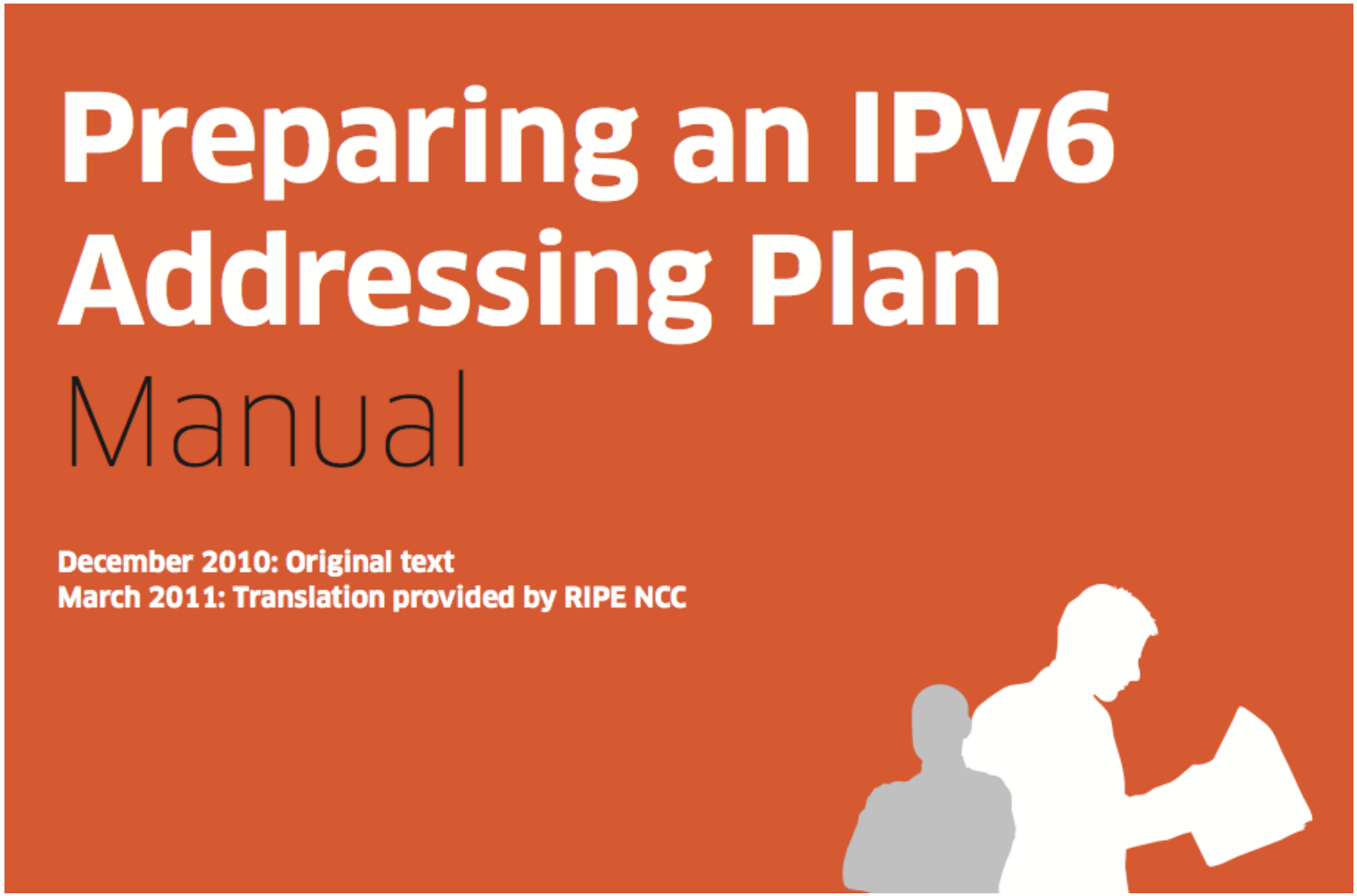
Providers and LIRs will typically get /32

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



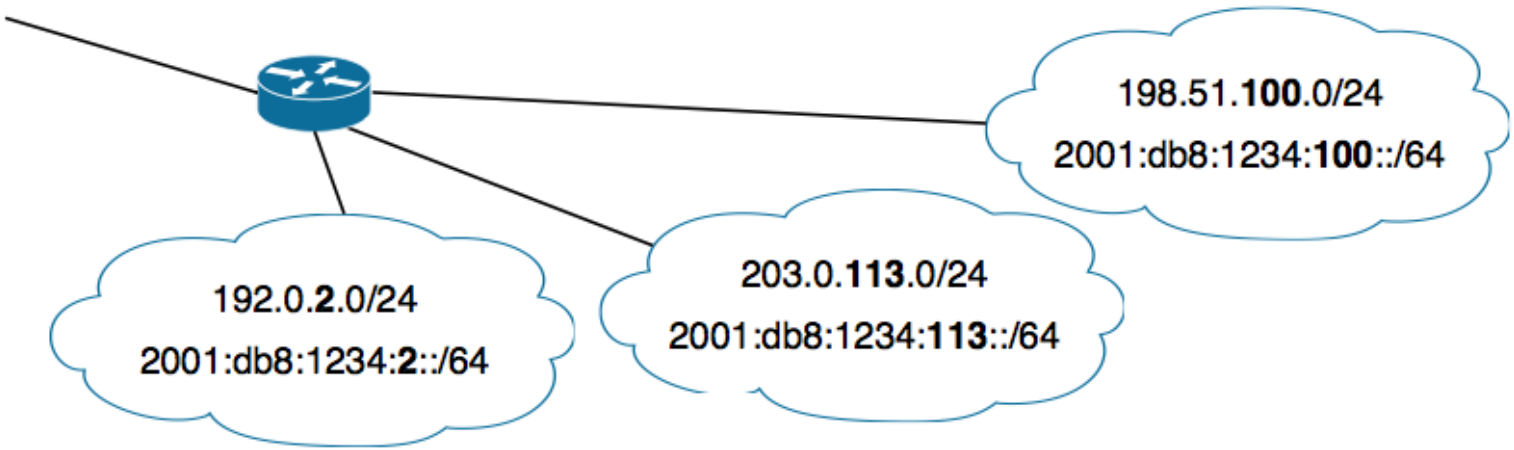
http://www.ripe.net/training/material/IPv6-for-LIRs-Training-Course/IPv6_addr_plan4.pdf

Example adress plan input

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 ☺

Run IPv6 in production

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

IPv6 business case

SOLIDO NETWORKS

- 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!

SOLIDO NETWORKS

Preserve the good stuff

back to basics, internet as it used to be!

fate 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!

Up and running with IPv6

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 ☺

Unix starthjælp

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`

Aftale om test af netværk

Straffelovens paragraf 263 Stk. 2. Med bøde eller fængsel indtil 6 måneder straffes den, som uberettiget skaffer sig adgang til en andens oplysninger eller programmer, der er bestemt til at bruges i et anlæg til elektronisk databehandling.

Hacking kan betyde:

- At man skal betale erstatning til personer eller virksomheder
- At man får konfiskeret sit udstyr af politiet
- At man, hvis man er over 15 år og bliver dømt for hacking, kan få en bøde - eller fængselsstraf i alvorlige tilfælde
- At man, hvis man er over 15 år og bliver dømt for hacking, får en plettet straffeattest. Det kan give problemer, hvis man skal finde et job eller hvis man skal rejse til visse lande, fx USA og Australien
- Frit efter: `http://www.stophacking.dk` lavet af Det Kriminalpræventive Råd
- Frygten for terror har forstærket ovenstående - så lad være!

Agenda Basale IPv6 begreber

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

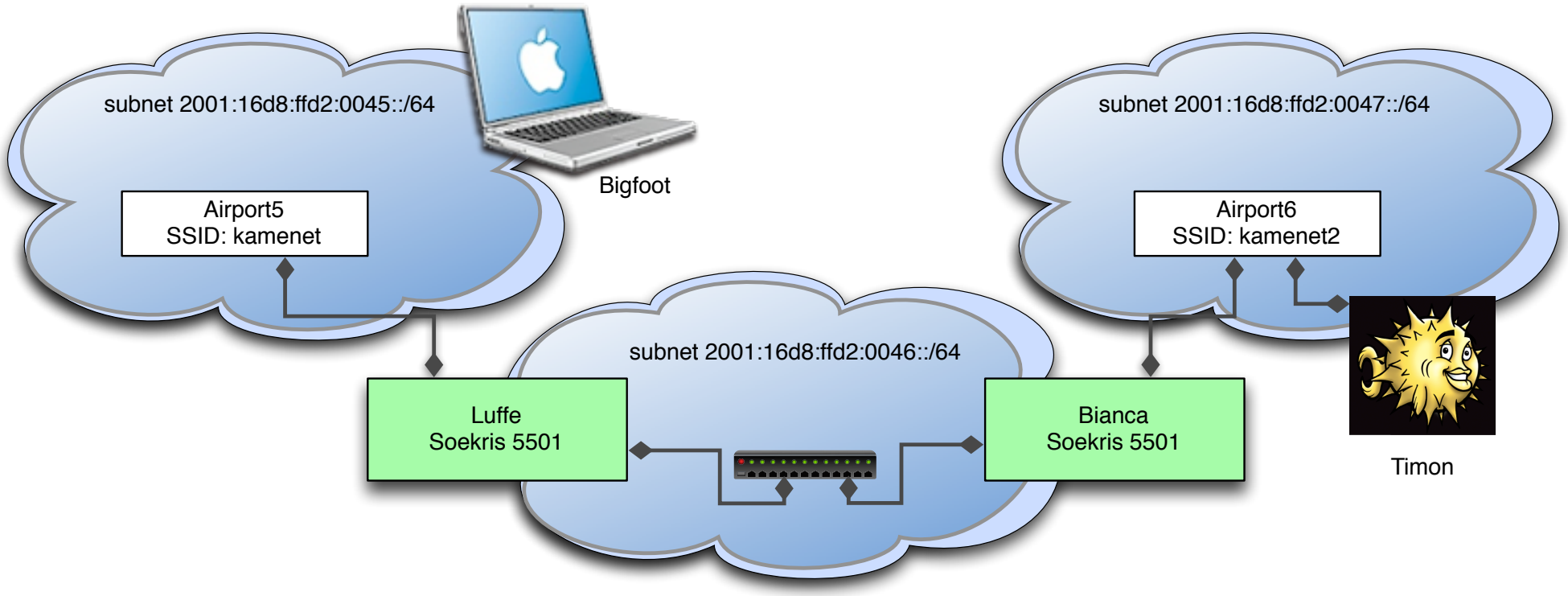
Basale værktøjer traceroute, ping, dig, host

Wireshark sniffer

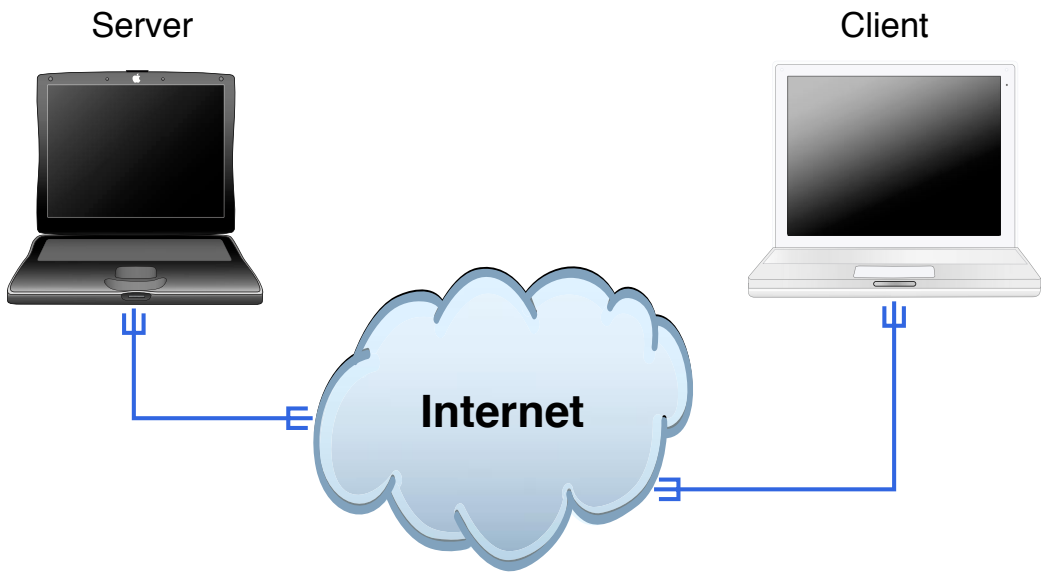
VLAN 802.1q

THC IPv6 attack toolkit

Netværk til routning



Internet today



Clients and servers

Rooted in academic networks

Protocols which are more than 20 years old, moved to TCP/IP in 1981

Hvad er Internet

Kommunikation mellem mennesker!

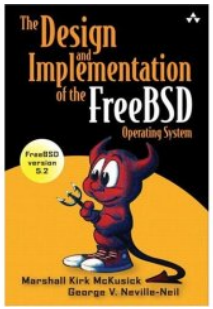
Baseret på TCP/IP

- 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.

BSD Unix



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

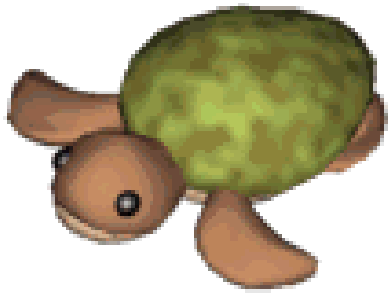
BSD Unix har blandt andet resulteret i virtual memory management og en masse TCP/IP relaterede applikationer

Specielt har BSD TCP/IP kernefunktionalitet været genbrugt mange steder

Tilsvarende genbruges KAME IPv6 implementationen mange steder

`http://en.wikipedia.org/wiki/BSD`

KAME - en IPv6 reference implementation



`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

Hvad er Internet

80’erne IP/TCP starten af 80’erne

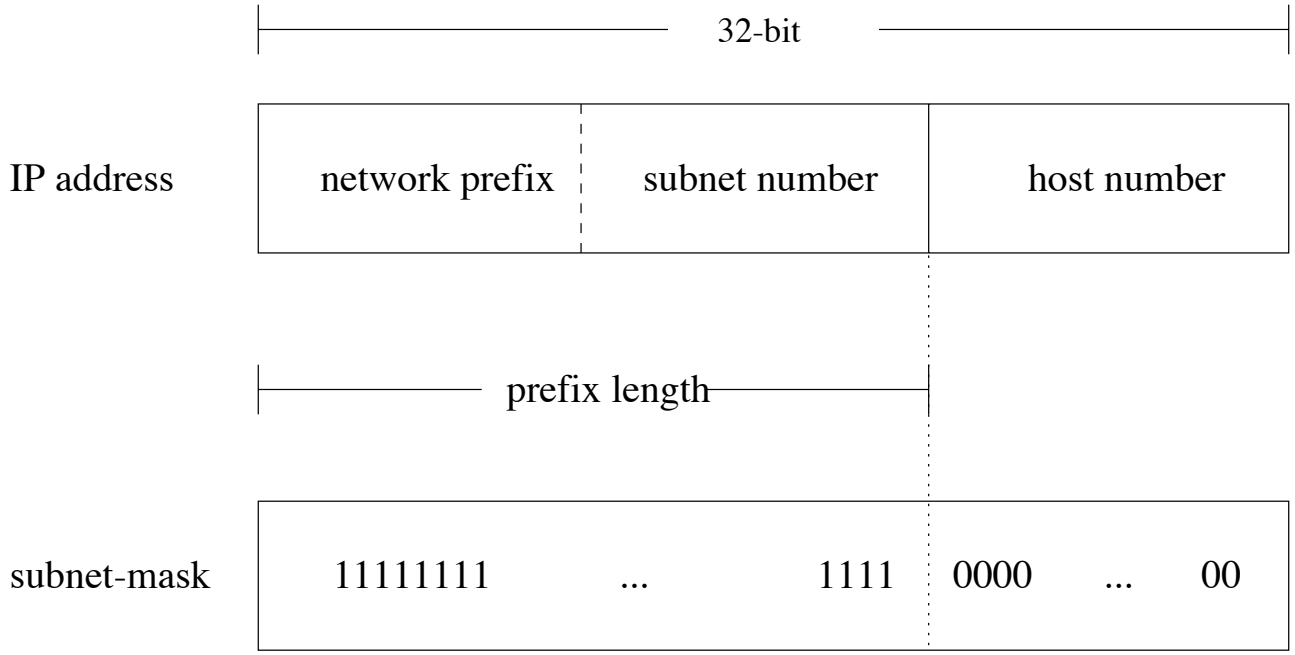
90’erne IP version 6 udarbejdes

- 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 traffik - siden overtog web/http førstepladsen

Fælles adresserum



Hvad kendetegner internet idag

Der er et fælles adresserum baseret på 32-bit adresser

En IP-adresse kunne være 10.0.0.1

IPv4 addresser og skrivemåde

```
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-adresser som bits

```
IP-adresse: 127.0.0.1
Heltal: 2130706433
Binary: 11111110000000000000000000000001
```

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

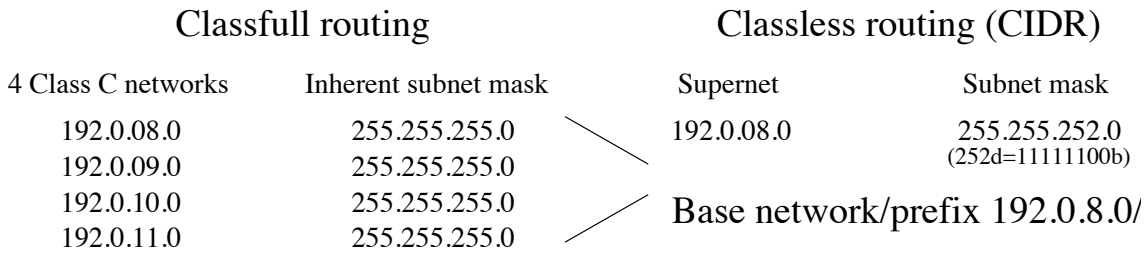
Derfor bad de fleste om adresser i B-klasser - så de var ved at løbe tør!

D-klasse benyttes til multicast

E-klasse er blot reserveret

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

CIDR Classless Inter-Domain Routing



Subnetmasker var oprindeligt indforstået

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

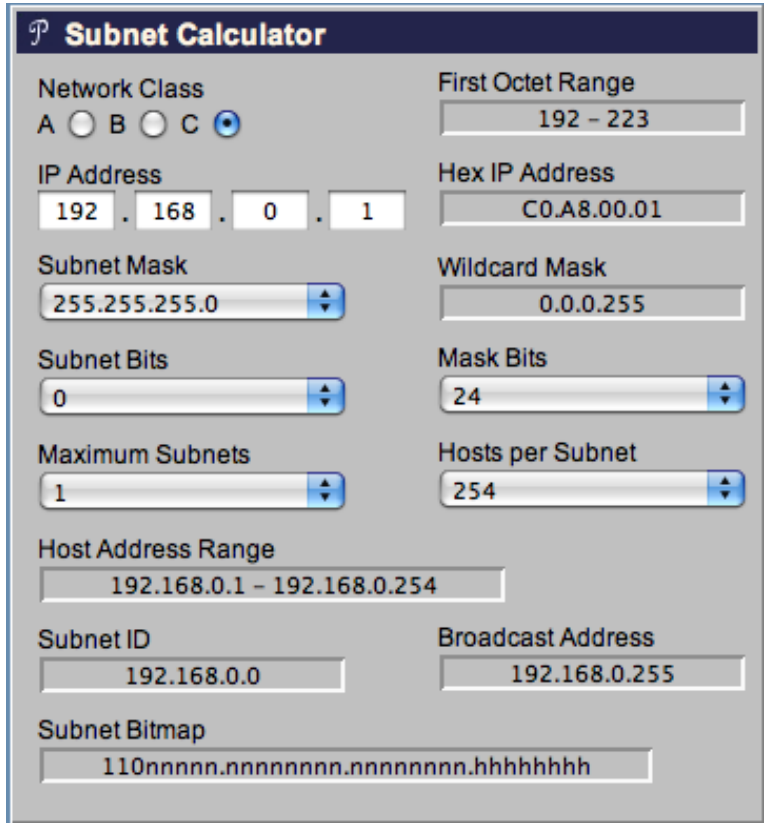
Ved at tildele flere C-klasser kunne man spare de resterende B-klasser - men det betød en routing table explosion

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

Subnet calculator, CIDR calculator



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

Screenshot fra `http://www.subnet-calculator.com/`

RFC-1918 private netværk

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

IPv4 addresser opsummering

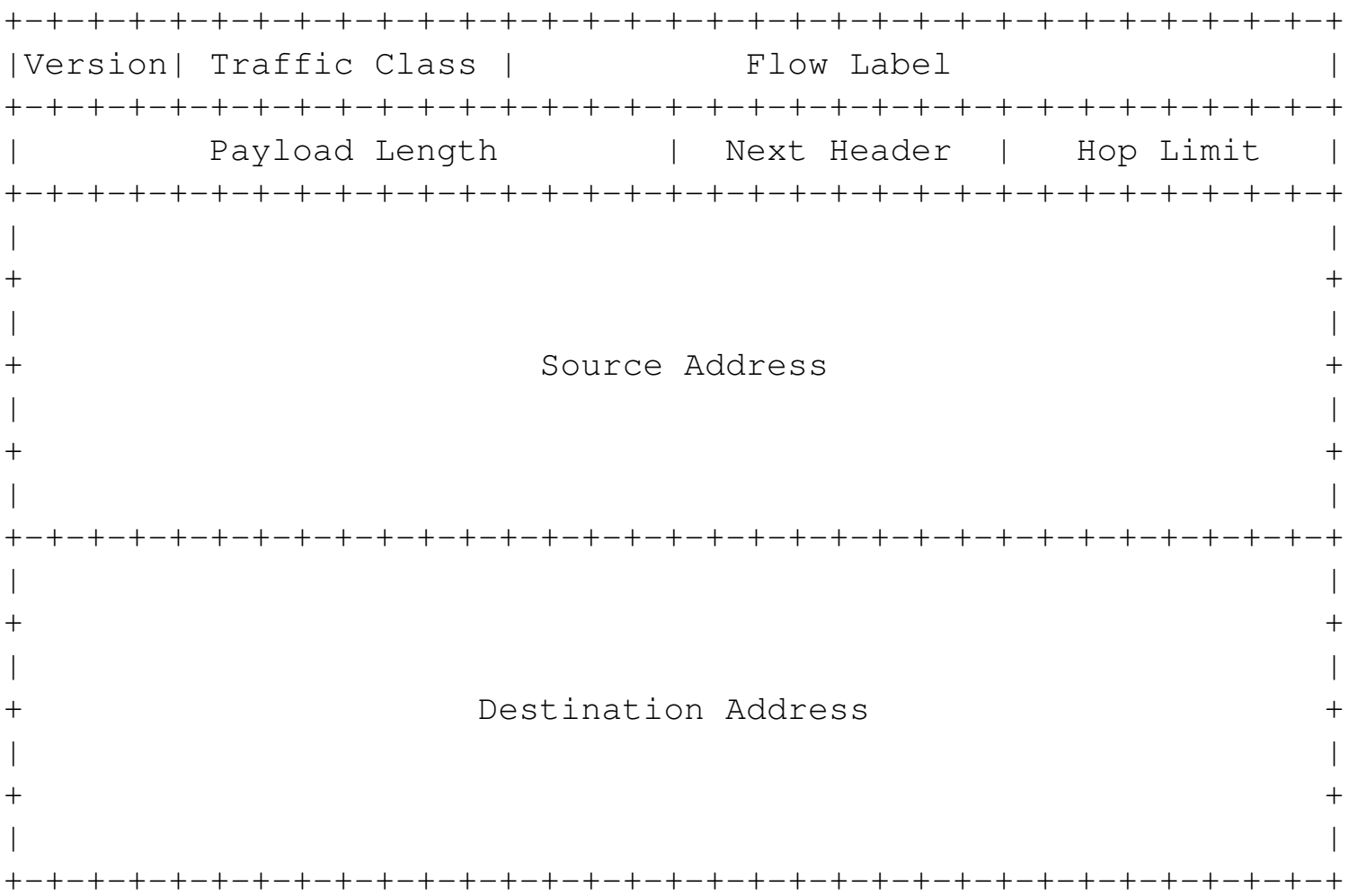
- Altid 32-bit adresser
- Skrives typisk med 4 decimaltal dot notation 10.1.2.3
- Netværk angives med CIDR Classless Inter-Domain Routing RFC-1519
- CIDR notation 10.0.0.0/8 - fremfor 10.0.0.0 med subnet maske 255.0.0.0
- Specielle adresser
  127.0.0.1 localhost/loopback
  0.0.0.0 default route
- RFC-1918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

IPv6 addresser og skrivemåde

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



IPv6 - new and improved

IPv6 - extension headers RFC-2460

- Hop-by-Hop Options
- Routing (Type 0)
- Fragment - fragmentation only at end-points!
- Destination Options
- Authentication
- Encapsulating Security Payload

Note: IPsec (AH and ESP) are mandatory for IPv6 hosts

Path MTU, PMTU implemented larger default MTU, at least 1280 bytes

Fragmentation only at the source host, no router fragmentation

IPv6 addressing RFC-4291

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**.

IPv6 addressing RFC-4291, cont.

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

8 times 4 hex-digits seperated by colon x:x:x:x:x:x:x:x

Written as ipv6-address/prefix-length CIDR notation

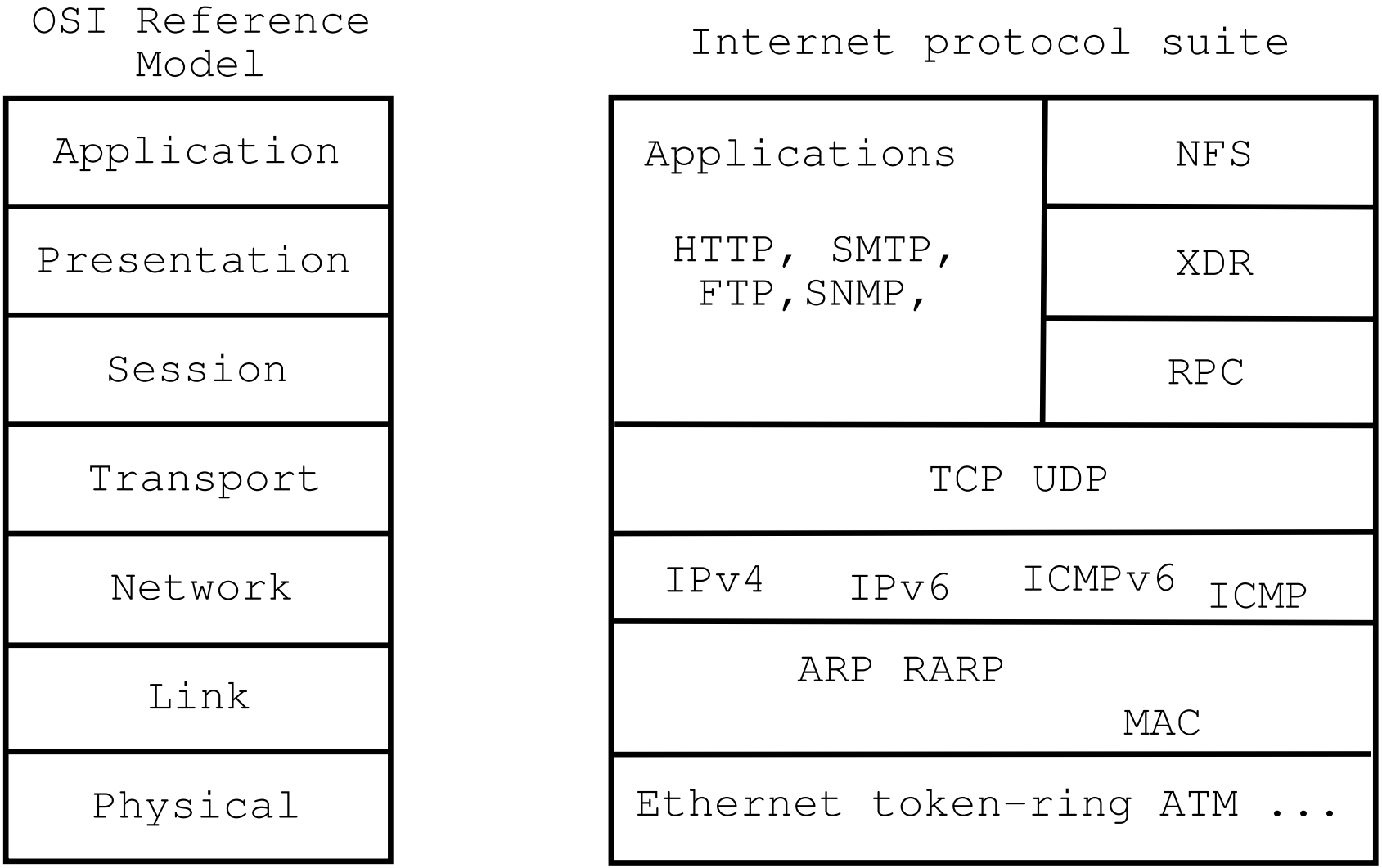
Leading zeros can be removed

One or more groups of 16 bits of zeros can be replaced by ::

IPv6 address - special prefixes

- link-local unicast addresses
  fe80::/10 generated from the interface MAC address EUI-64
- FEC0::/10 site-local - deprecated in RFC-3879
- FC00::/7 Unique Local IPv6 Unicast Addresses RFC-4193
  `http://www.simpledns.com/private-ipv6.aspx`
- 2001:0DB8::/32 NON-ROUTABLE range to be used for documentation purpose RFC-3849.

OSI og Internet modellerne

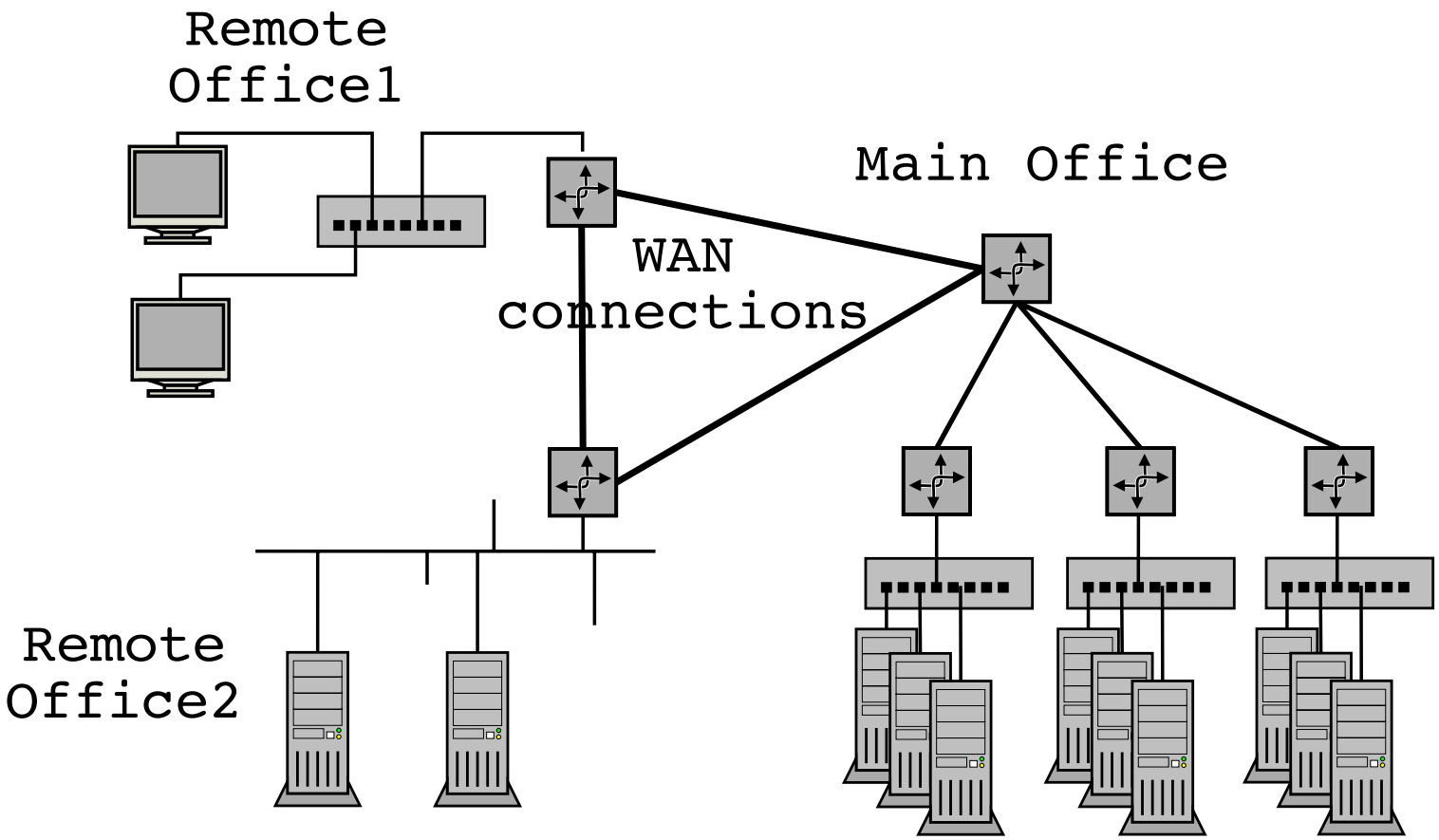


Trådløse teknologier

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

Første dag bruger vi blot trådløse netværk, på dag 2 gennemgår vi 802.11

Broer og routere



Fysisk er der en begrænsing for hvor lange ledningerne må være

Bridges

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

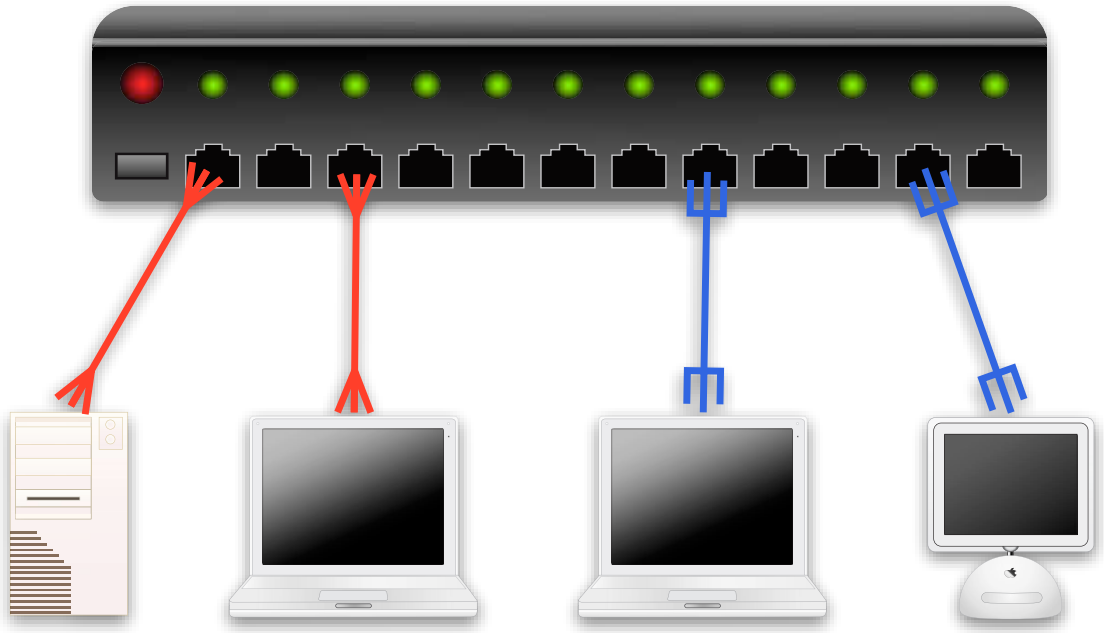
Virker som en forstærker på signalet, men mere intelligent

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`

En switch

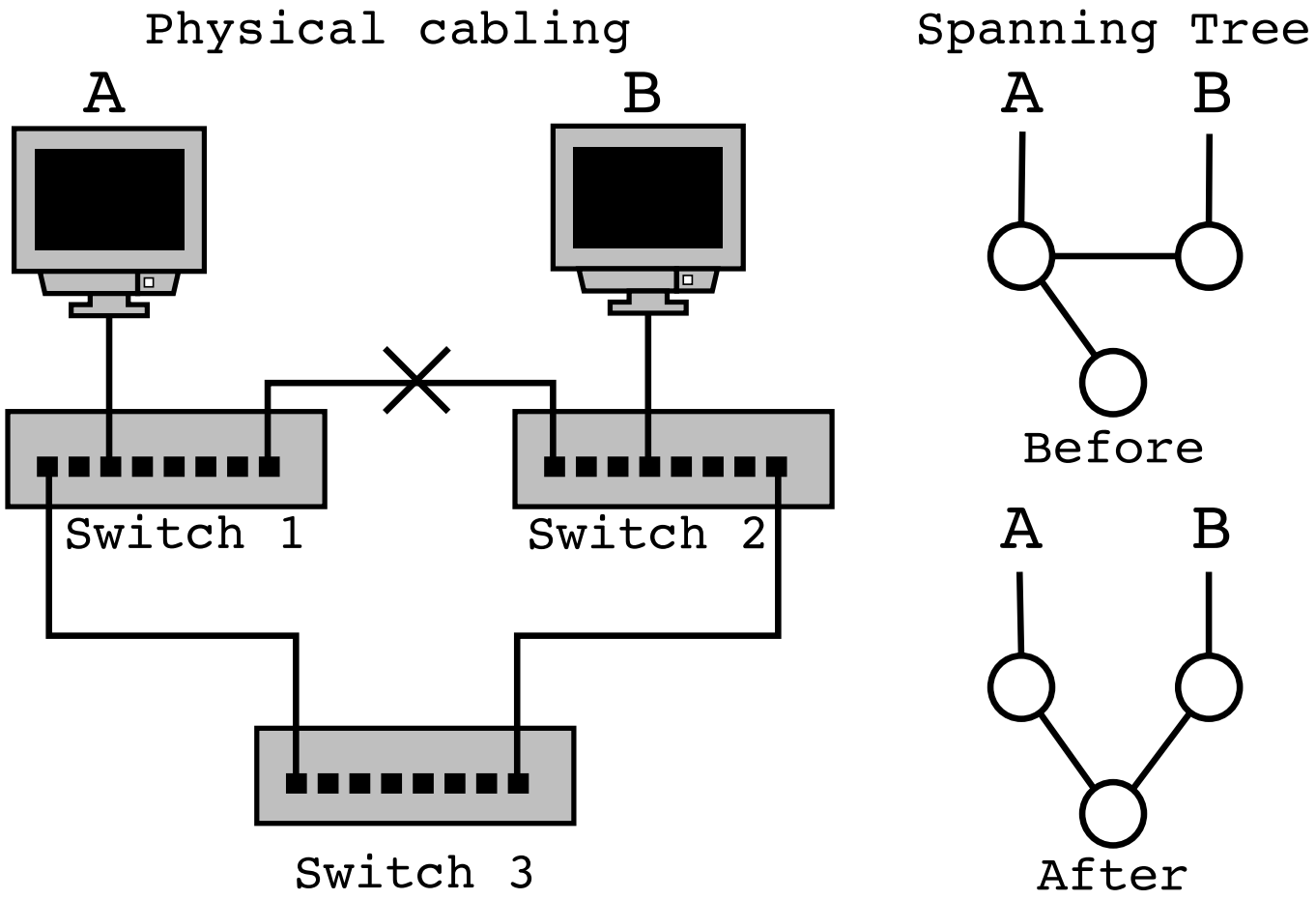


Ved at fortsætte udviklingen kunne man samle broer til en switch

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

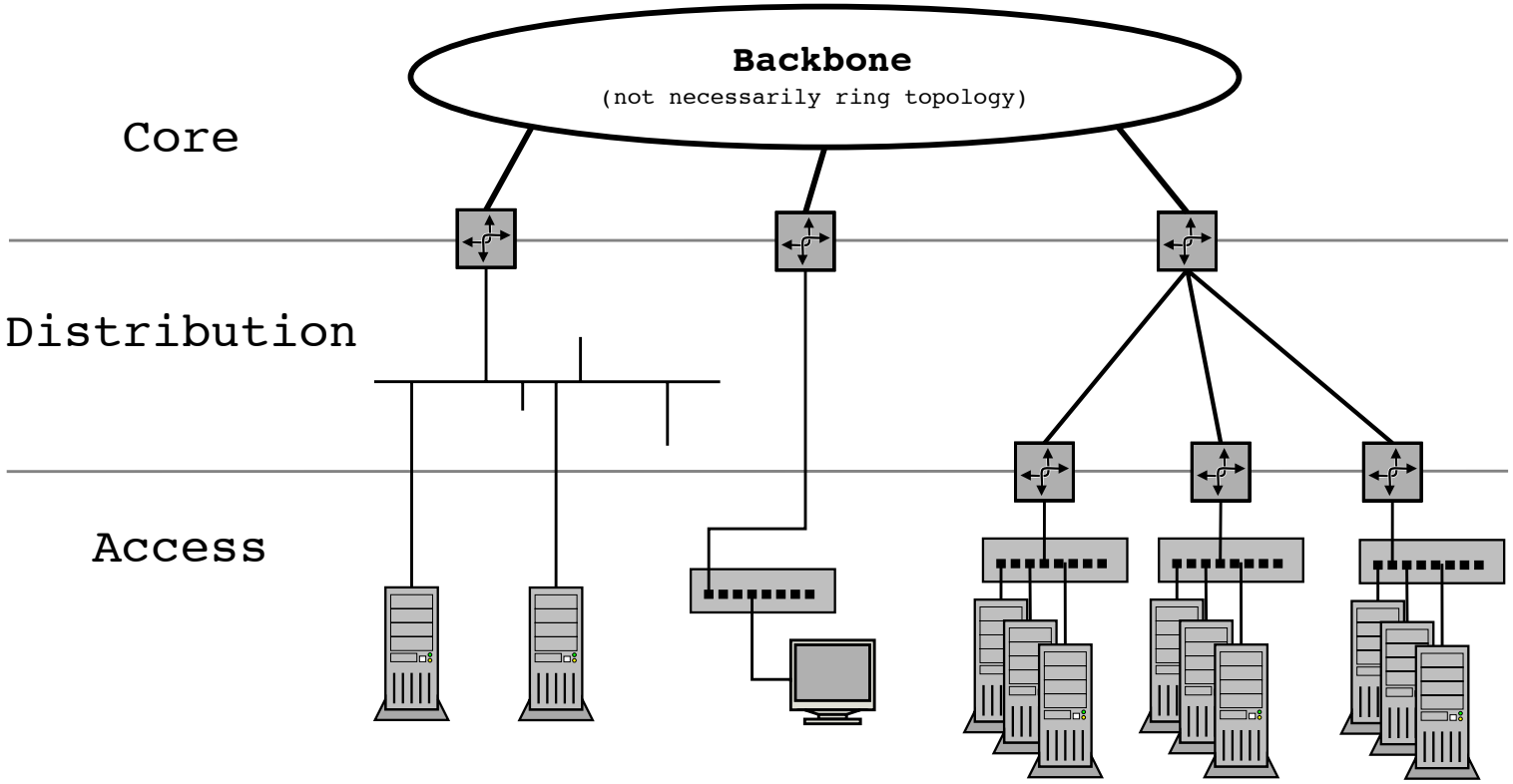
Bemærk performance begrænses af backplane i switchen

Topologier og Spanning Tree Protocol



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

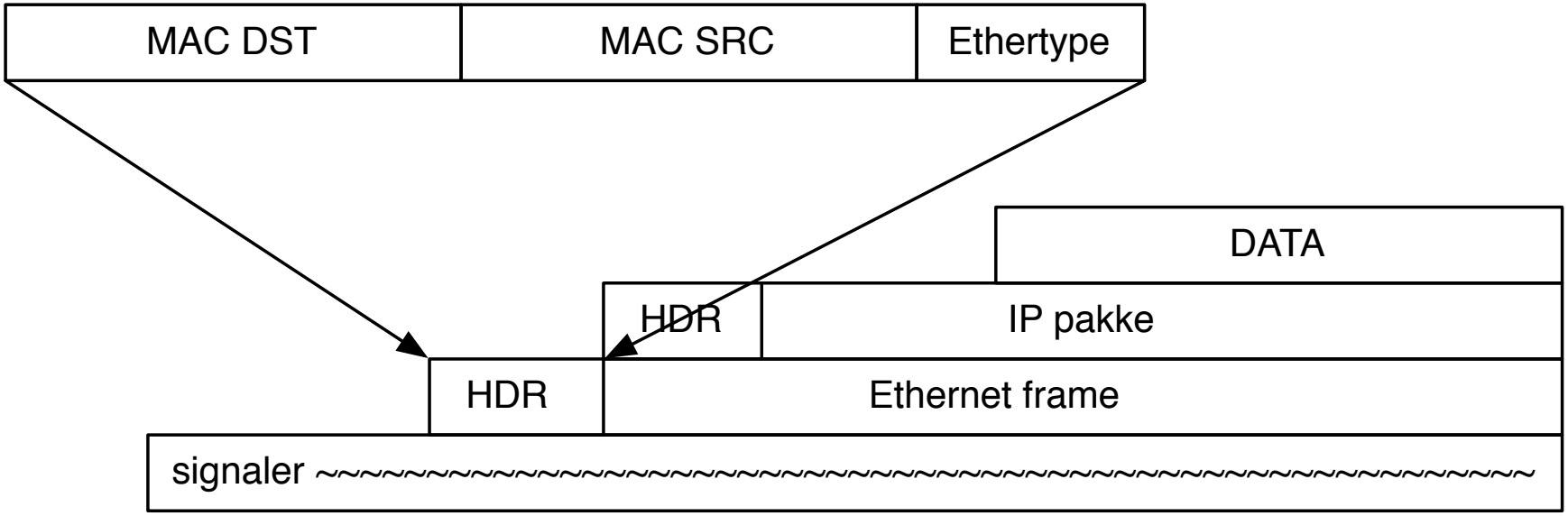
Core, Distribution og Access net



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

~~Where are the NAT gateways?~~

Pakker i en datastrøm

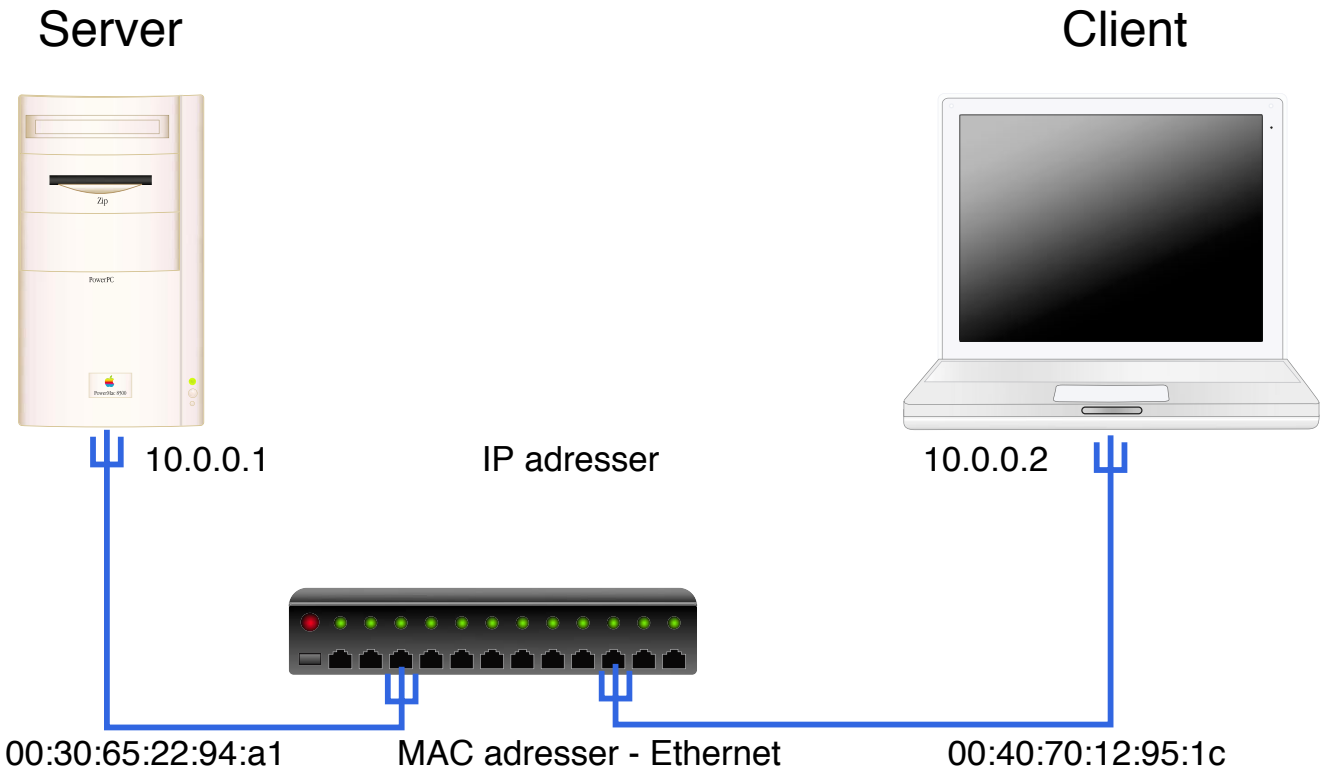


Ser vi data som en datastrøm er pakkerne blot et mønster lagt henover data

Netværksteknologien definerer start og slut på en frame

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

ARP in Ipv4



ARP request and reply

ping 10.0.0.2 from server

ARP Address Resolution Protocol request/reply:

- ARP request broadcasted on layer 2 - Who has 10.0.0.2 Tell 10.0.0.1
- ARP reply (from 10.0.0.2) 10.0.0.2 is at 00:40:70:12:95:1c

IP ICMP request/reply:

- Echo (ping) request from 10.0.0.1 to 10.0.0.2
- Echo (ping) reply from 10.0.0.2 to 10.0.0.1
- ...

ARP is performed on Ethernet before IP can be transmitted

ARP cache

```
hlk@bigfoot:hlk$ 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]
```

ARP cache kan vises med kommandoen `arp -an`

-a viser alle

-n viser kun adresserne, prøver ikke at slå navne op - typisk hurtigere

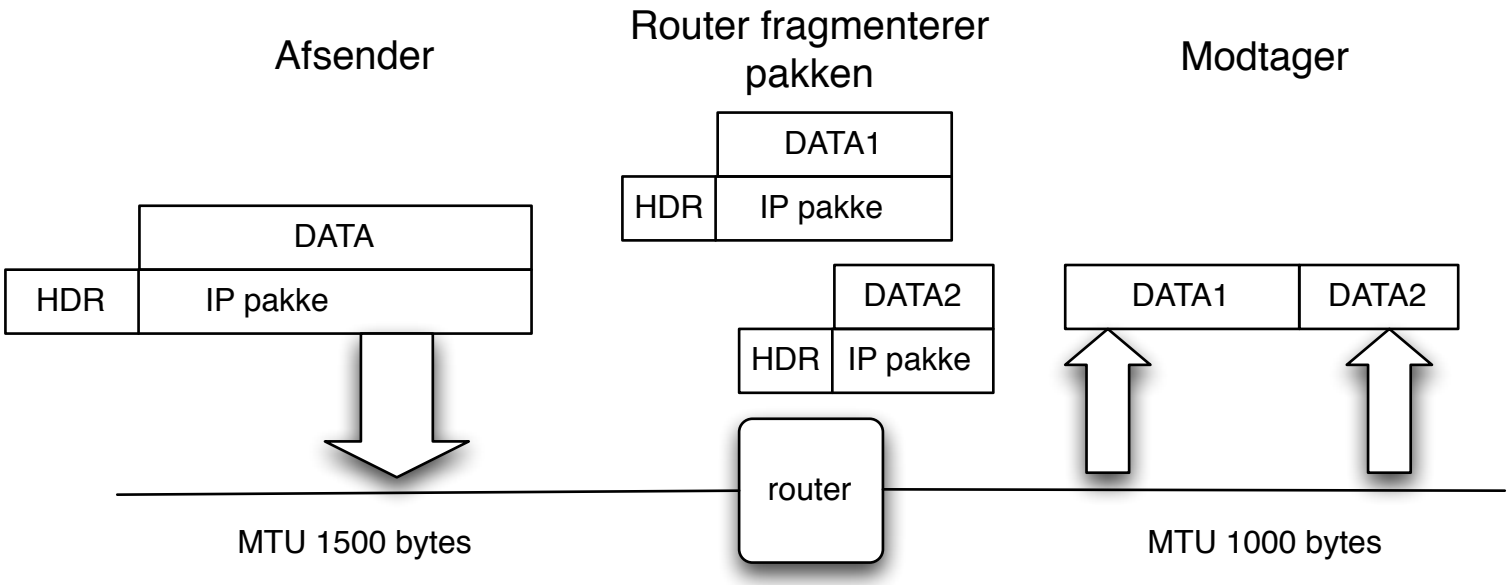
ARP cache er dynamisk og adresser fjernes automatisk efter 5-20 minutter hvis de ikke bruges mere

Læs mere med `man 4 arp`

ARP vs NDP

```
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
```

Fragmentering og PMTU



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

ICMP Internet Control Message Protocol

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!

ICMP beskedtyper

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

ICMPv6 has more

Autoconfiguration - what is the network prefix

Duplicate Address Detection - can I use this address

Neighbor Discovery - which neighbors exist

Link layer addresses - ”ARP” for IPv6

Neighbor Unreachability Detection, or NUD) - neighbors still alive

IPv6 firewalls - you MUST allow SOME ICMPv6

```
# Simple stateful network firewall rules for IPv6
# using IPv4 file for input and inspiration from
# http://www.ipv6style.jp/en/building/20040526/2.shtml
# input from
        $fwcmd6 -f flush
        $fwcmd6 add allow all from any to any via lo0
# Allow ICMPv6 destination unreach
        $fwcmd6 add pass ipv6-icmp from any to any icmptypes 1
# Allow NS/NA/toobig (don't filter it out)
        $fwcmd6 add pass ipv6-icmp from any to any icmptypes 2
# Allow timex Time exceeded
        $fwcmd6 add pass ipv6-icmp from any to any icmptypes 3
# Allow parameter problem
        $fwcmd6 add pass ipv6-icmp from any to any icmptypes 4
# IPv6 ICMP - echo request (128) and echo reply (129)
        $fwcmd6 add pass ipv6-icmp from any to any icmptypes 128,129
# IPv6 ICMP - router solicitation (133) and router advertisement (134)
        $fwcmd6 add pass ipv6-icmp from any to any icmptypes 133,134
# IPv6 ICMP - neighbour discovery solicitation (135) and advertisement (136)
        $fwcmd6 add pass ipv6-icmp from any to any icmptypes 135,136
```

IPv6 firewalls, cont. allowing services

SOLIDO NETWORKS

```
#       Allow all established connections to persist (setup required
#       for new connections).
        $fwcmd6 add allow tcp from any to any established
        $fwcmd6 add allow tcp from any to any out setup
# allow access to my webserver and ssh
#       $fwcmd6 add allow tcp from any to any 80,443  setup
        $fwcmd6 add allow tcp from any to any $ssh  setup

# allow access to X11 forwarding over ::1
        $fwcmd6 add allow tcp from any to ::1 6010  setup

#       Politely rejects AUTH requests (e.g. email and ftp)
        $fwcmd6 add reset tcp from any to any 113

#       Deny everything else ipv6
        $fwcmd6 add 65435 deny log ipv6 from any to any
```

Collect information about your network

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

Ping

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
```

Flere små forskelle

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

Unix - practical examples ifconfig and ping

SOLIDO NETWORKS

```
$ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet6 fe80::216:cbff:feac:1d9f%en0 prefixlen 64 scopeid 0x4
inet 10.0.42.15 netmask 0xffffff00 broadcast 10.0.42.255
inet6 2001:16d8:dd0f:cf0f:216:cbff:feac:1d9f prefixlen 64 autoconf
ether 00:16:cb:ac:1d:9f
media: autoselect (1000baseT <full-duplex>) status: active

$ ping6 ::1
PING6(56=40+8+8 bytes) ::1 --> ::1
16 bytes from ::1, icmp_seq=0 hlim=64 time=0.089 ms
16 bytes from ::1, icmp_seq=1 hlim=64 time=0.155 ms

$ traceroute6 2001:16d8:dd0f:cf0f::1
traceroute6 to 2001:16d8:dd0f:cf0f::1 (2001:16d8:dd0f:cf0f::1)
from 2001:16d8:dd0f:cf0f:216:cbff:feac:1d9f, 64 hops max, 12 byte packets
 1  2001:16d8:dd0f:cf0f::1  0.399 ms  0.371 ms  0.294 ms
```

ping6 global unicast address

```
root# ping6 2001:1448:81:beef:20a:95ff:fef5:34df
PING6(56=40+8+8 bytes) 2001:1448:81:beef::1 --> 2001:1448:81:beef:20a:95ff:fef5:34df
16 bytes from 2001:1448:81:beef:20a:95ff:fef5:34df, icmp_seq=0 hlim=64 time=10.639 ms
16 bytes from 2001:1448:81:beef:20a:95ff:fef5:34df, icmp_seq=1 hlim=64 time=1.615 ms
16 bytes from 2001:1448:81:beef:20a:95ff:fef5:34df, icmp_seq=2 hlim=64 time=2.074 ms
^C
--- 2001:1448:81:beef:20a:95ff:fef5:34df ping6 statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1.615/4.776/10.639 ms
```

ping6 link-local address

```
hlk@bigfoot:hlk$ ping6 -I en1 fe80::20d:93ff:fe4d:55fe
PING6(56=40+8+8 bytes) fe80::223:6cff:fe9a:f52c%en1 --> fe80::20d:93ff:fe4d:55fe
16 bytes from fe80::20d:93ff:fe4d:55fe%en1, icmp_seq=0 hlim=64 time=1.557 ms
16 bytes from fe80::20d:93ff:fe4d:55fe%en1, icmp_seq=1 hlim=64 time=1.725 ms
^C
--- fe80::20d:93ff:fe4d:55fe ping6 statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 1.557/1.641/1.725/0.084 ms
```

Note: -I en1 specifies that this interface is being used.

ping6 til specielle adresser

```
root# ping6 -I en1 ff02::1

PING6(56=40+8+8 bytes) fe80::230:65ff:fe17:94d1 --> ff02::1
16 bytes from fe80::230:65ff:fe17:94d1, icmp_seq=0 hlim=64 time=0.483 ms
16 bytes from fe80::20a:95ff:fef5:34df, icmp_seq=0 hlim=64 time=982.932 ms
16 bytes from fe80::230:65ff:fe17:94d1, icmp_seq=1 hlim=64 time=0.582 ms
16 bytes from fe80::20a:95ff:fef5:34df, icmp_seq=1 hlim=64 time=9.6 ms
16 bytes from fe80::230:65ff:fe17:94d1, icmp_seq=2 hlim=64 time=0.489 ms
16 bytes from fe80::20a:95ff:fef5:34df, icmp_seq=2 hlim=64 time=7.636 ms
^C
--- ff02::1 ping6 statistics ---
4 packets transmitted, 4 packets received, +4 duplicates, 0% packet loss
round-trip min/avg/max = 0.483/126.236/982.932 ms

ff02::1 multicast address of all-hosts on the local link
ff02::2 multicast address of all-routers on the local link
```

Hello neighbors

```
$ 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
```

traceroute

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

traceroute - med UDP

```
#  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
```

Værdien af traceroute

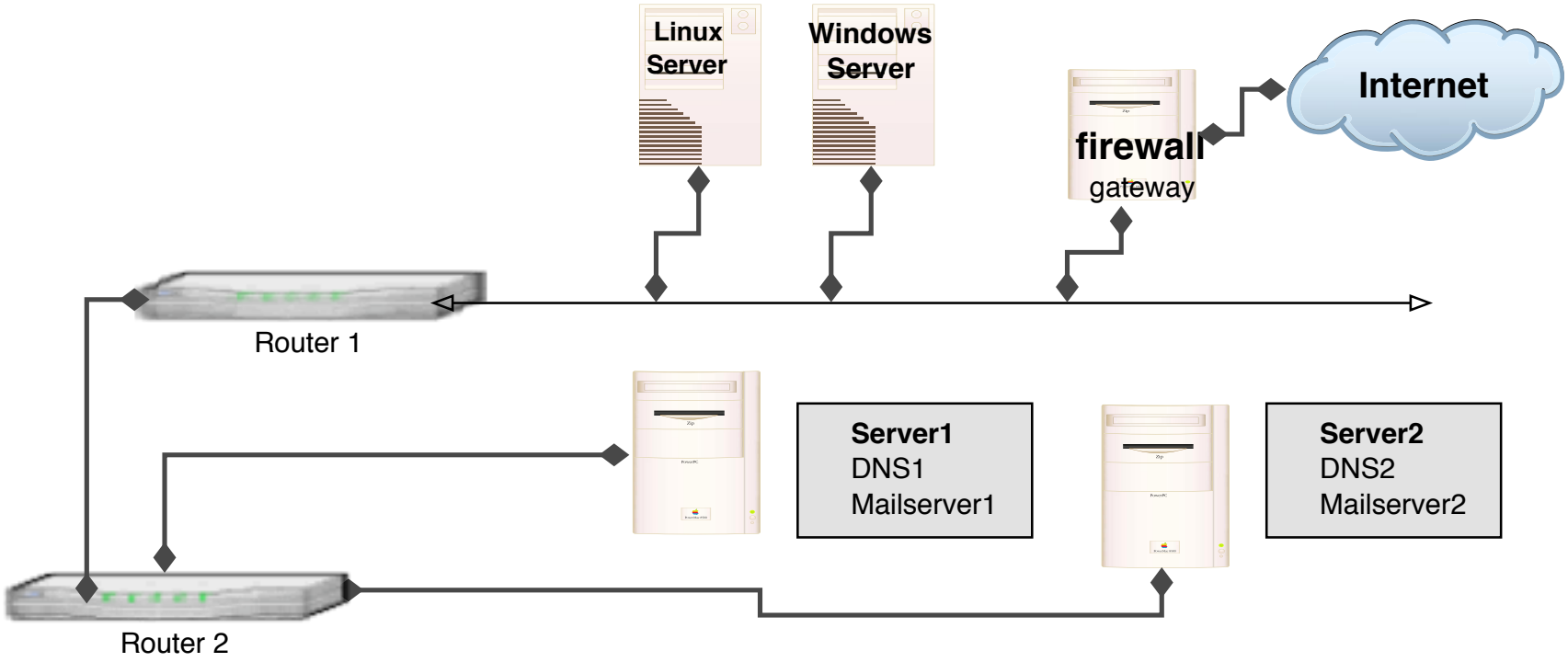
Diagnosticering af netværksproblemer - formålet med traceroute

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!

Network mapping



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

Kommandolinien på Unix

Shells kommandofortolkere:

- sh - Bourne Shell
- bash - Bourne Again Shell
- ksh - Korn shell, lavet af David Korn
- csh - C shell, syntaks der minder om C sproget
- flere andre, zsh, tcsh

Svarer til command.com og cmd.exe på Windows

Kan bruges som komplette programmeringssprog

Kommandoprompten

```
[hlk@fischer hlk]$ id
uid=6000(hlk) gid=20(staff) groups=20(staff),
0(wheel), 80(admin), 160(cvs)
[hlk@fischer hlk]$

[root@fischer hlk]# id
uid=0(root) gid=0(wheel) groups=0(wheel), 1(daemon),
2(kmem), 3(sys), 4(tty), 5(operator), 20(staff),
31(guest), 80(admin)
[root@fischer hlk]#
```

typisk viser et dollartegn at man er logget ind som almindelig bruge

mens en havelåge at man er root - superbruger

Kommandoliniens opbygning

```
echo [-n] [string ...]
```

Kommandoerne der skrives på kommandolinien skrives sådan:

- Starter altid med kommandoen, man kan ikke skrive `henrik echo`
- Options skrives typisk med bindestreg foran, eksempelvis `-n`
- Flere options kan sættes sammen, `tar -cvf` eller `tar cvf`
- I manualsystemet kan man se valgfrie options i firkantede klammer `[]`
- Argumenterne til kommandoen skrives typisk til sidst (eller der bruges redirection)

Vi laver nu øvelsen

Putty installation - Secure Shell login

som er øvelse 1 fra øvelseshæftet.

SOLIDO NETWORKS

Vi laver nu øvelsen

Login på Unix systemerne

som er øvelse 3 fra øvelseshæftet.

TCP/IP basiskonfiguration

```
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:

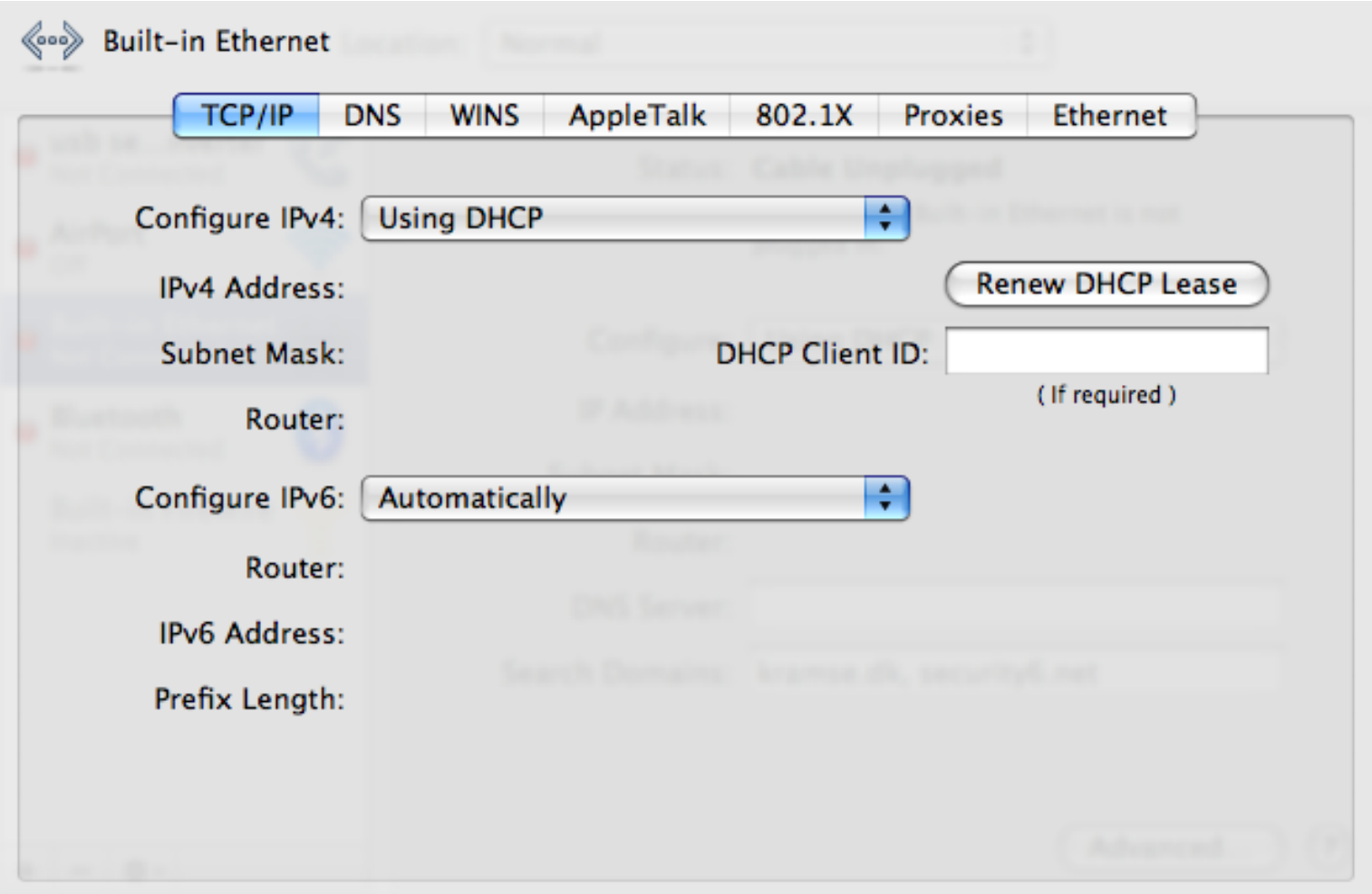
`ifconfig`, `route` og `netstat`

- ofte pakket ind i konfigurationsmenuer m.v.

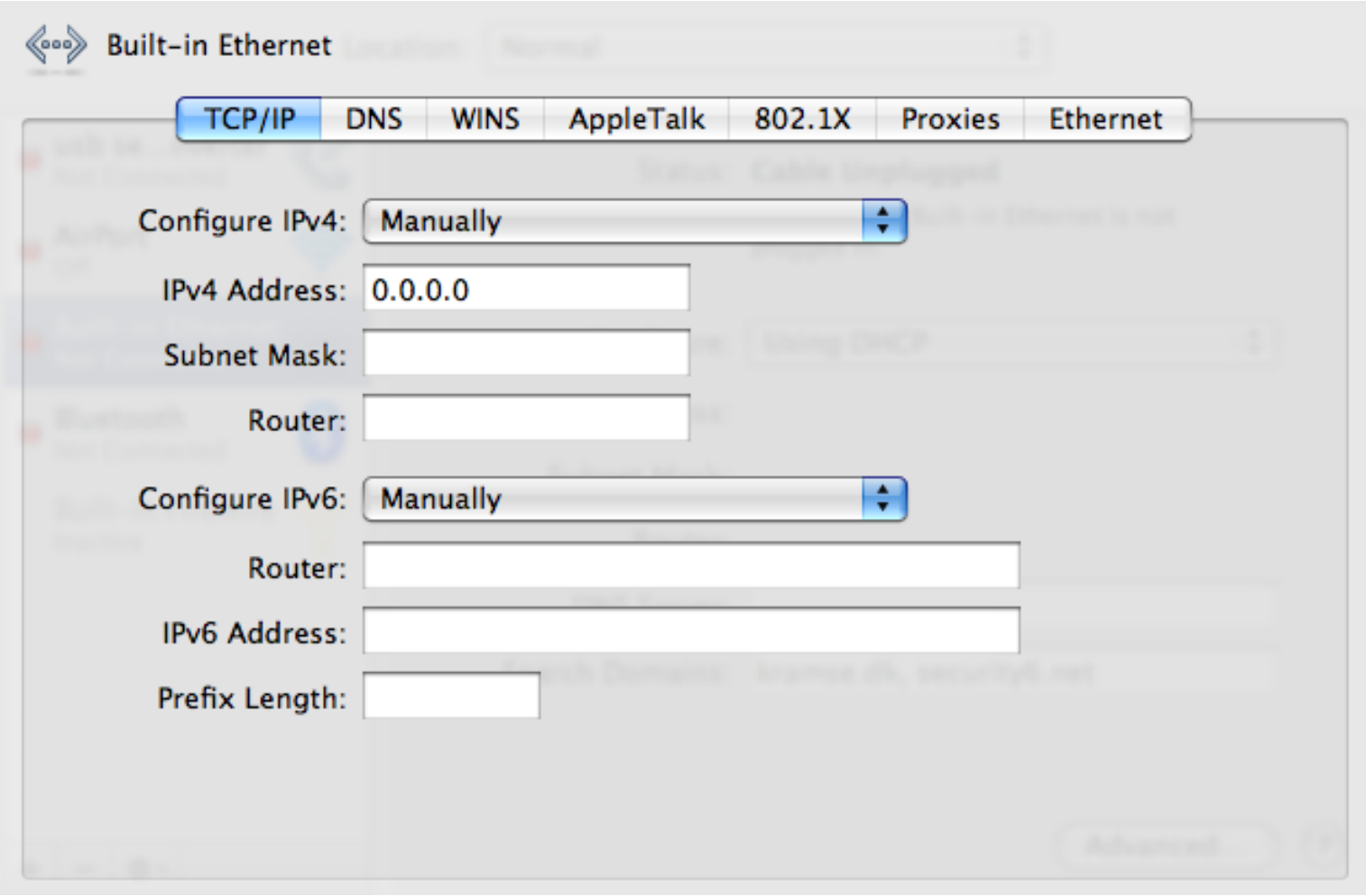
fejlsøgning foregår typisk med `ping` og `traceroute`

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

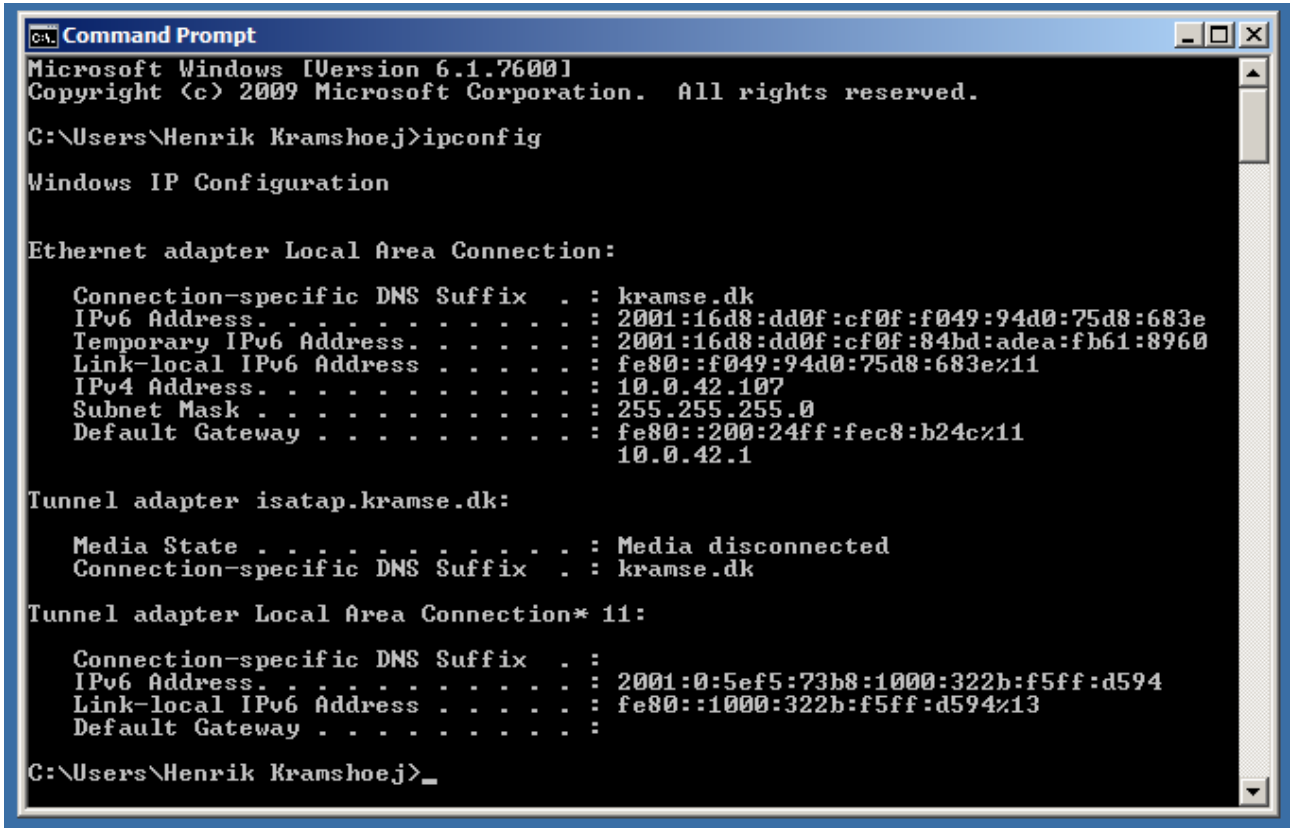
GUI værktøjer - autoconfiguration



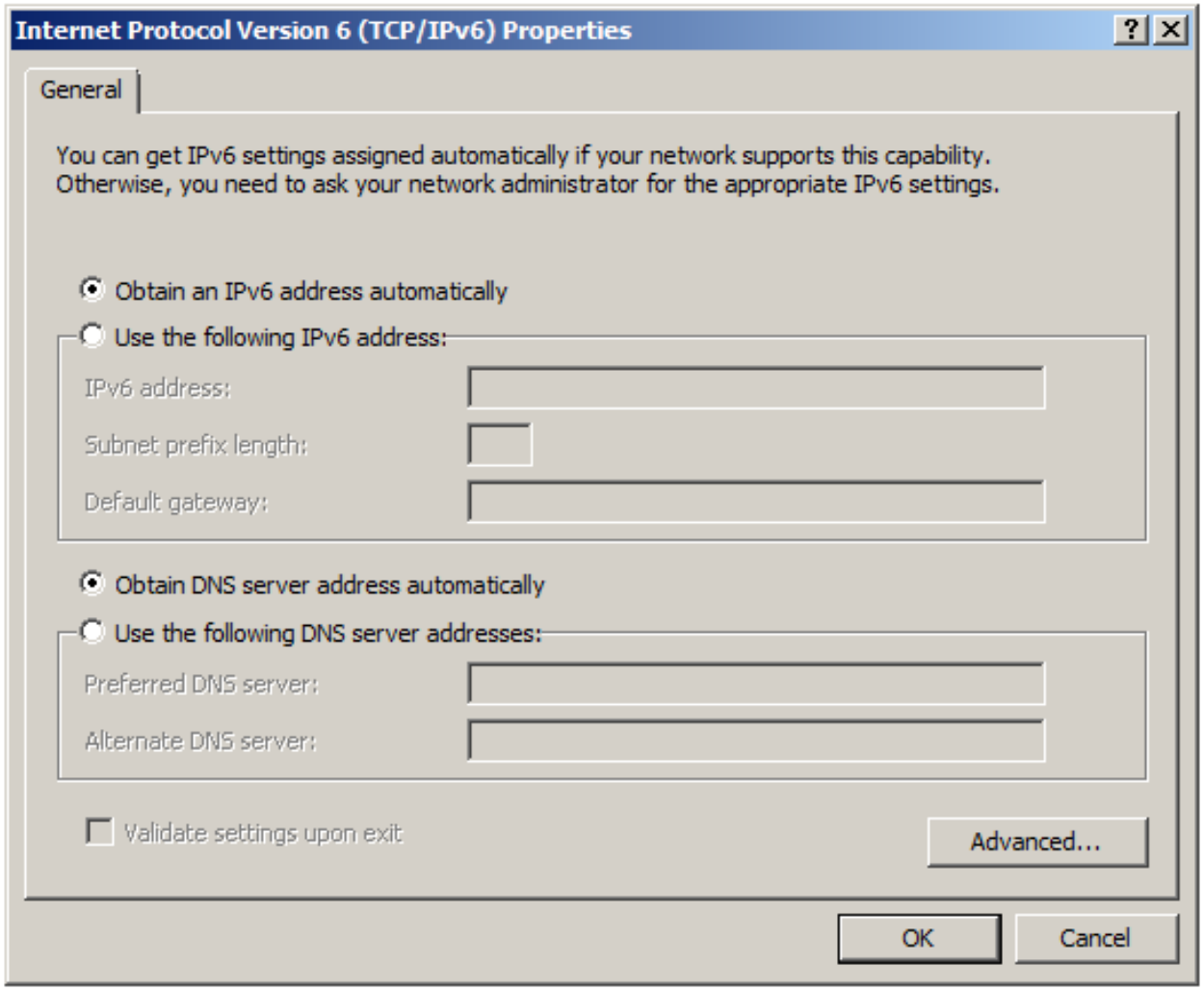
GUI værktøjer - manuel konfiguration



Windows - ipconfig



Windows - control panel



ifconfig output

```
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

Vigtigste protokoller

ARP Address Resolution Protocol

IP og ICMP Internet Control Message Protocol

UDP User Datagram Protocol

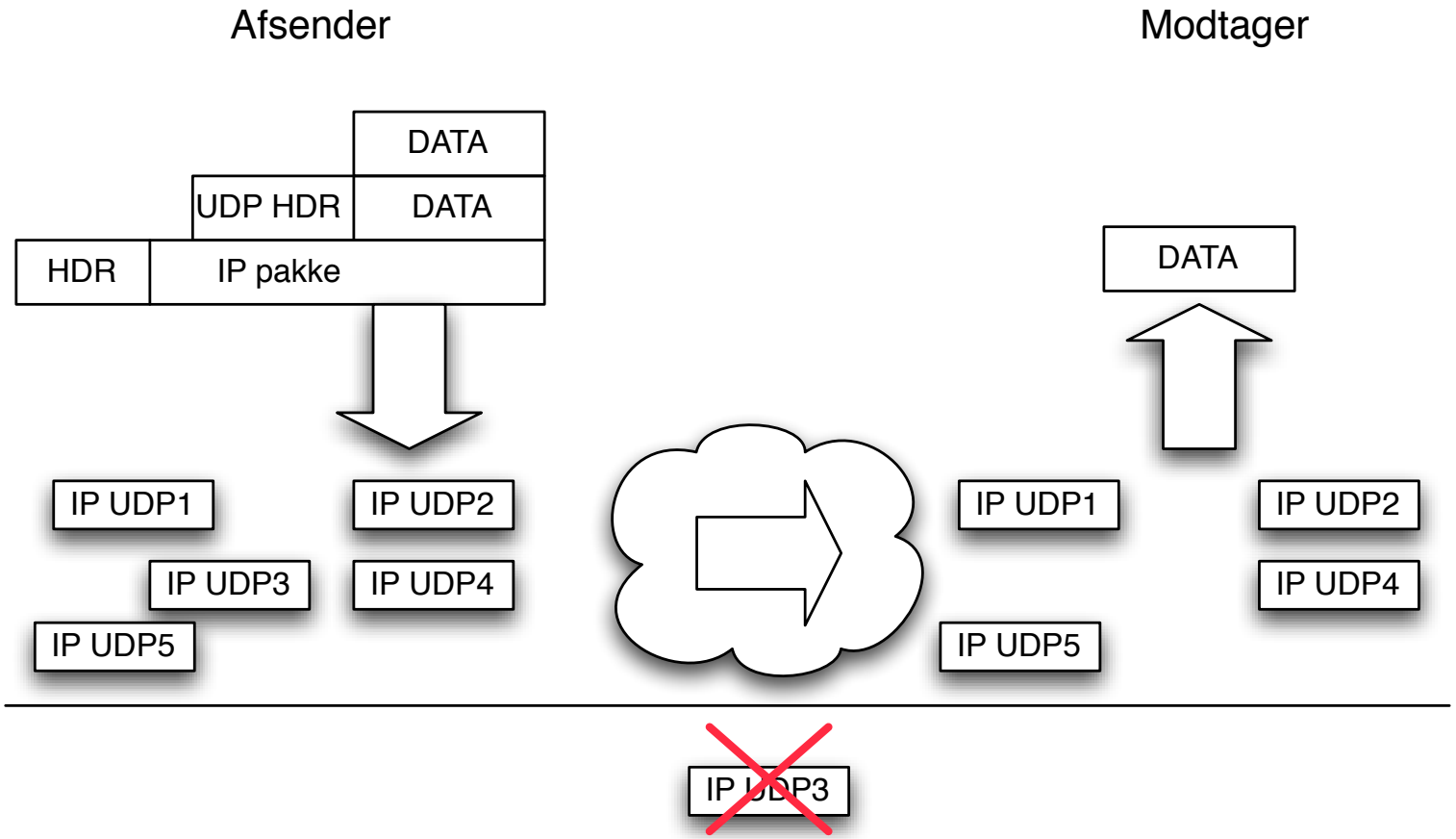
TCP Transmission Control Protocol

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

Ovenstående er omtrent minimumskrav for at komme på internet

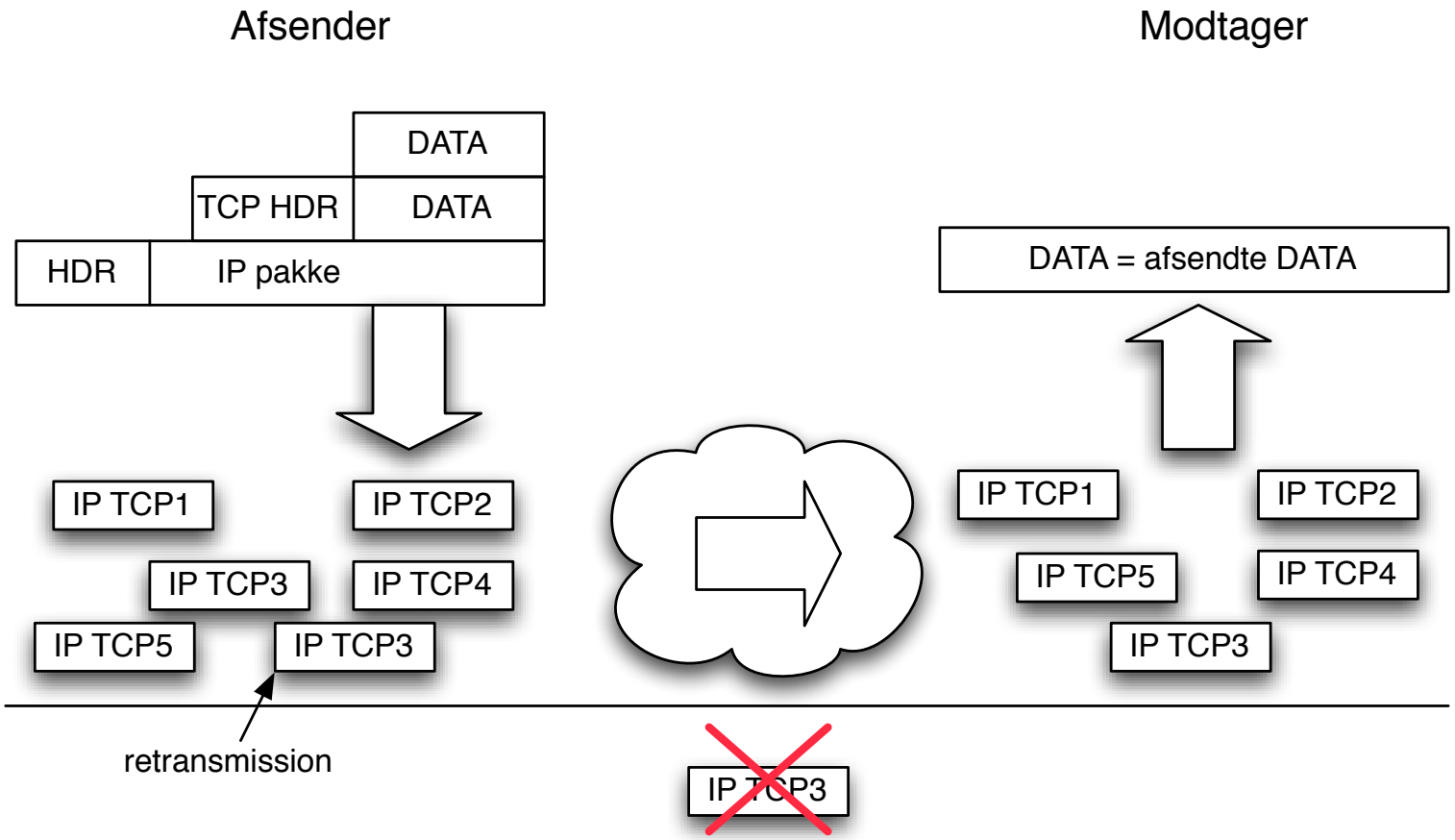
UDP User Datagram Protocol



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

Kan benyttes til multicast/broadcast - flere modtagere

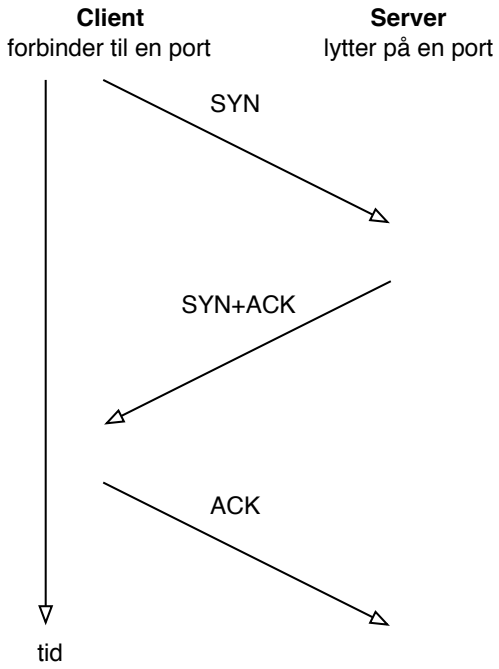
TCP Transmission Control Protocol



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**

Well-known port numbers

IANA vedligeholder en liste over magiske konstanter i IP

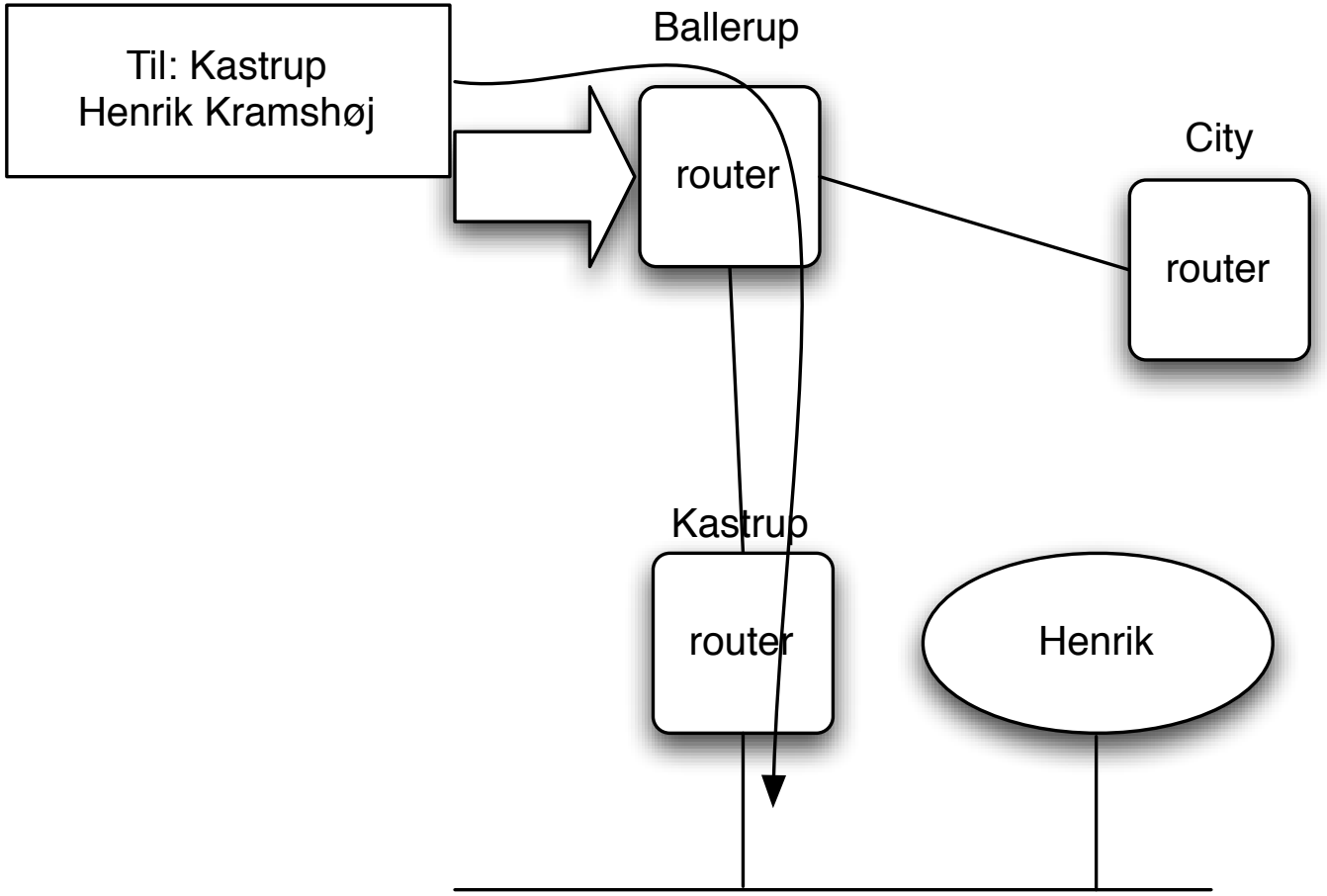
De har lister med hvilke protokoller har hvilke protokol ID m.v.

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 80 HTTP Hyper Text Transfer Protocol over TLS/SSL
- Port 443 HTTP over TLS/SSL

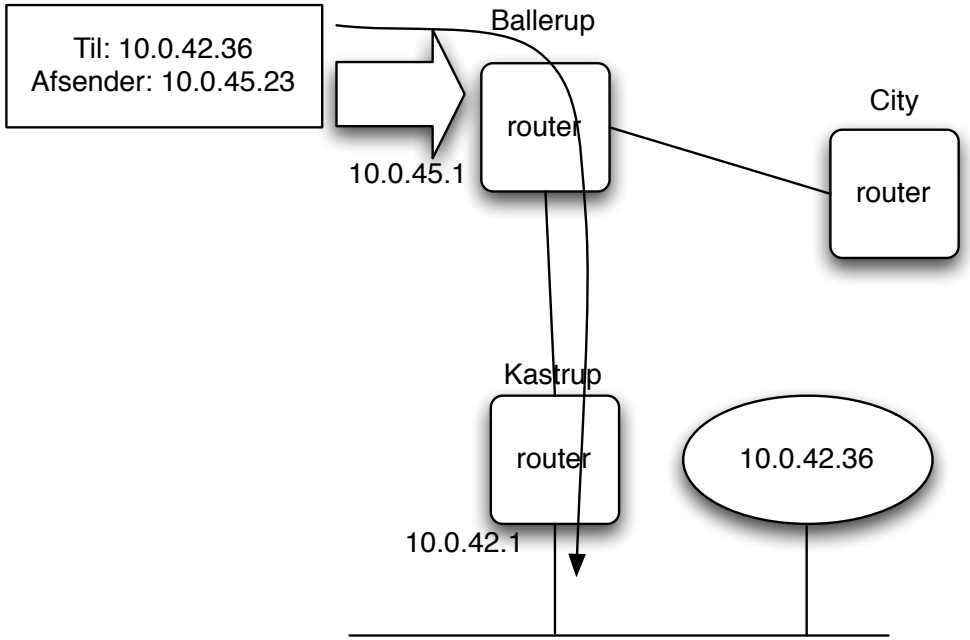
Se flere på `http://www.iana.org`

Hierarkisk routing



Hvordan kommer pakkerne frem til modtageren

IP default gateway



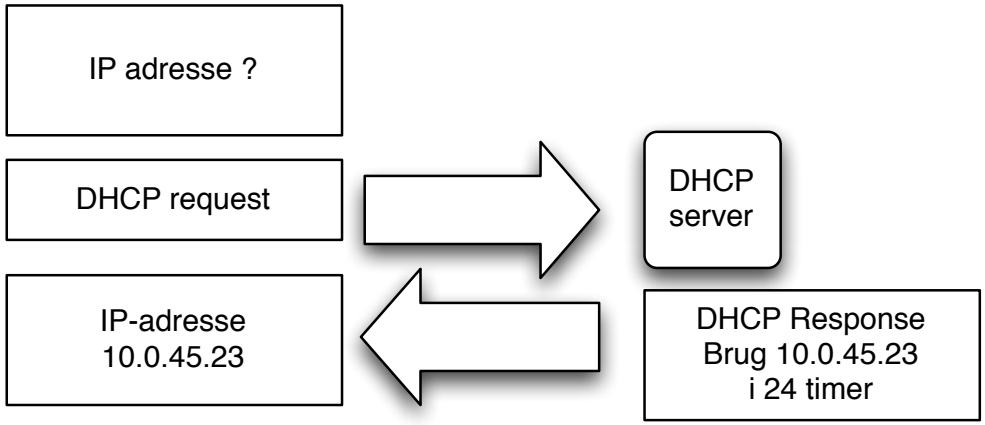
IP routing er nemt

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

DHCP Dynamic Host Configuration Protocol



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

IPv6 router advertisement daemon

```
/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

IPv6 autoconfiguration

SOLIDO NETWORKS

```
Modified EUI-64 format-based interface identifiers
```

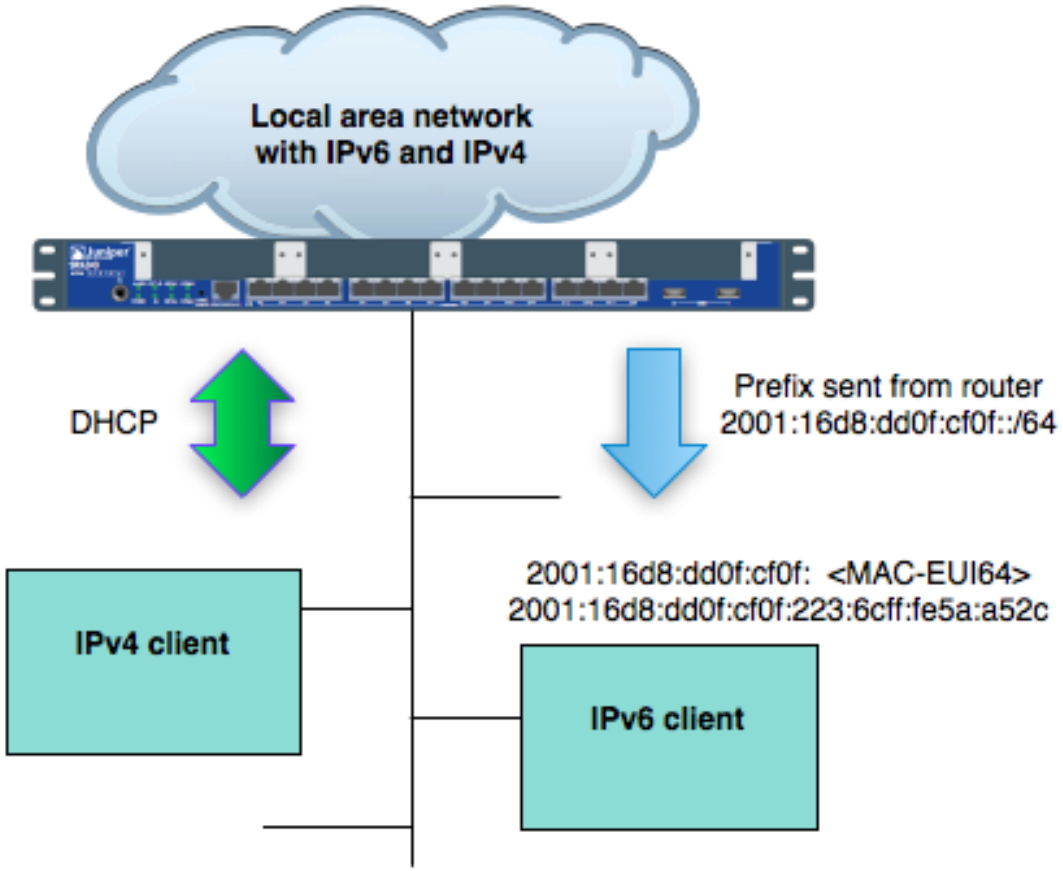
```
ifconfig en1
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 00:23:6c:9a:f5:2c
                00-23-6c-ff-fe-9a-f5-2c  48-bit MAC stretched to become EUI-64
                02-23-6c-ff-fe-9a-f5-2c  inverting the "u" bit (universal/local bit)
                fe80::  +  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

Router advertisement daemon



Routing

SOLIDO NETWORKS

routing table - tabel over netværkskort og tilhørende adresser

default gateway - den adresse hvortil man sender non-local pakker
kaldes 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!

Routing forståelse

```
$ 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.

SOLIDO NETWORKS

Vi laver nu øvelsen

Netværksinformation: netstat

som er øvelse **5** fra øvelseshæftet.

SOLIDO NETWORKS

Vi laver nu øvelsen

ping og traceroute

som er øvelse 6 fra øvelseshæftet.

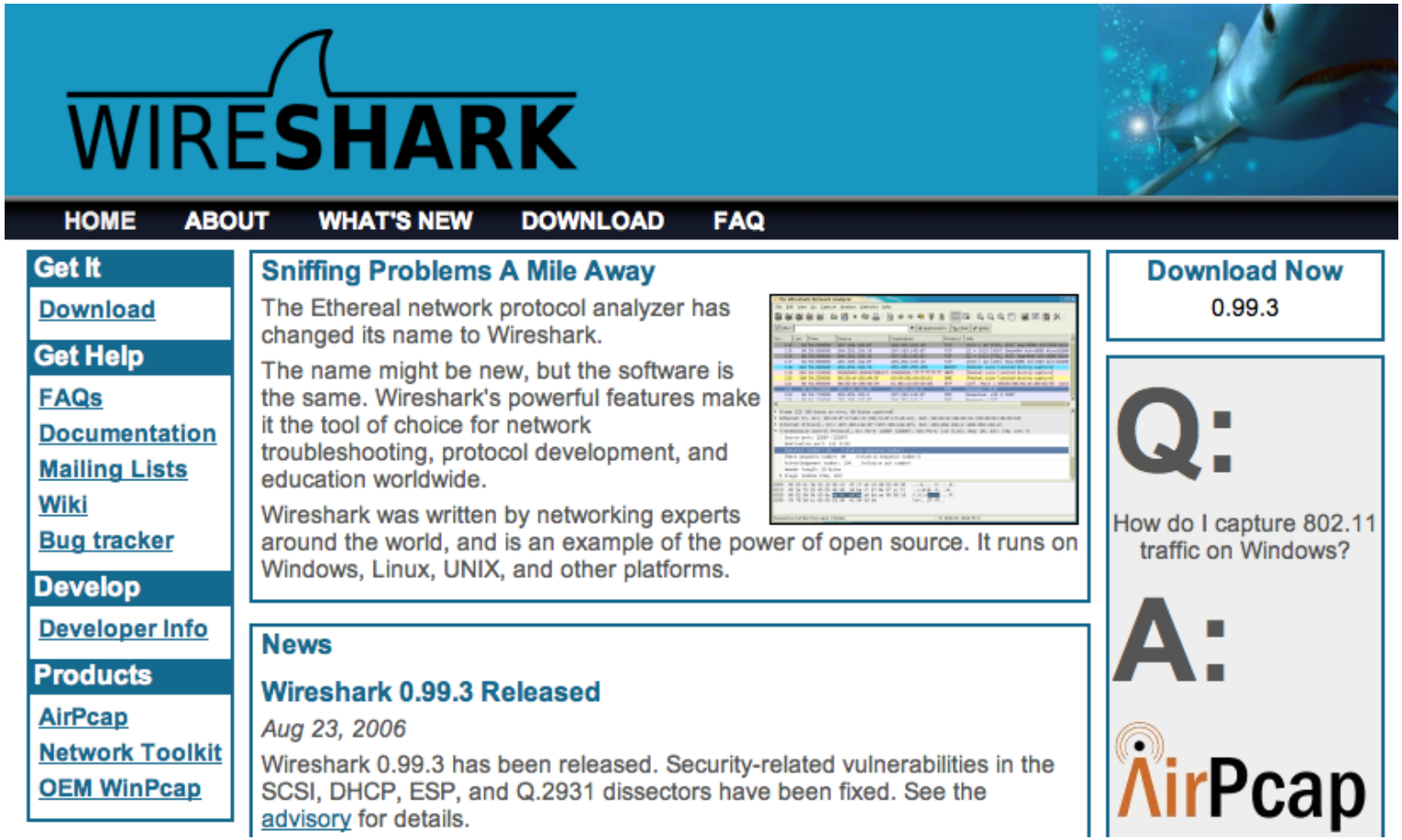
SOLIDO NETWORKS

Vi laver nu øvelsen

ping6 og traceroute6

som er øvelse 7 fra øvelseshæftet.

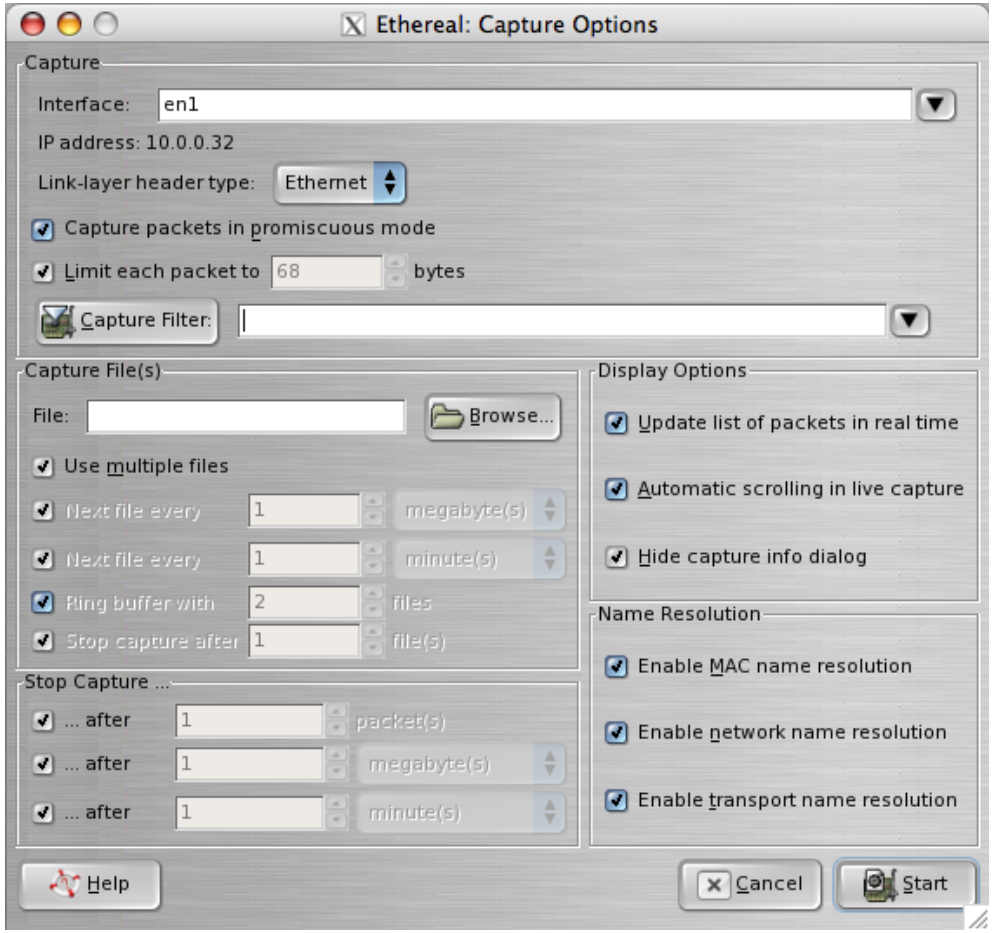
Wireshark - grafisk pakkesniffer



`http://www.wireshark.org`

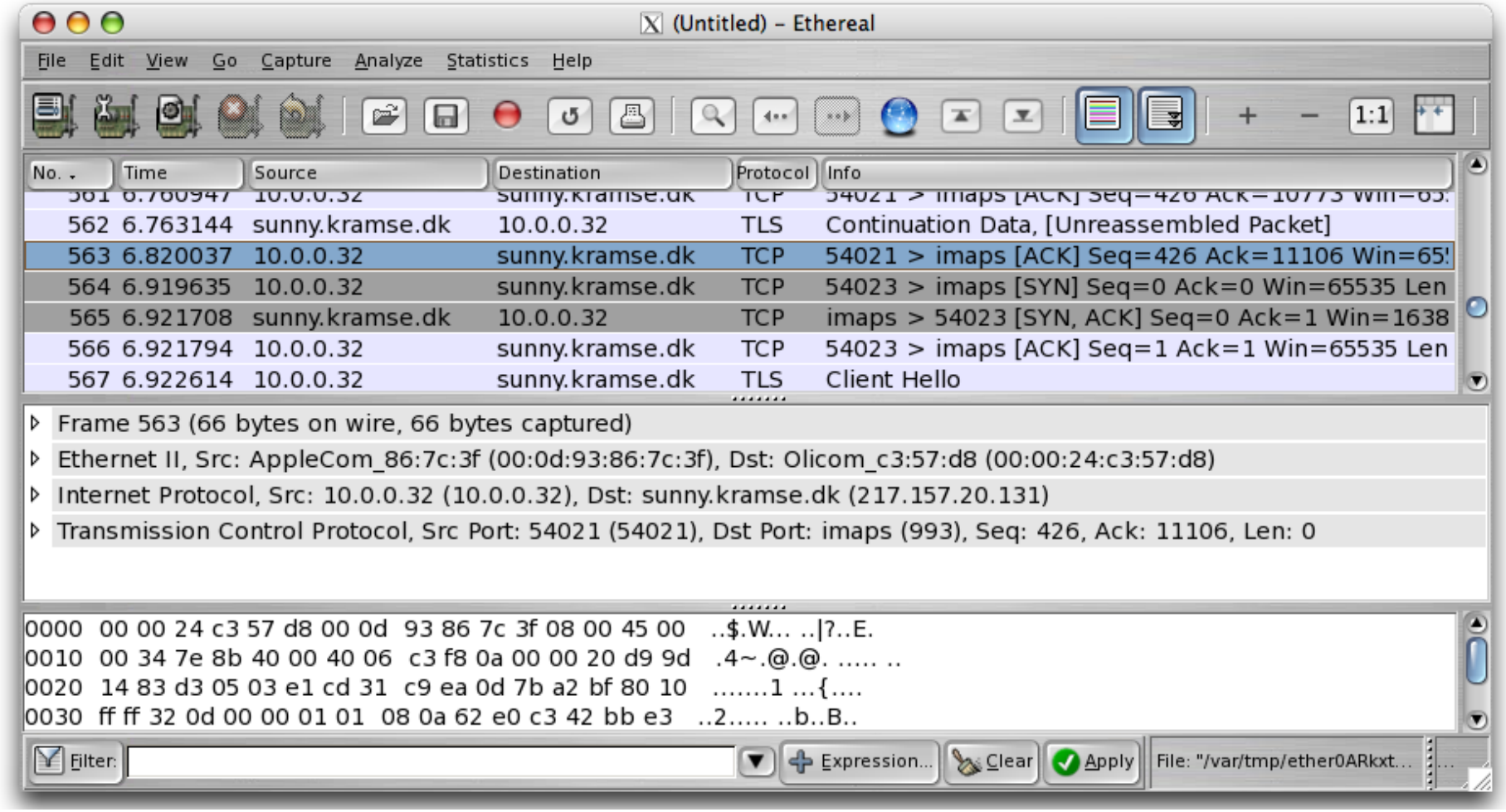
både til Windows og Unix, tidligere kendt som Ethereal

Brug af Wireshark



Man starter med Capture - Options

Brug af Wireshark



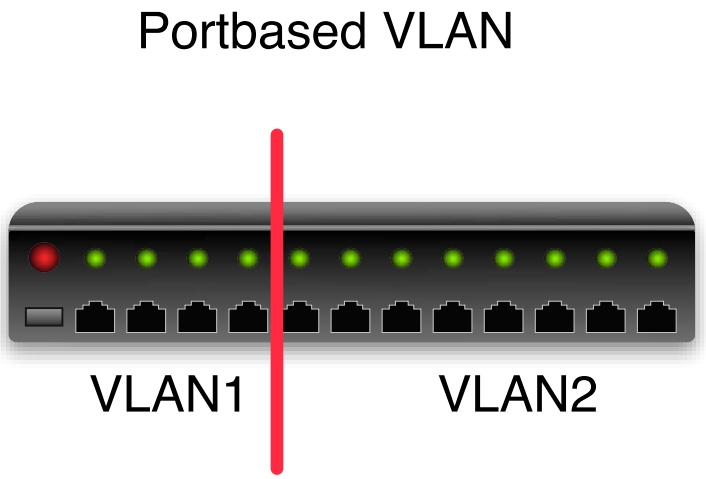
Læg mærke til filtermulighederne

Vi laver nu øvelsen

Wireshark netværksniffer

som er øvelse 8 fra øvelseshæftet.

VLAN Virtual LAN



Nogle switche tillader at man opdeler portene

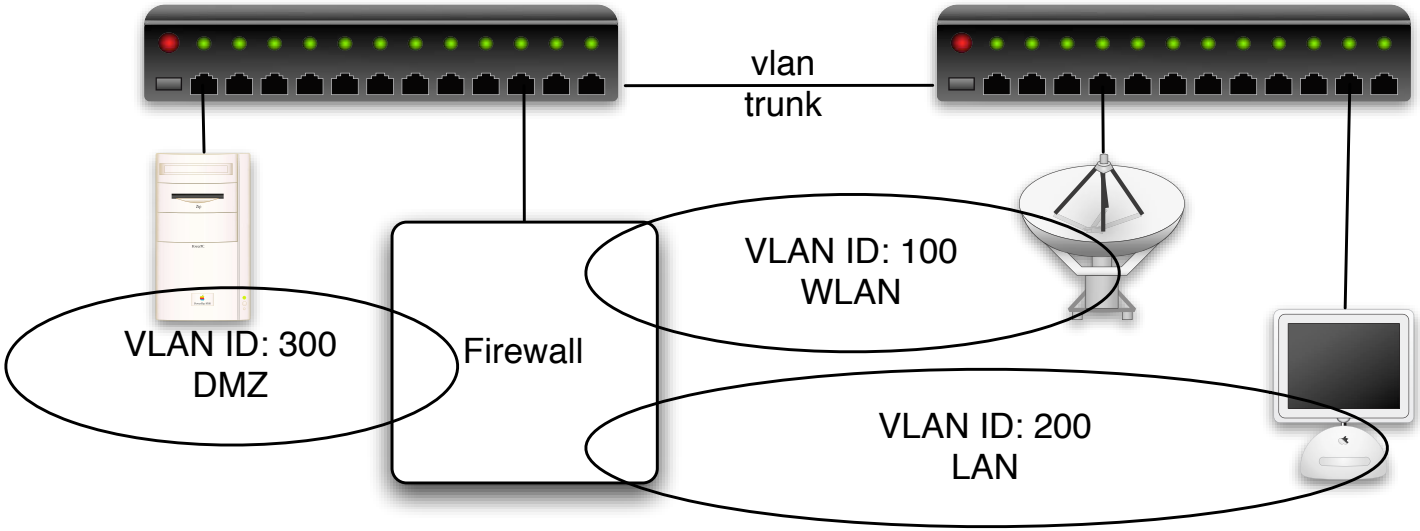
Denne opdeling kaldes VLAN og portbaseret er det mest simple

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

IEEE 802.1q



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

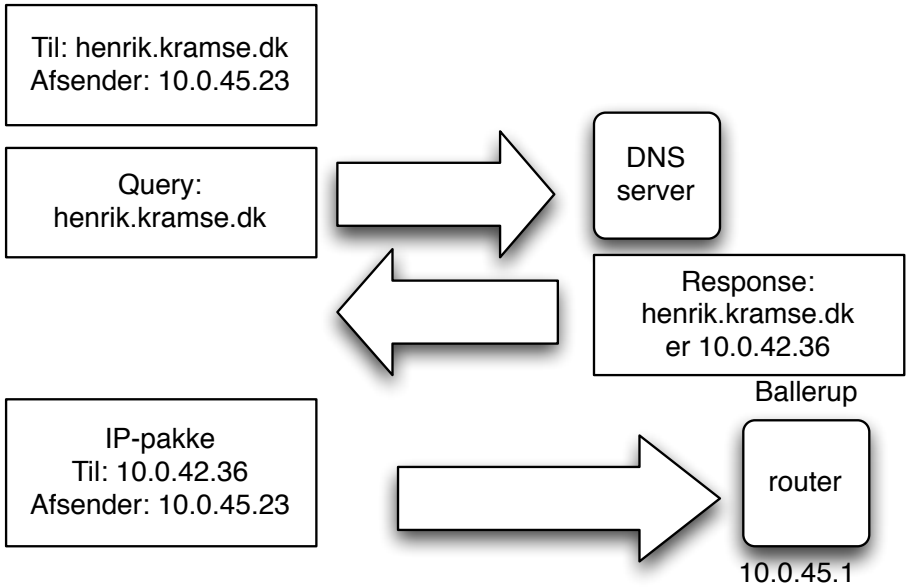
Der findes administrationsværktøjer der letter dette arbejde: OpenNAC FreeNAC, Cisco VMPS

Vi laver nu øvelsen

VLAN 802.1q

som er øvelse 9 fra øvelseshæftet.

Domain Name System



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

Mere end navneopslag

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.
```

Basal DNS opsætning på klienter

`/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 security6.net
nameserver 212.242.40.3
nameserver 212.242.40.51
```

BIND DNS server

Berkeley Internet Name Daemon server

Mange bruger BIND fra Internet Systems Consortium - altså Open Source

konfigureres gennem `named.conf`

det anbefales at bruge BIND version 9

- 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`

BIND konfiguration - et udgangspunkt

```
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;   };
...
}
```

Other DNS software

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

NSD `http://www.nlnetlabs.nl/projects/nsd/`

OpenDNSSEC `http://www.opendnssec.org/`

SOLIDO NETWORKS

Vi laver nu øvelsen

DNS og navneopslag

som er øvelse **10** fra øvelseshæftet.

Vi laver nu øvelsen

DNS og navneopslag - IPv6

som er øvelse 11 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

IP netværkstuning

IP har eksisteret mange år

Vi har udskiftet langsommme 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

Anbefalet netværkstuning - hvad skal tunes

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

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 thrashe

Måling af througput

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, blandt andet Iperf

Målinger med Nping fra Nmap

```
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
```

Målinger med Iperf

```
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 fluffy`

Antal pakker per sekund

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

SOLIDO NETWORKS

Vi laver nu øvelsen

Performance tool - iperf

som er øvelse **15** fra øvelseshæftet.

Opsummering

Husk følgende:

- Unix og Linux er blot eksempler - navneservice eller HTTP server kører fint på Windows
- DNS er grundlaget for Internet
- Sikkerheden på internet er generelt dårlig, brug SSL!
- Procedurerne og vedligeholdelse er essentiel for alle operativsystemer!
- Man skal hærde operativsystemer før man sætter dem på Internet
- Husk: IT-sikkerhed er ikke kun netværkssikkerhed!
- God sikkerhed kommer fra langsigtede intiativer

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

Spørgsmål?

SOLIDO NETWORKS

Henrik Lund Kramshøj
hlk@solido.net

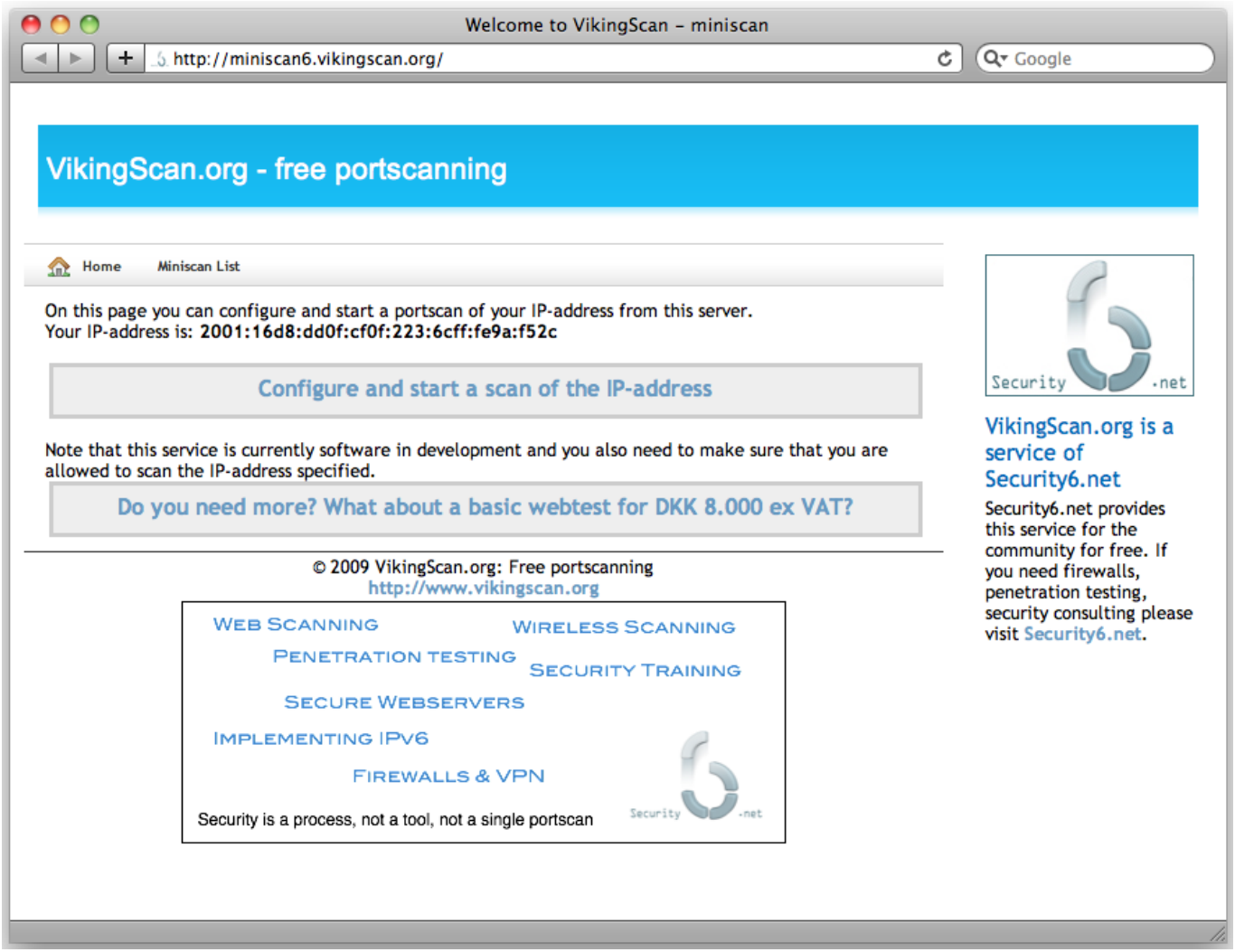
http://www.solidonetworks.com

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

Danish resources - get involved

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

VikingScan.org - free portscanning



PROSA CTF

PROSA afholdt fredag 17. september - til lørdag 18. september Capture the Flag

Distribueret CTF med 6 hold og arrangørerne i Aalborg

Sjovt og lærerigt - gentages helt sikkert

Kilde: `http://prosa-ctf.the-playground.dk/`

Referencer: netværksbøger

- O'Reilly Network Warrior - god allround bog, men også Cisco centrisk
- Stevens, Comer klassiske bøger om TCP/IP
- TCP/IP bogen på dansk måske
- O'Reilly IPv6 Network Administration
- KAME bøgerne om IPv6 protokollerne, meget detaljerede
- O'Reilly cookbooks: Cisco, BIND og Apache HTTPD m.fl.
- Cisco Press og website
- Juniper website
- Firewall bøger Cheswick
- Der findes mange gode bøger om netværk

Bøger om IPv6

IPv6 Network Administration af David Malone og Niall Richard Murphy - god til real-life admins, typisk O'Reilly bog

IPv6 Essentials af Silvia Hagen, O'Reilly 2nd edition (May 17, 2006) god reference om emnet

IPv6 Core Protocols Implementation af Qing Li, Tatuya Jinmei og Keiichi Shima

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

- flere andre