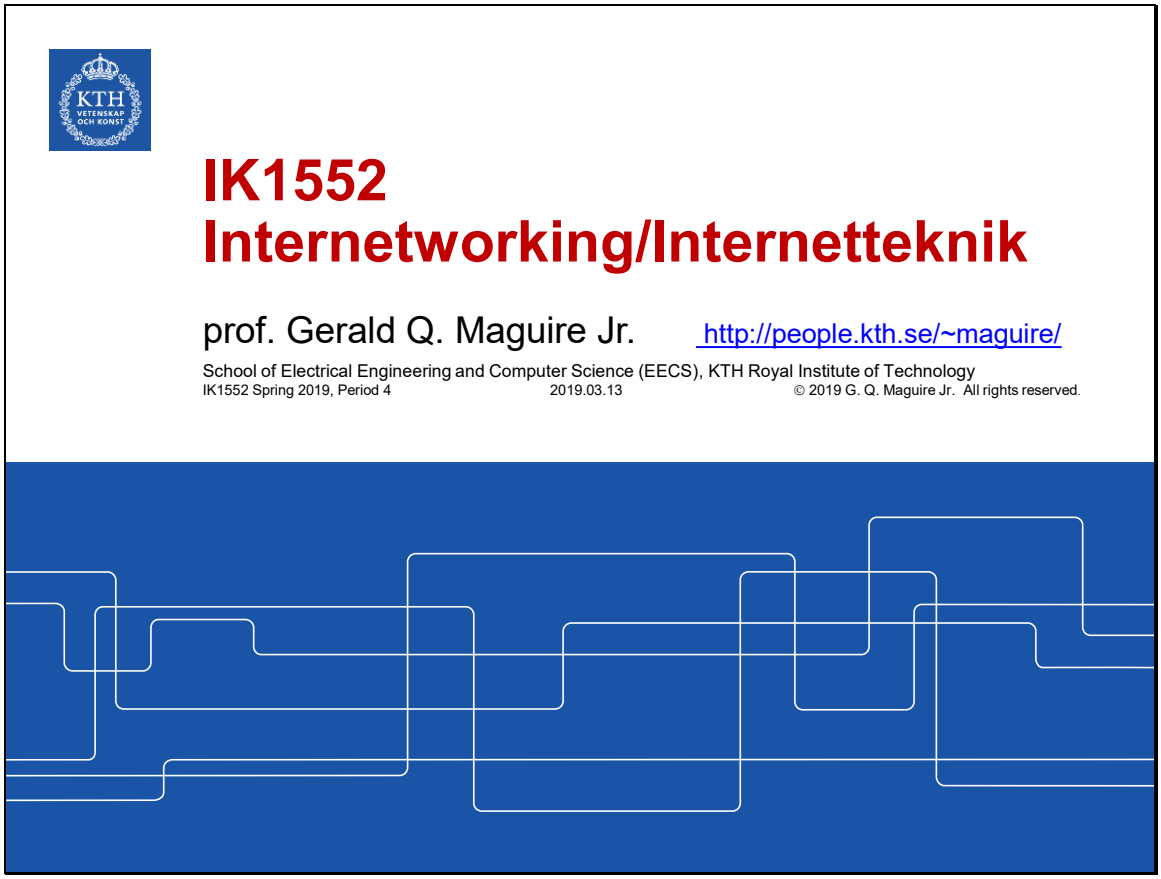
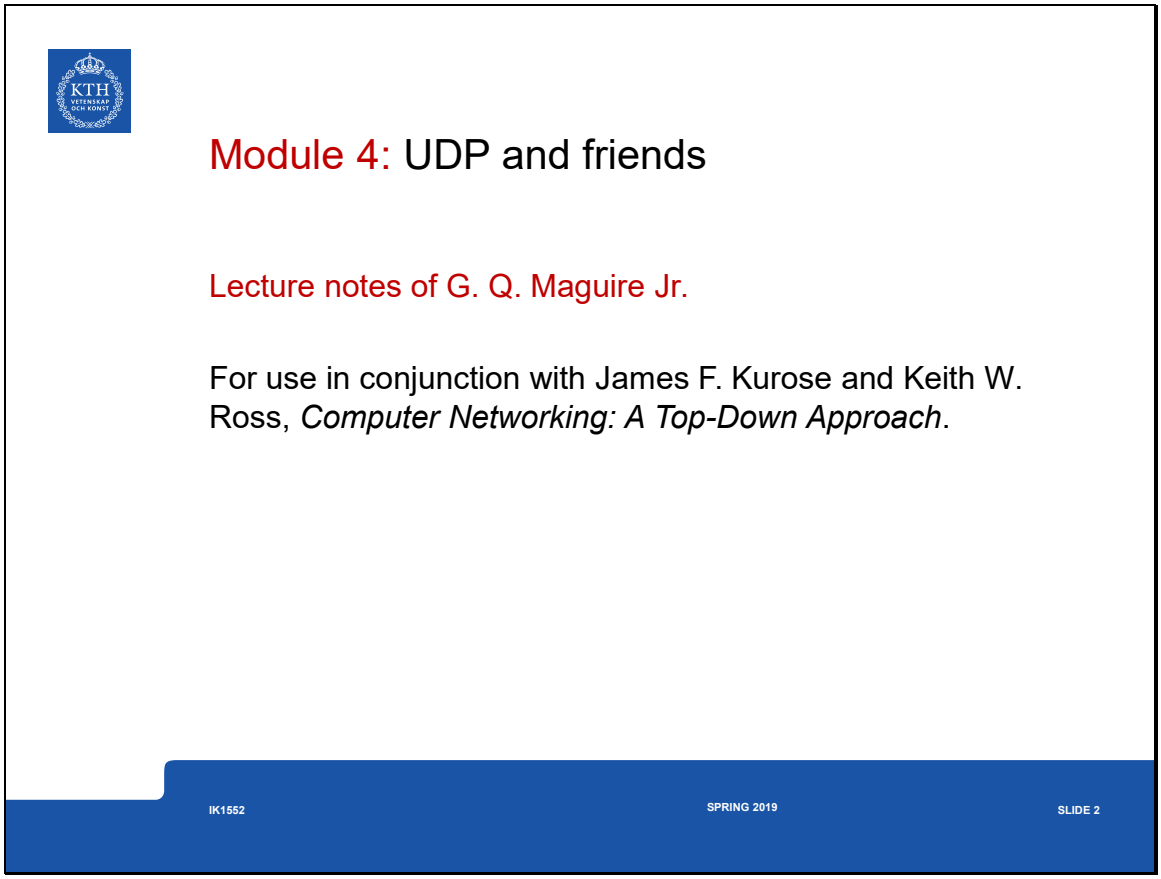
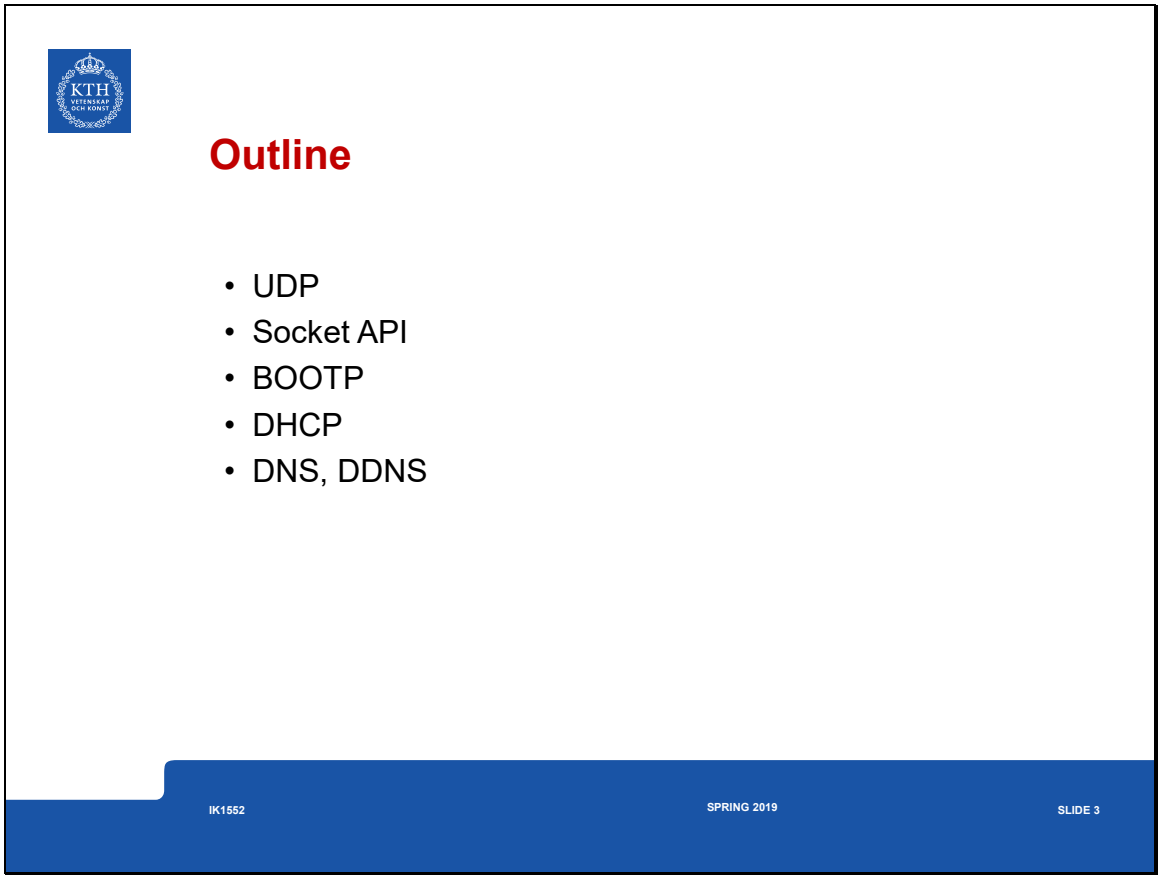
Slide 1


IK1552
Internetworking/Internetteknik
prof. Gerald Q. Maguire Jr.
http://people.kth.se/~maguire/
School of Electrical Engineering and Computer Science (EECS), KTH Royal Institute of Technology
IK1552 Spring 2019, Period 4
2019.03.13
© 2019 G. Q. Maguire Jr. All rights reserved.

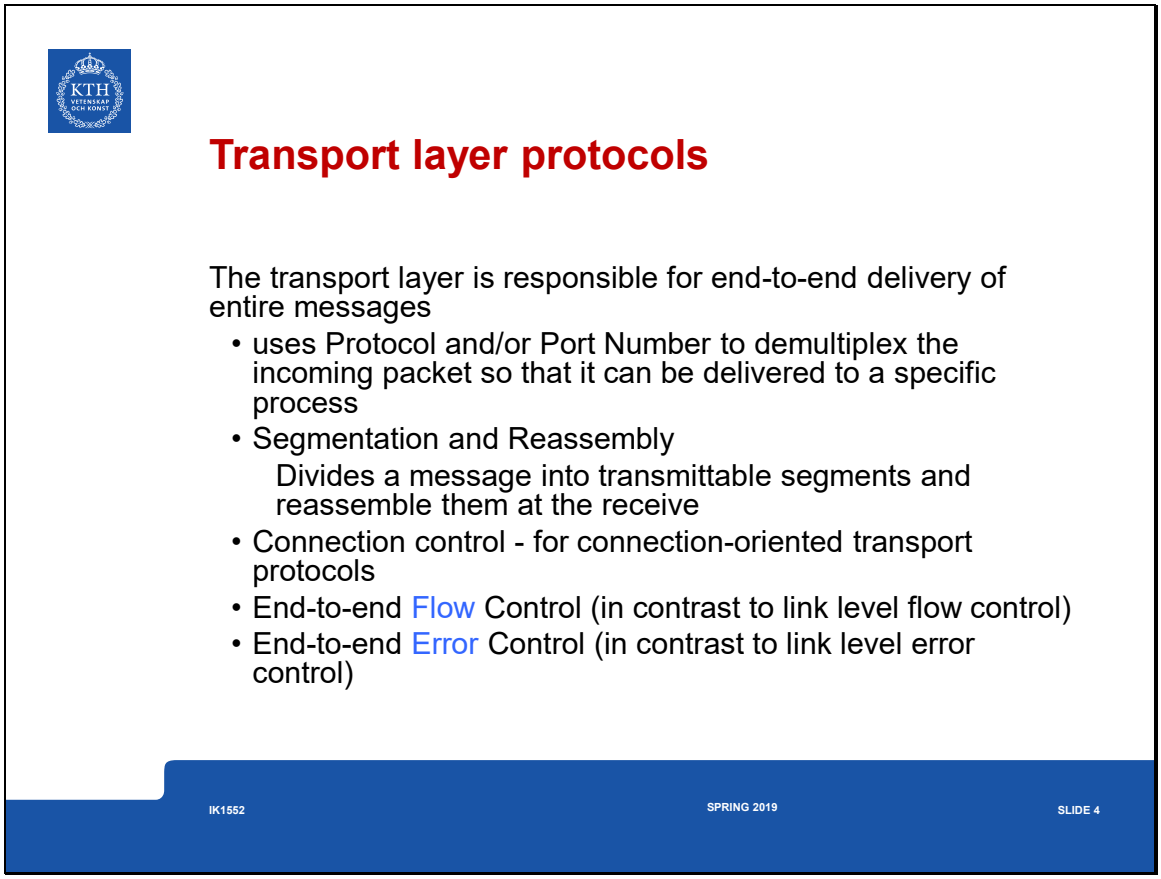
Slide 2


Module 4: UDP and friends
Lecture notes of G. Q. Maguire Jr.
For use in conjunction with James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach.
IK1552
SPRING 2019
SLIDE 2

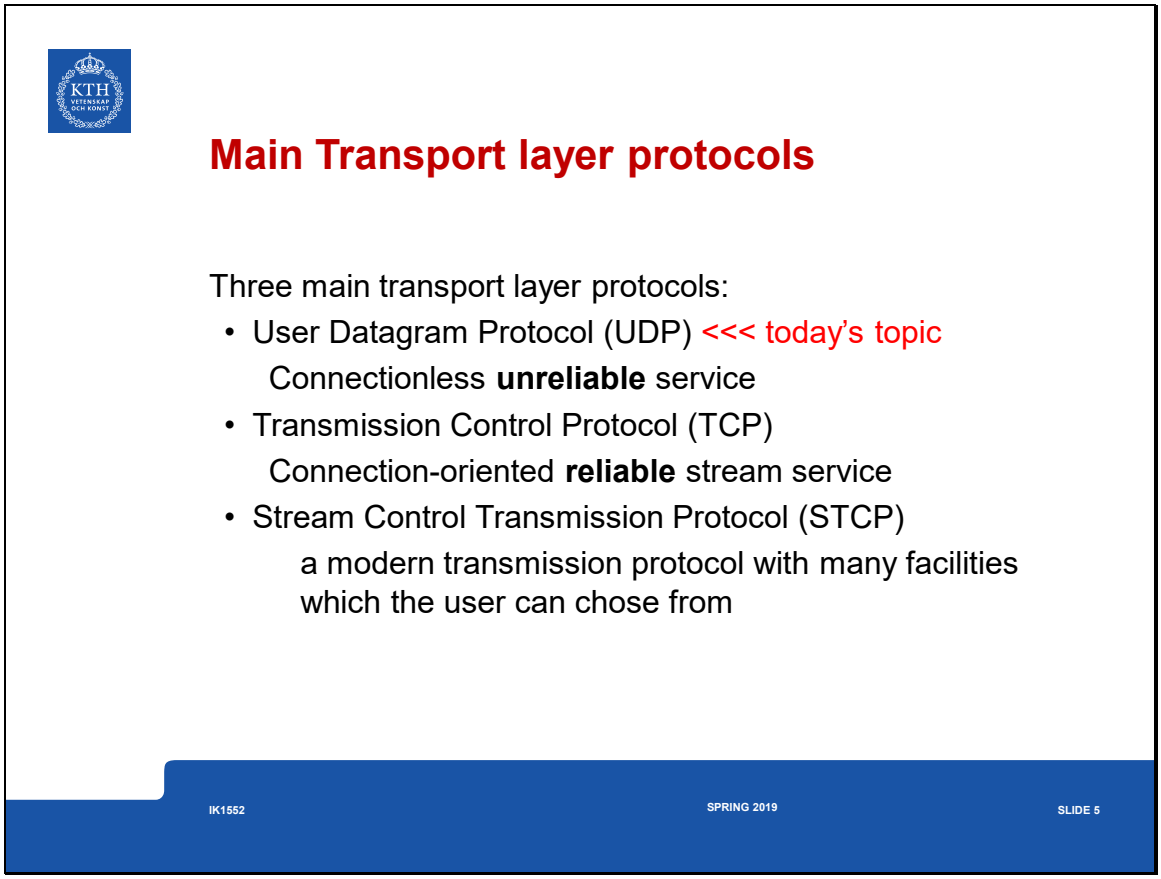
Slide 3


Outline
• UDP
• Socket API
• BOOTP
• DHCP
• DNS, DDNS
IK1552
SPRING 2019
SLIDE 3

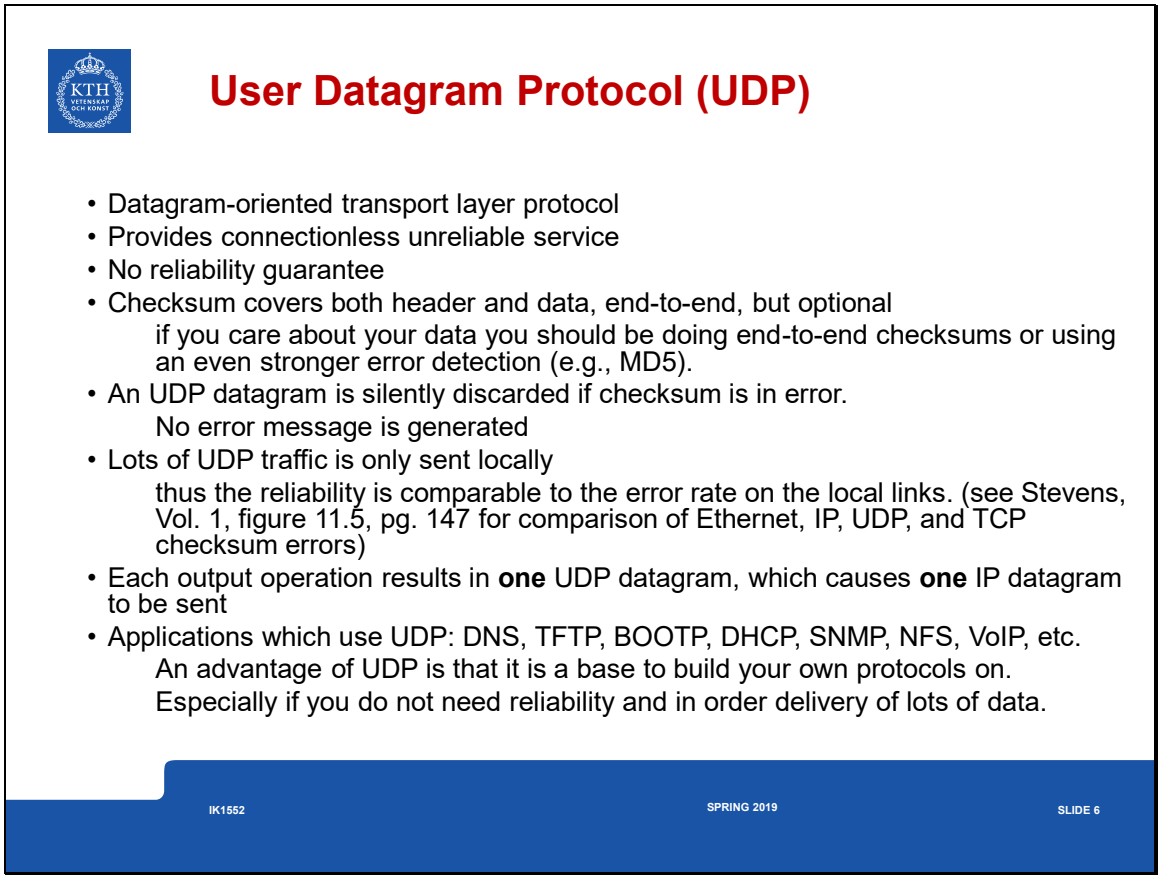
Slide 4


Transport layer protocols
The transport layer is responsible for end-to-end delivery of entire messages
• uses Protocol and/or Port Number to demultiplex the incoming packet so that it can be delivered to a specific process
• Segmentation and Reassembly
Divides a message into transmittable segments and reassemble them at the receive
• Connection control - for connection-oriented transport protocols
• End-to-end Flow Control (in contrast to link level flow control)
• End-to-end Error Control (in contrast to link level error control)
IK1552
SPRING 2019
SLIDE 4

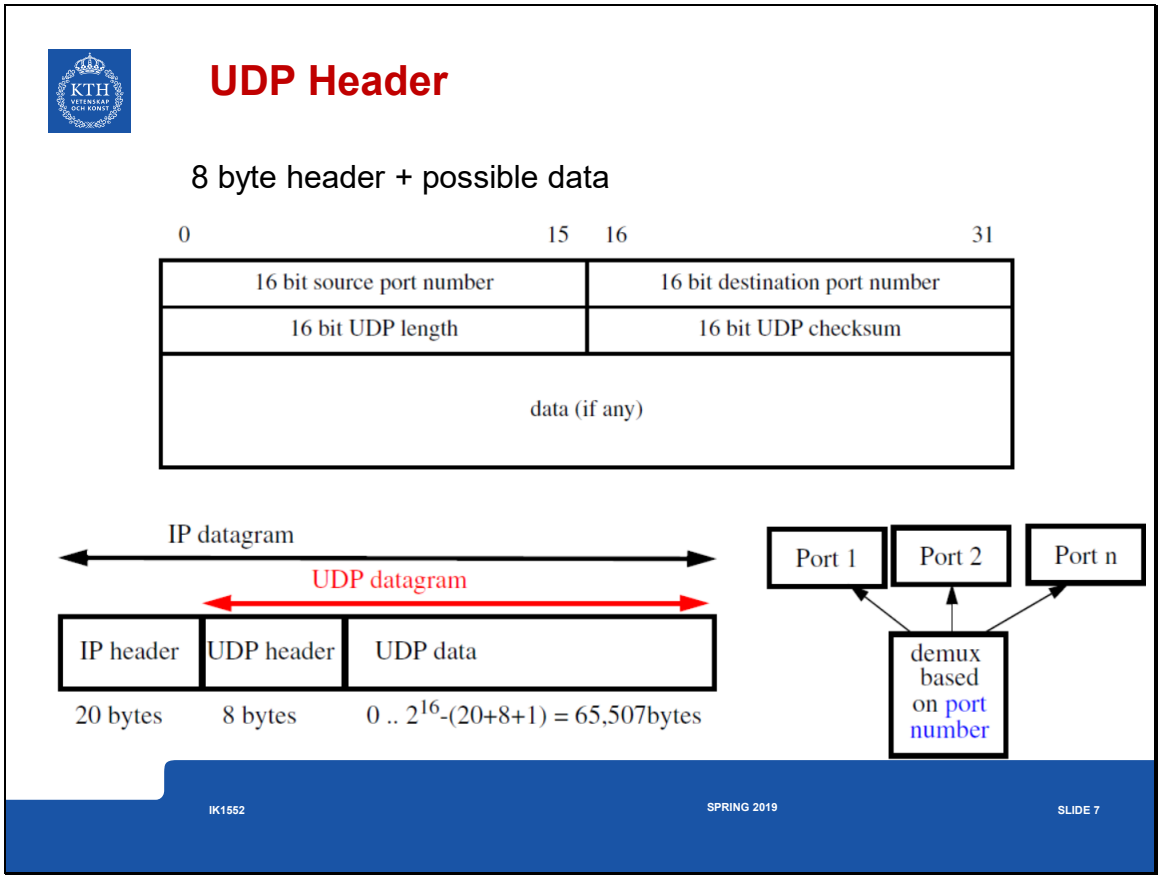
Slide 5


Main Transport layer protocols
Three main transport layer protocols:
• User Datagram Protocol (UDP) <<< today's topic
Connectionless unreliable service
• Transmission Control Protocol (TCP)
Connection-oriented reliable stream service
• Stream Control Transmission Protocol (STCP)
a modern transmission protocol with many facilities which the user can chose from
IK1552
SPRING 2019
SLIDE 5

Slide 6


User Datagram Protocol (UDP)
• Datagram-oriented transport layer protocol
• Provides connectionless unreliable service
• No reliability guarantee
• Checksum covers both header and data, end-to-end, but optional
if you care about your data you should be doing end-to-end checksums or using an even stronger error detection (e.g., MD5).
• An UDP datagram is silently discarded if checksum is in error.
No error message is generated
• Lots of UDP traffic is only sent locally
thus the reliability is comparable to the error rate on the local links. (see Stevens, Vol. 1, figure 11.5, pg. 147 for comparison of Ethernet, IP, UDP, and TCP checksum errors)
• Each output operation results in one UDP datagram, which causes one IP datagram to be sent
• Applications which use UDP: DNS, TFTP, BOOTP, DHCP, SNMP, NFS, VoIP, etc.
An advantage of UDP is that it is a base to build your own protocols on.
Especially if you do not need reliability and in order delivery of lots of data.
IK1552
SPRING 2019
SLIDE 6

Slide 7


UDP Header
8 byte header + possible data
0
15
16
31
16 bit source port number
16 bit destination port number
16 bit UDP length
16 bit UDP checksum
data (if any)
IP datagram
UDP datagram
IP header
UDP header
UDP data
20 bytes
8 bytes
0 .. 2^16-(20+8+1) = 65,507bytes
Port 1
Port 2
Port n
demux based on port number
IK1552
SPRING 2019
SLIDE 7

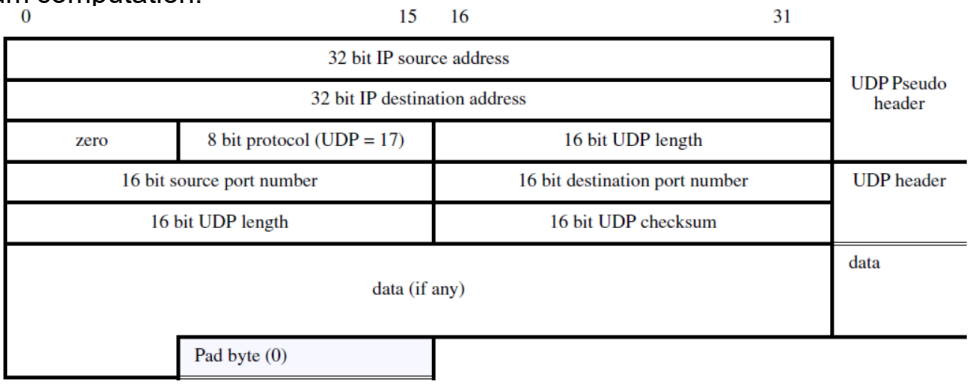
Slide 8

UDP Checksum and Pseudo-Header

UDP checksum covers more info than is present in the UDP datagram alone: **pseudo-header** and **pad byte** (0) {to even number of 16 bit words}.

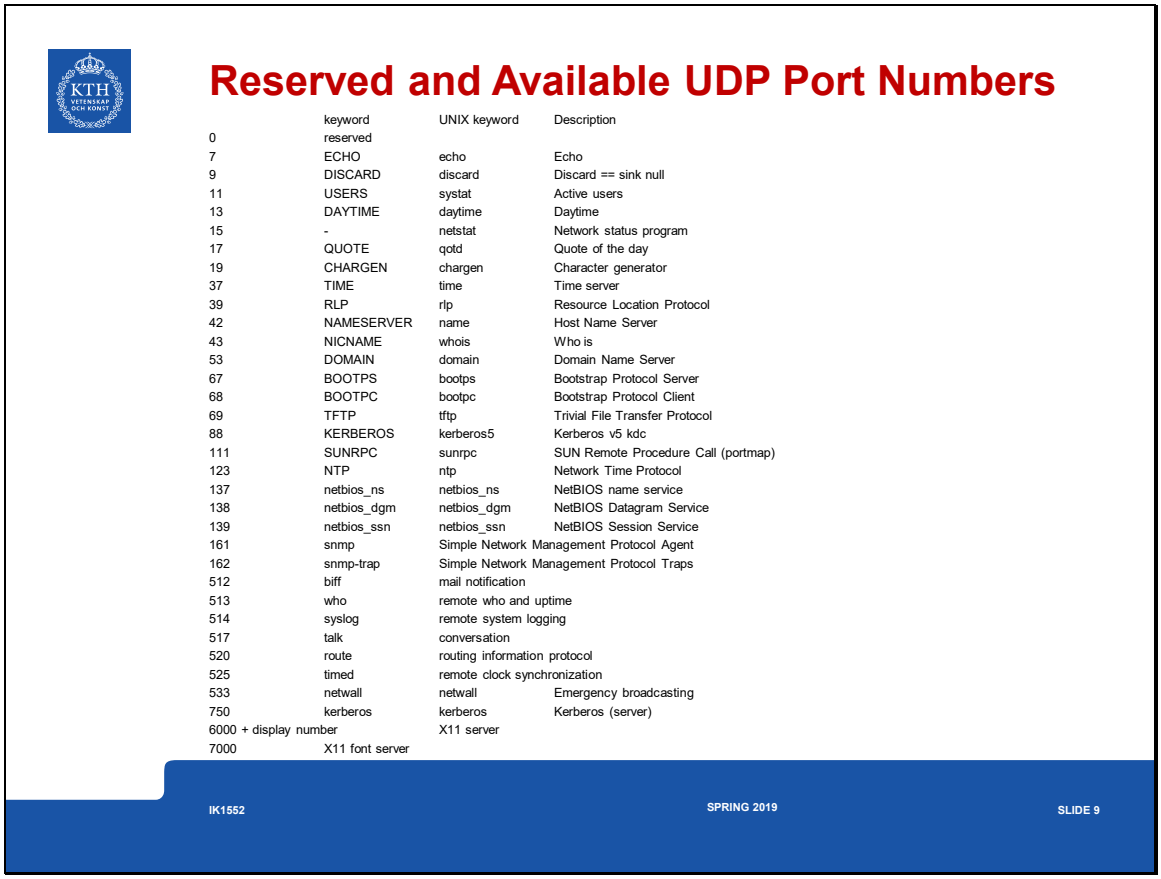
Propose: to verify the UDP datagram reached its correct destination: right port number at the right IP address.

Pseudo-header and pad byte are not transmitted with the UDP datagram, only used for checksum computation.



IK1552 SPRING 2019 SLIDE 8

Slide 9

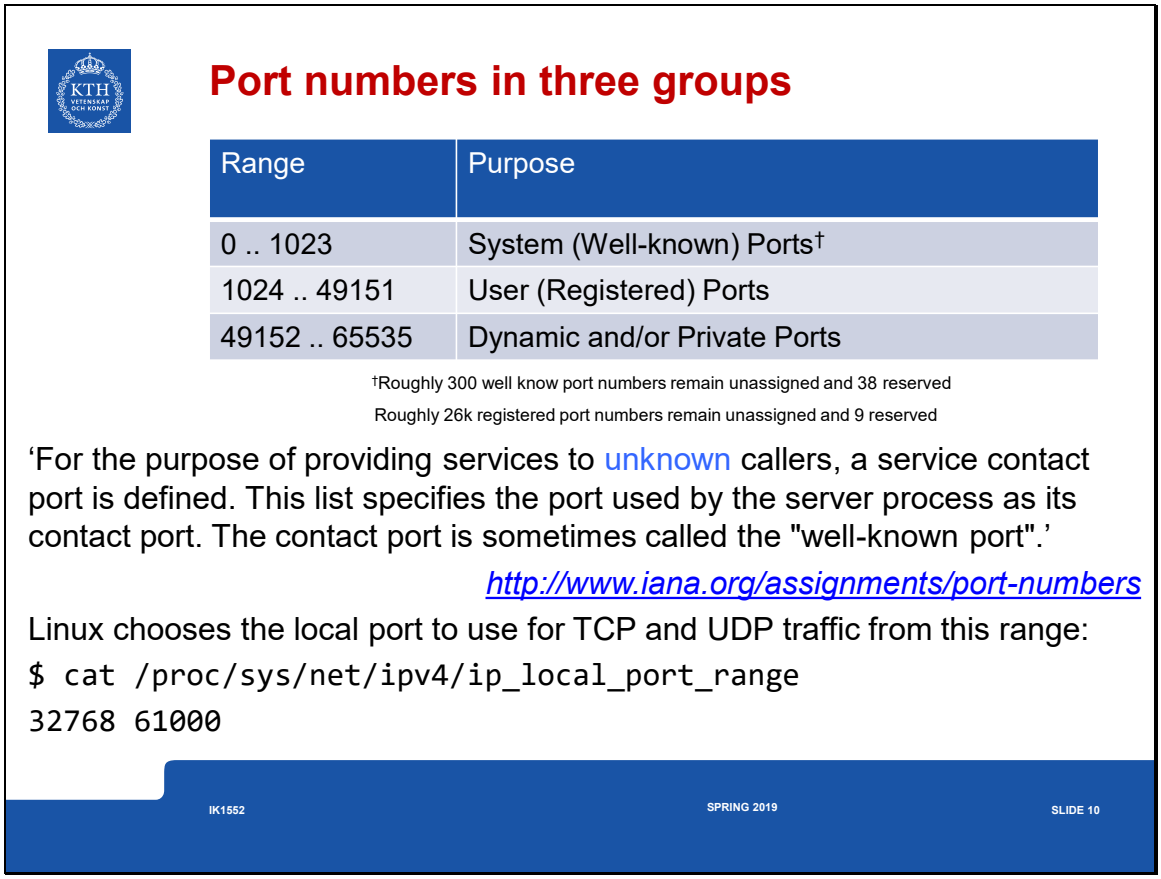


Reserved and Available UDP Port Numbers

	keyword	UNIX keyword	Description
0	reserved		
7	ECHO	echo	Echo
9	DISCARD	discard	Discard == sink null
11	USERS	systat	Active users
13	DAYTIME	daytime	Daytime
15	-	netstat	Network status program
17	QUOTE	qotd	Quote of the day
19	CHARGEN	chargen	Character generator
37	TIME	time	Time server
39	RLP	rlp	Resource Location Protocol
42	NAMESERVER	name	Host Name Server
43	NICNAME	whois	Who is
53	DOMAIN	domain	Domain Name Server
67	BOOTPS	bootps	Bootstrap Protocol Server
68	BOOTPC	bootpc	Bootstrap Protocol Client
69	TFTP	tftp	Trivial File Transfer Protocol
88	KERBEROS	kerberos5	Kerberos v5 kdc
111	SUNRPC	sunrpc	SUN Remote Procedure Call (portmap)
123	NTP	ntp	Network Time Protocol
137	netbios_ns	netbios_ns	NetBIOS name service
138	netbios_dgm	netbios_dgm	NetBIOS Datagram Service
139	netbios_ssn	netbios_ssn	NetBIOS Session Service
161	snmp	Simple Network Management Protocol Agent	
162	snmp-trap	Simple Network Management Protocol Traps	
512	biff	mail notification	
513	who	remote who and uptime	
514	syslog	remote system logging	
517	talk	conversation	
520	route	routing information protocol	
525	timed	remote clock synchronization	
533	netwall	netwall	Emergency broadcasting
750	kerberos	kerberos	Kerberos (server)
6000 + display number		X11 server	
7000	X11 font server		

IK1552 SPRING 2019 SLIDE 9

Slide 10



Port numbers in three groups

Range	Purpose
0 .. 1023	System (Well-known) Ports†
1024 .. 49151	User (Registered) Ports
49152 .. 65535	Dynamic and/or Private Ports

†Roughly 300 well know port numbers remain unassigned and 38 reserved

Roughly 26k registered port numbers remain unassigned and 9 reserved

‘For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the "well-known port".’

http://www.iana.org/assignments/port-numbers

Linux chooses the local port to use for TCP and UDP traffic from this range:

```
$ cat /proc/sys/net/ipv4/ip_local_port_range
32768 61000
```

IK1552 SPRING 2019 SLIDE 10

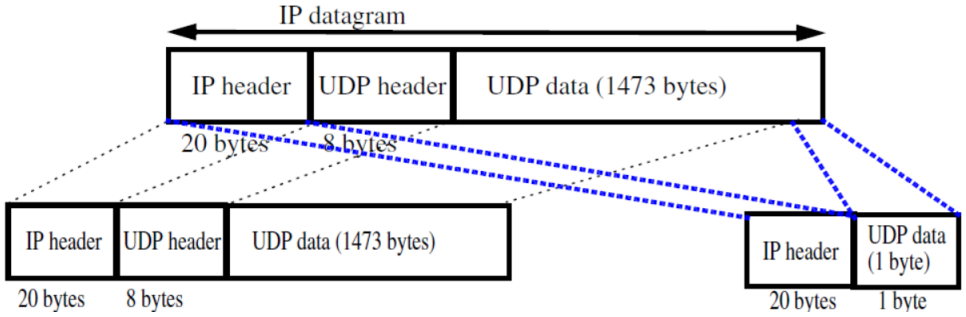
Slide 11

MTU and Datagram Fragmentation

If datagram size > MTU, perform fragmentation.

- At sending host or at intermediate router (IPv4).
- Reassembled only at final destination.

Example: 1501 (20 + 8 + 1473 data) on Ethernet (MTU=1500):



Note there is no UDP header in the second fragment.
Therefore, a frequent operation is to compute the path MTU before sending anything else. (see RFC 1191 for the table of common MTUs)

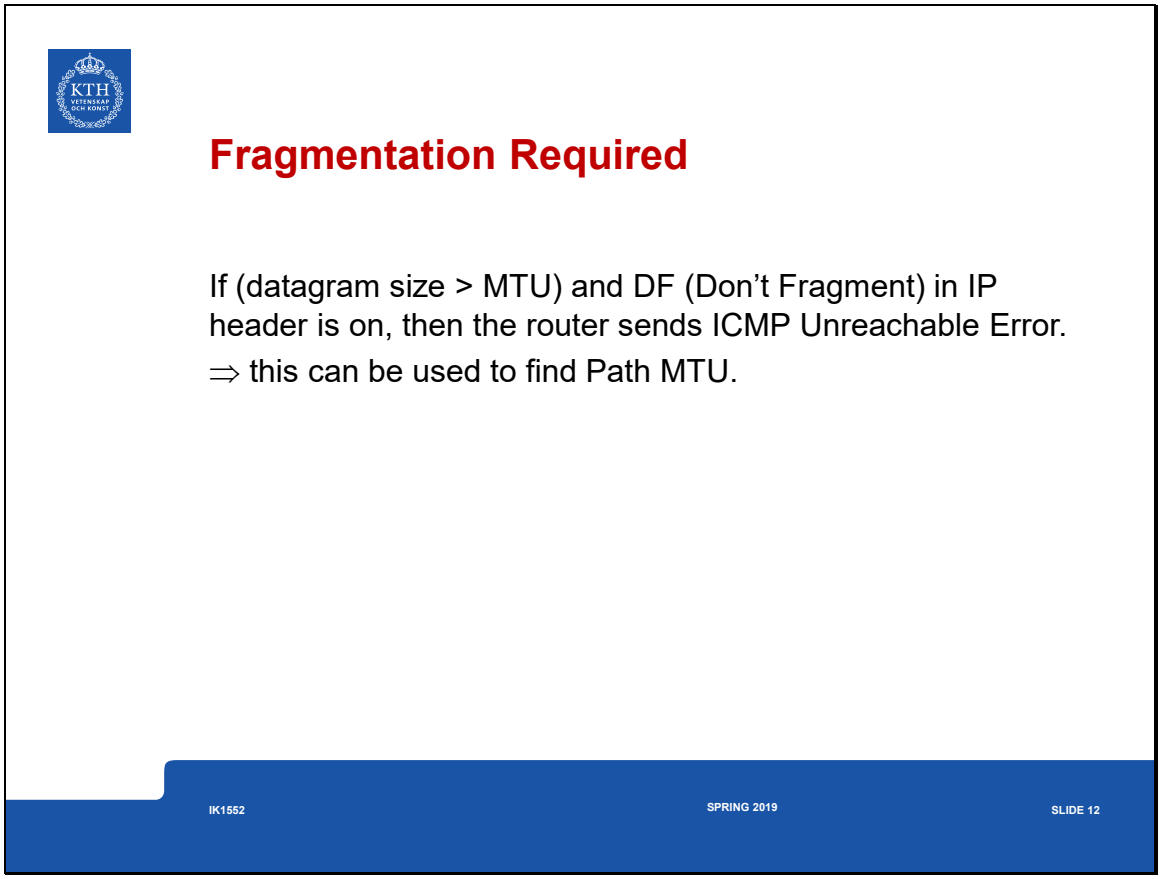
IK1552 SPRING 2019 SLIDE 11

J. C. Mogul and S. E. Deering, 'Path MTU discovery', *Internet Request for Comments*, vol. RFC 1191 (Draft Standard), Nov. 1990 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1191.txt

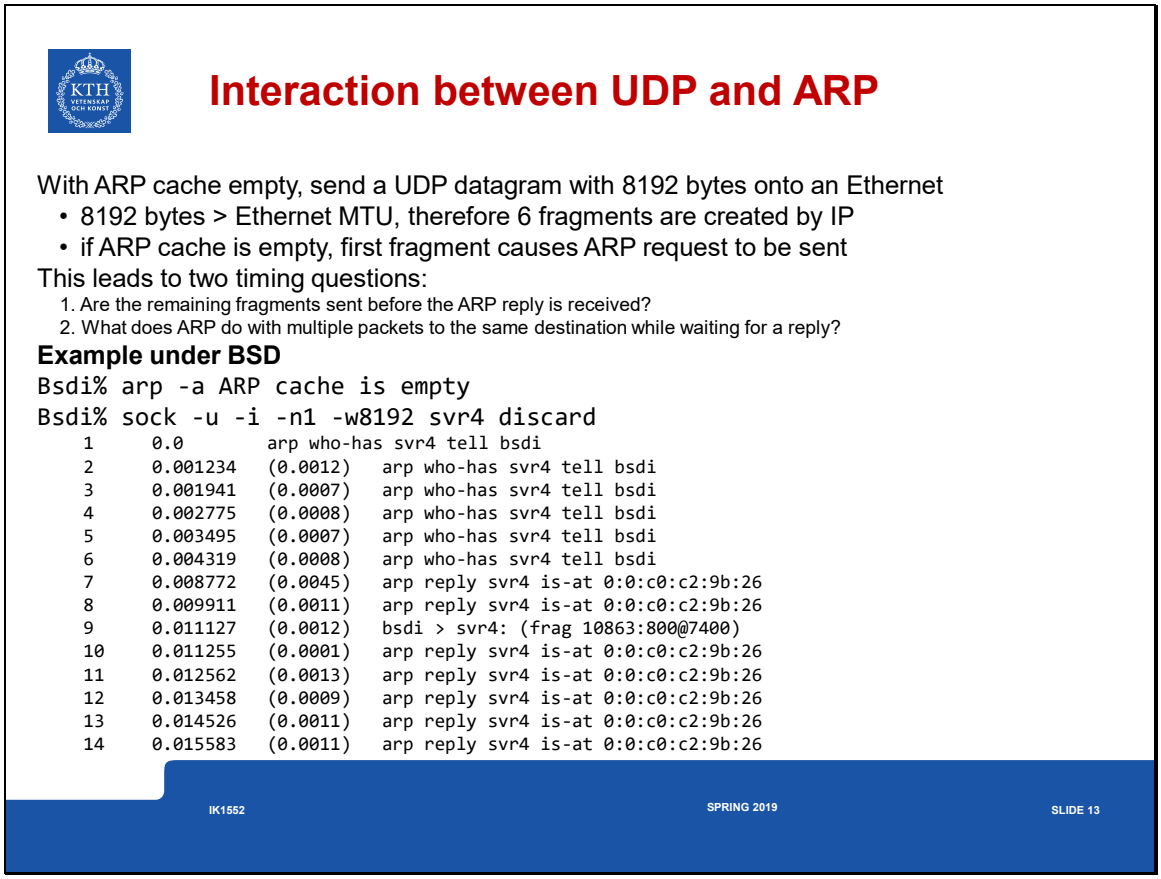
J. McCann, S. Deering, and J. Mogul, 'Path MTU Discovery for IP version 6', *Internet Request for Comments*, vol. RFC 1981 (Draft Standard), Aug. 1996 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1981.txt

B. Black and K. Kompella, 'Maximum Transmission Unit Signalling Extensions for the Label Distribution Protocol', *Internet Request for Comments*, vol. RFC 3988 (Experimental), Jan. 2005 [Online]. Available: http://www.rfc-editor.org/rfc/rfc3988.txt

M. Mathis and J. Heffner, 'Packetization Layer Path MTU Discovery', *Internet Request for Comments*, vol. RFC 4821 (Proposed Standard), Mar. 2007 [Online]. Available: http://www.rfc-editor.org/rfc/rfc4821.txt


Fragmentation Required
If (datagram size > MTU) and DF (Don't Fragment) in IP header is on, then the router sends ICMP Unreachable Error.
⇒ this can be used to find Path MTU.
IK1552
SPRING 2019
SLIDE 12

Slide 13



Interaction between UDP and ARP

With ARP cache empty, send a UDP datagram with 8192 bytes onto an Ethernet

- 8192 bytes > Ethernet MTU, therefore 6 fragments are created by IP
- if ARP cache is empty, first fragment causes ARP request to be sent

This leads to two timing questions:

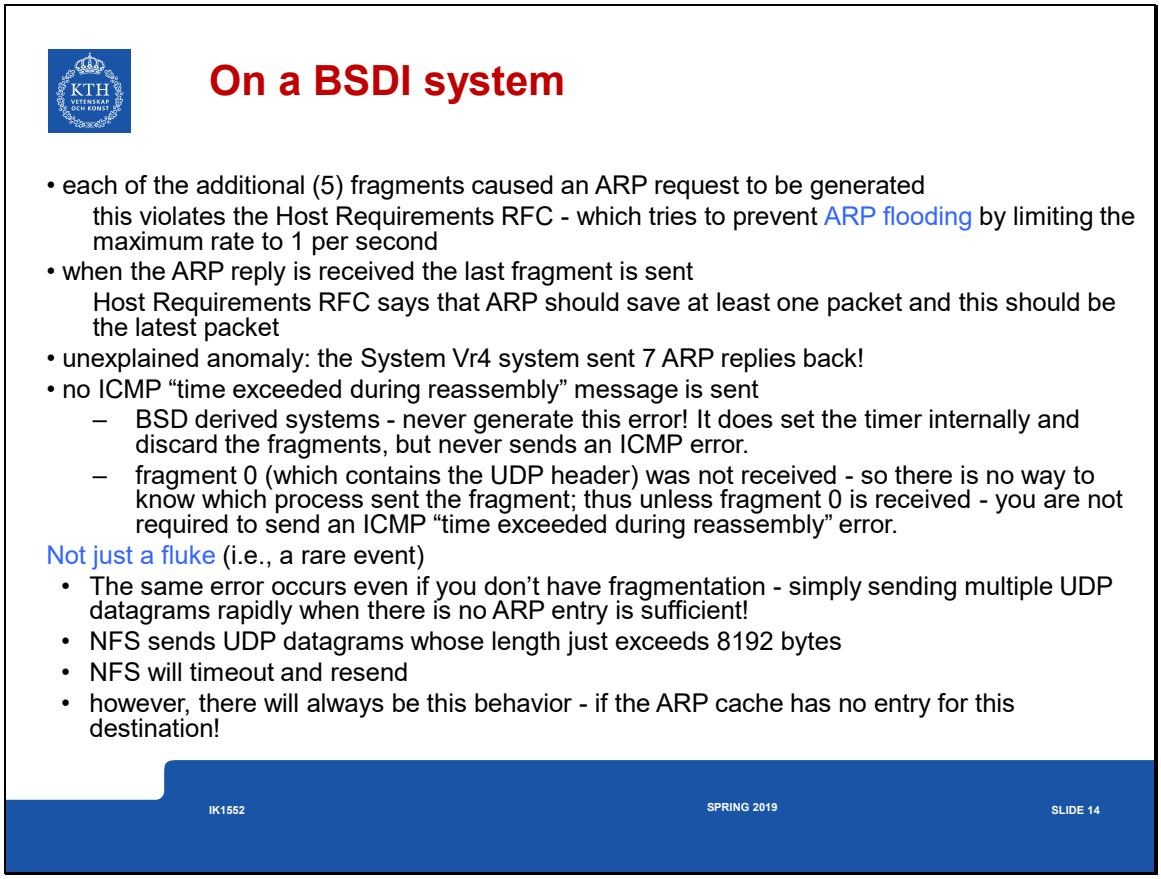
1. Are the remaining fragments sent before the ARP reply is received?
2. What does ARP do with multiple packets to the same destination while waiting for a reply?

Example under BSD

```
Bsdi% arp -a ARP cache is empty
Bsdi% sock -u -i -n1 -w8192 svr4 discard
    1     0.0        arp who-has svr4 tell bsdi
    2     0.001234   (0.0012)   arp who-has svr4 tell bsdi
    3     0.001941   (0.0007)   arp who-has svr4 tell bsdi
    4     0.002775   (0.0008)   arp who-has svr4 tell bsdi
    5     0.003495   (0.0007)   arp who-has svr4 tell bsdi
    6     0.004319   (0.0008)   arp who-has svr4 tell bsdi
    7     0.008772   (0.0045)   arp reply svr4 is-at 0:0:c0:c2:9b:26
    8     0.009911   (0.0011)   arp reply svr4 is-at 0:0:c0:c2:9b:26
    9     0.011127   (0.0012)   bsdi > svr4: (frag 10863:800@7400)
    10    0.011255   (0.0001)   arp reply svr4 is-at 0:0:c0:c2:9b:26
    11    0.012562   (0.0013)   arp reply svr4 is-at 0:0:c0:c2:9b:26
    12    0.013458   (0.0009)   arp reply svr4 is-at 0:0:c0:c2:9b:26
    13    0.014526   (0.0011)   arp reply svr4 is-at 0:0:c0:c2:9b:26
    14    0.015583   (0.0011)   arp reply svr4 is-at 0:0:c0:c2:9b:26
```

IK1552 SPRING 2019 SLIDE 13

Slide 14



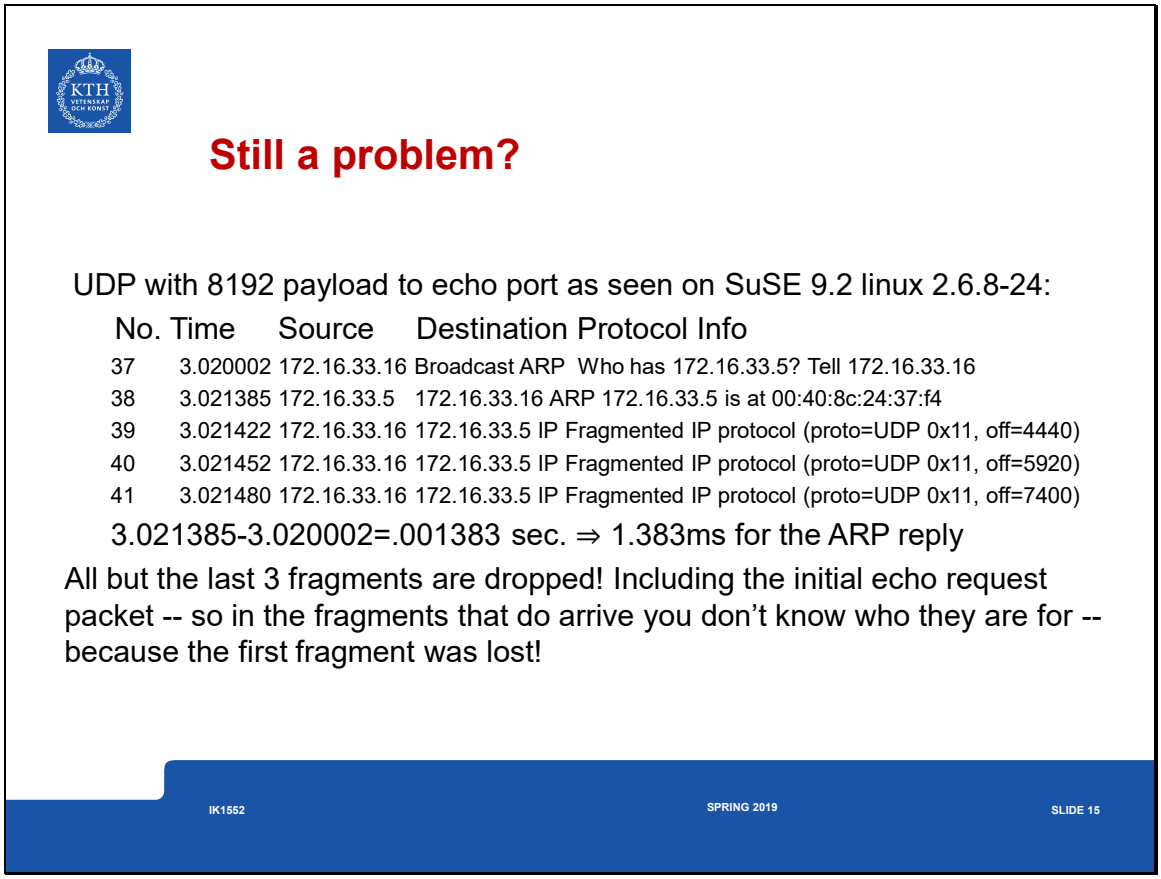
On a BSDI system

- each of the additional (5) fragments caused an ARP request to be generated
  this violates the Host Requirements RFC - which tries to prevent ARP flooding by limiting the maximum rate to 1 per second
- when the ARP reply is received the last fragment is sent
  Host Requirements RFC says that ARP should save at least one packet and this should be the latest packet
- unexplained anomaly: the System Vr4 system sent 7 ARP replies back!
- no ICMP “time exceeded during reassembly” message is sent
  - BSD derived systems - never generate this error! It does set the timer internally and discard the fragments, but never sends an ICMP error.
  - fragment 0 (which contains the UDP header) was not received - so there is no way to know which process sent the fragment; thus unless fragment 0 is received - you are not required to send an ICMP “time exceeded during reassembly” error.

Not just a fluke (i.e., a rare event)

- The same error occurs even if you don’t have fragmentation - simply sending multiple UDP datagrams rapidly when there is no ARP entry is sufficient!
- NFS sends UDP datagrams whose length just exceeds 8192 bytes
- NFS will timeout and resend
- however, there will always be this behavior - if the ARP cache has no entry for this destination!

IK1552 SPRING 2019 SLIDE 14



Still a problem?

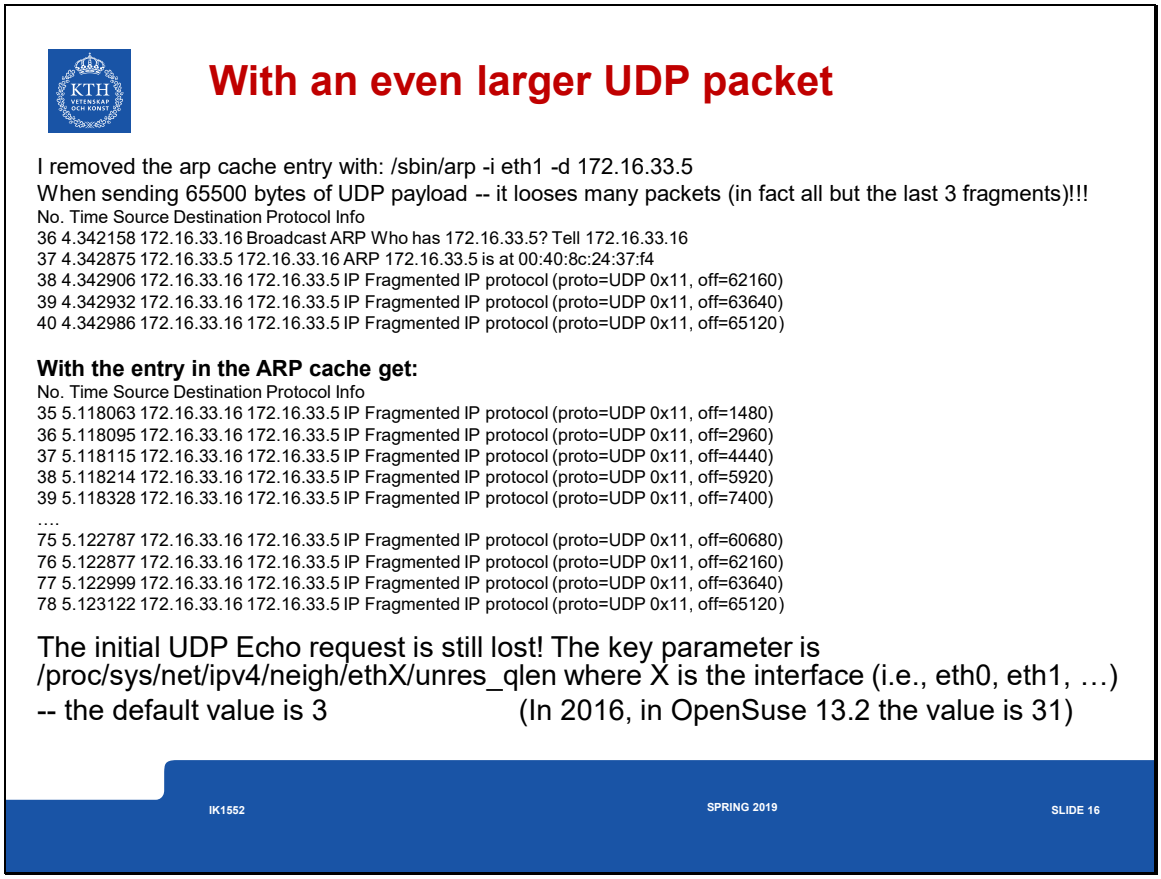
UDP with 8192 payload to echo port as seen on SuSE 9.2 linux 2.6.8-24:

No.	Time	Source	Destination	Protocol	Info
37	3.020002	172.16.33.16	Broadcast	ARP	Who has 172.16.33.5? Tell 172.16.33.16
38	3.021385	172.16.33.5	172.16.33.16	ARP	172.16.33.5 is at 00:40:8c:24:37:f4
39	3.021422	172.16.33.16	172.16.33.5	IP	Fragmented IP protocol (proto=UDP 0x11, off=4440)
40	3.021452	172.16.33.16	172.16.33.5	IP	Fragmented IP protocol (proto=UDP 0x11, off=5920)
41	3.021480	172.16.33.16	172.16.33.5	IP	Fragmented IP protocol (proto=UDP 0x11, off=7400)

3.021385-3.020002=.001383 sec. ⇒ 1.383ms for the ARP reply

All but the last 3 fragments are dropped! Including the initial echo request packet -- so in the fragments that do arrive you don't know who they are for -- because the first fragment was lost!

Slide 16



With an even larger UDP packet

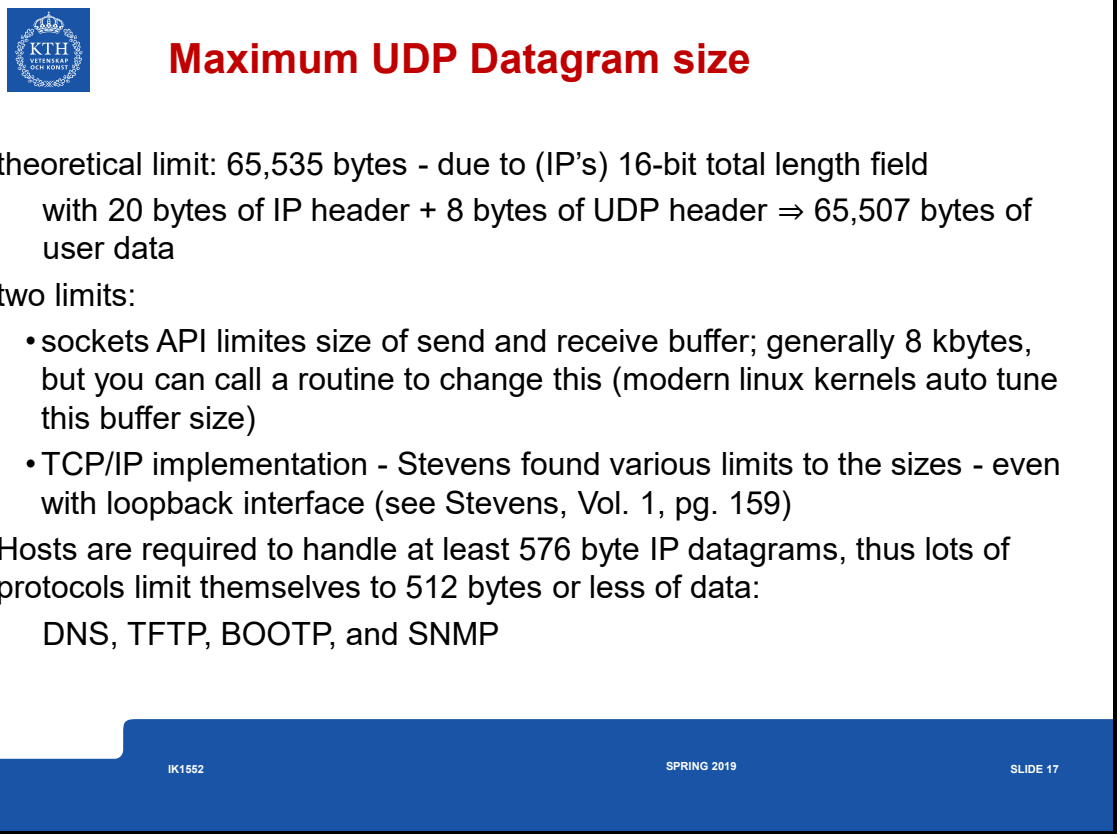
I removed the arp cache entry with: /sbin/arp -i eth1 -d 172.16.33.5
When sending 65500 bytes of UDP payload -- it looses many packets (in fact all but the last 3 fragments)!!!
No. Time Source Destination Protocol Info
36 4.342158 172.16.33.16 Broadcast ARP Who has 172.16.33.5? Tell 172.16.33.16
37 4.342875 172.16.33.5 172.16.33.16 ARP 172.16.33.5 is at 00:40:8c:24:37:f4
38 4.342906 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=62160)
39 4.342932 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=63640)
40 4.342986 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=65120)

With the entry in the ARP cache get:
No. Time Source Destination Protocol Info
35 5.118063 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=1480)
36 5.118095 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=2960)
37 5.118115 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=4440)
38 5.118214 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=5920)
39 5.118328 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=7400)
....
75 5.122787 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=60680)
76 5.122877 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=62160)
77 5.122999 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=63640)
78 5.123122 172.16.33.16 172.16.33.5 IP Fragmented IP protocol (proto=UDP 0x11, off=65120)

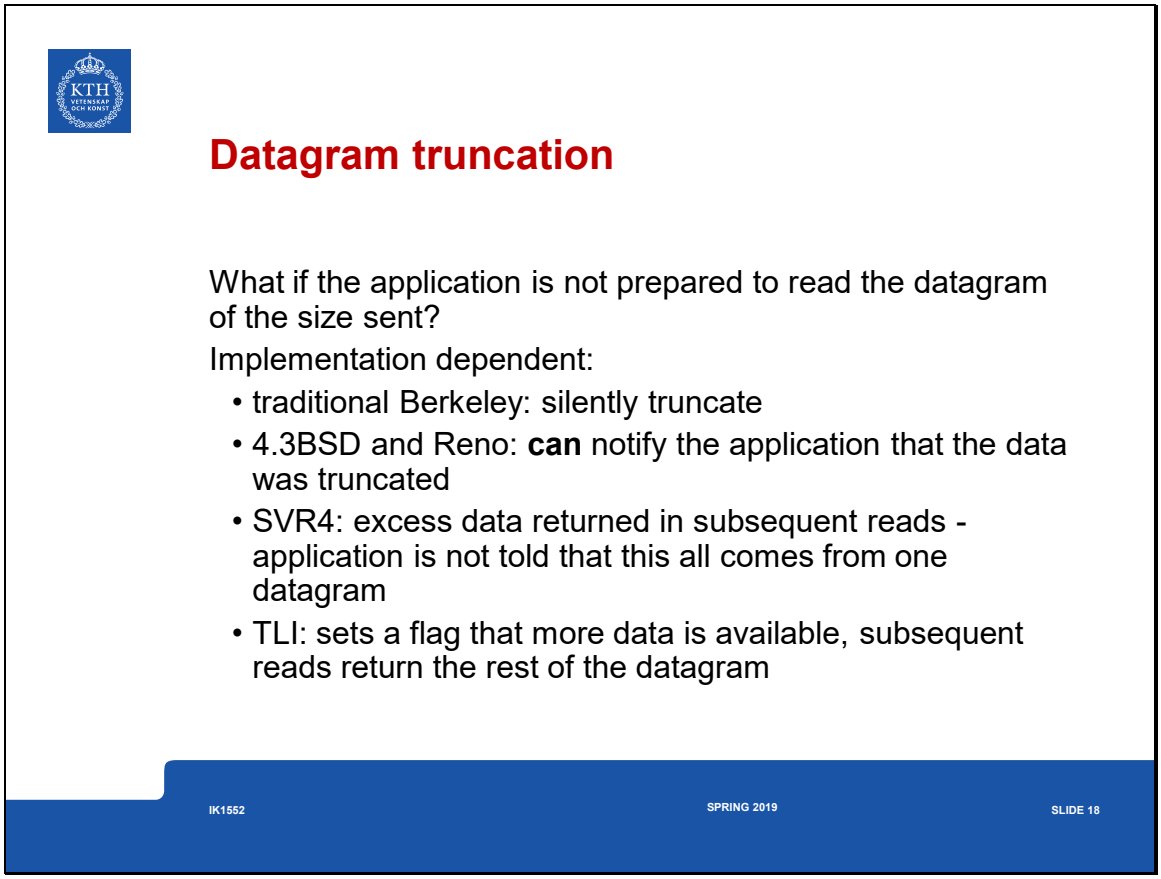
The initial UDP Echo request is still lost! The key parameter is /proc/sys/net/ipv4/neigh/ethX/unres_qlen where X is the interface (i.e., eth0, eth1, …)
-- the default value is 3 (In 2016, in OpenSuse 13.2 the value is 31)

IK1552 SPRING 2019 SLIDE 16

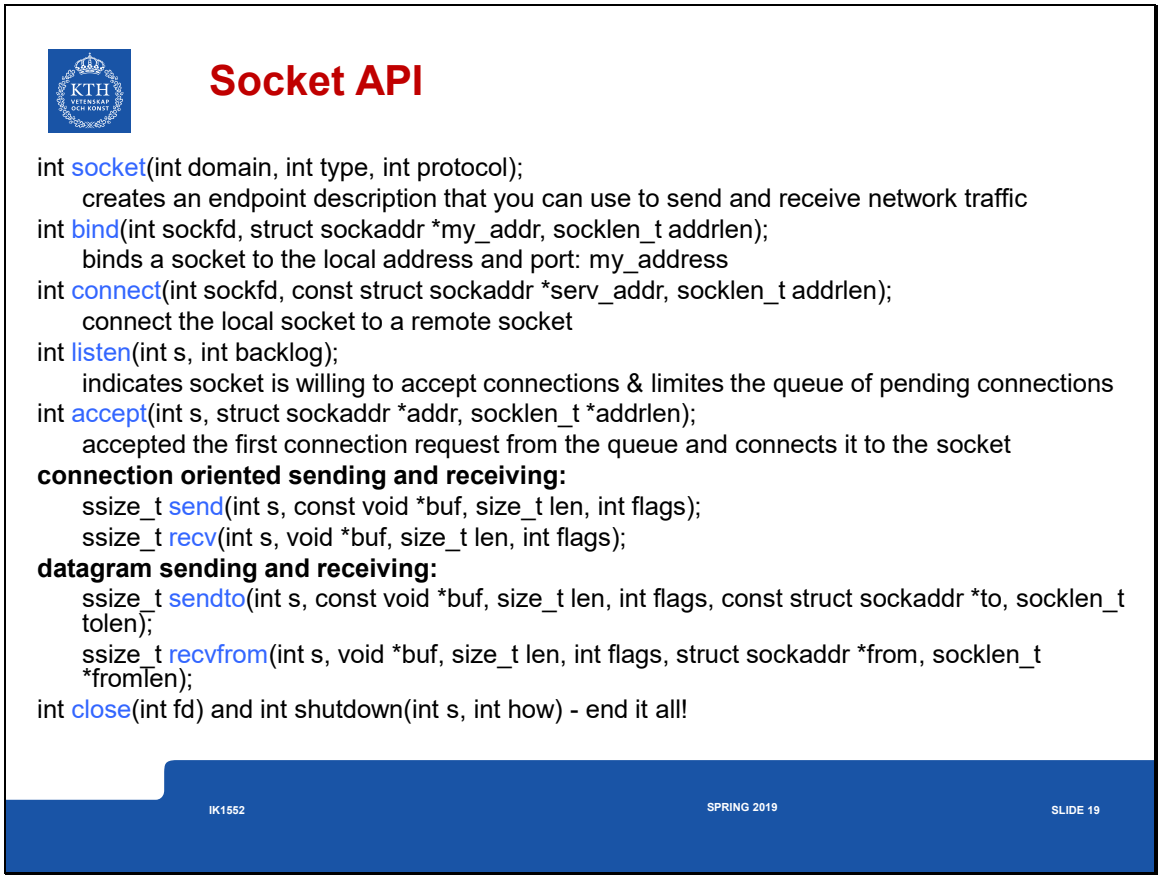
Slide 17

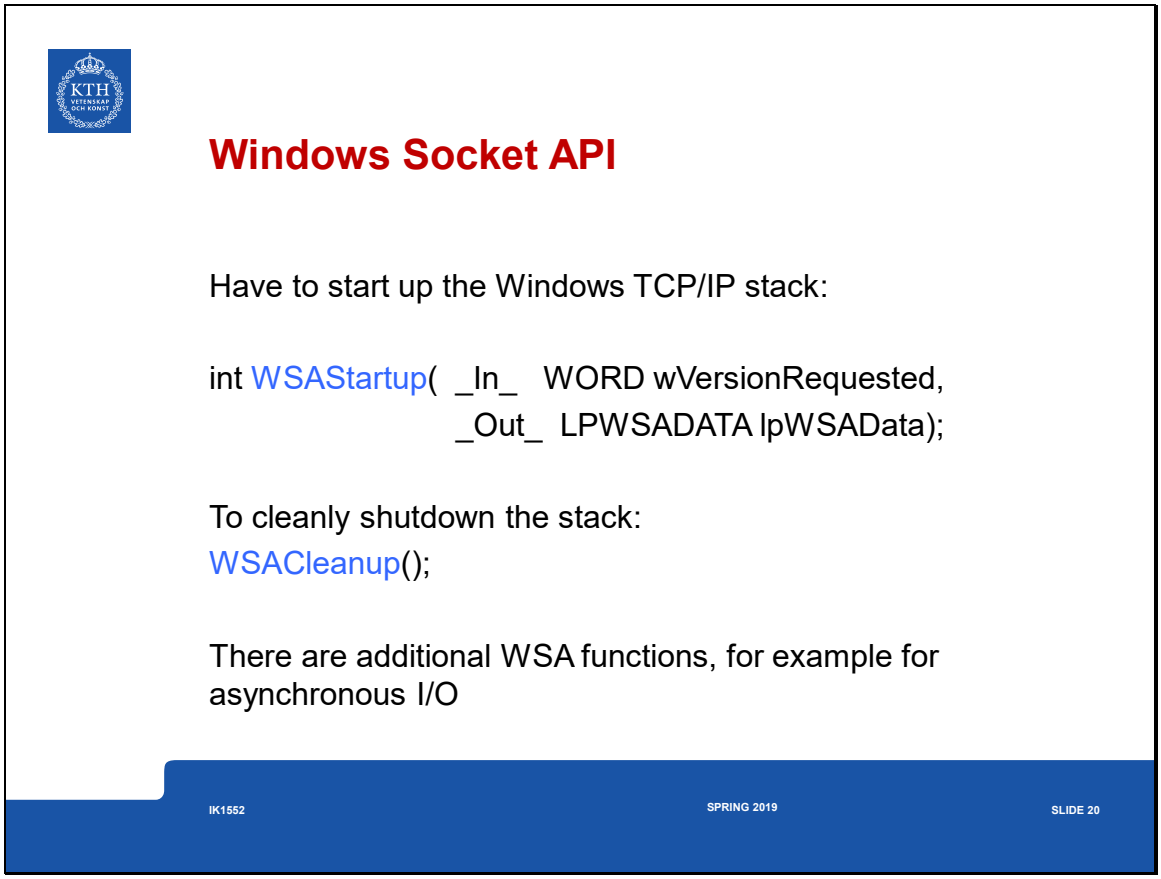

Maximum UDP Datagram size
theoretical limit: 65,535 bytes - due to (IP's) 16-bit total length field
with 20 bytes of IP header + 8 bytes of UDP header ⇒ 65,507 bytes of user data
two limits:
• sockets API limites size of send and receive buffer; generally 8 kbytes, but you can call a routine to change this (modern linux kernels auto tune this buffer size)
• TCP/IP implementation - Stevens found various limits to the sizes - even with loopback interface (see Stevens, Vol. 1, pg. 159)
Hosts are required to handle at least 576 byte IP datagrams, thus lots of protocols limit themselves to 512 bytes or less of data:
DNS, TFTP, BOOTP, and SNMP
IK1552
SPRING 2019
SLIDE 17

Slide 18

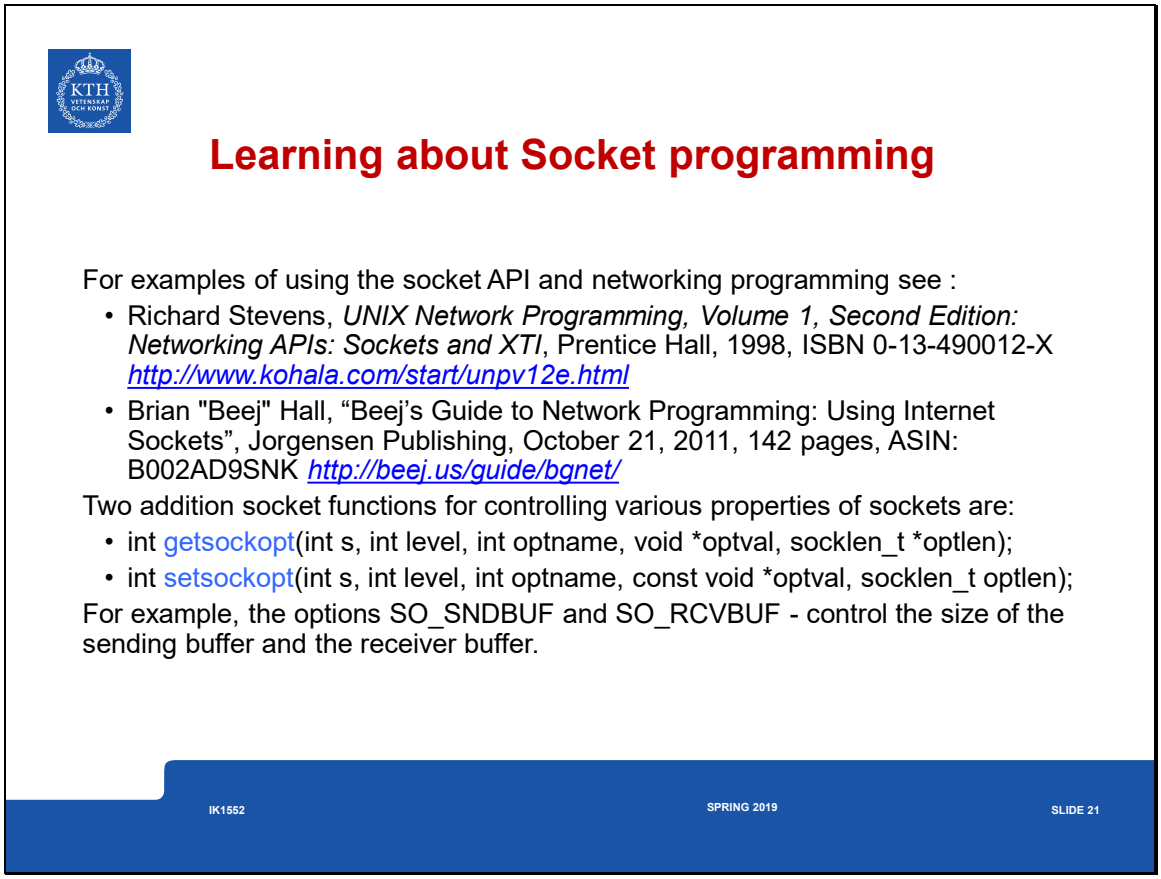

Datagram truncation
What if the application is not prepared to read the datagram of the size sent?
Implementation dependent:
• traditional Berkeley: silently truncate
• 4.3BSD and Reno: can notify the application that the data was truncated
• SVR4: excess data returned in subsequent reads - application is not told that this all comes from one datagram
• TLI: sets a flag that more data is available, subsequent reads return the rest of the datagram
IK1552
SPRING 2019
SLIDE 18

Slide 19


Socket API
int socket(int domain, int type, int protocol);
creates an endpoint description that you can use to send and receive network traffic
int bind(int sockfd, struct sockaddr *my_addr, socklen_t addrlen);
binds a socket to the local address and port: my_address
int connect(int sockfd, const struct sockaddr *serv_addr, socklen_t addrlen);
connect the local socket to a remote socket
int listen(int s, int backlog);
indicates socket is willing to accept connections & limites the queue of pending connections
int accept(int s, struct sockaddr *addr, socklen_t *addrlen);
accepted the first connection request from the queue and connects it to the socket
connection oriented sending and receiving:
ssize_t send(int s, const void *buf, size_t len, int flags);
ssize_t recv(int s, void *buf, size_t len, int flags);
datagram sending and receiving:
ssize_t sendto(int s, const void *buf, size_t len, int flags, const struct sockaddr *to, socklen_t tolen);
ssize_t recvfrom(int s, void *buf, size_t len, int flags, struct sockaddr *from, socklen_t *fromlen);
int close(int fd) and int shutdown(int s, int how) - end it all!
IK1552
SPRING 2019
SLIDE 19

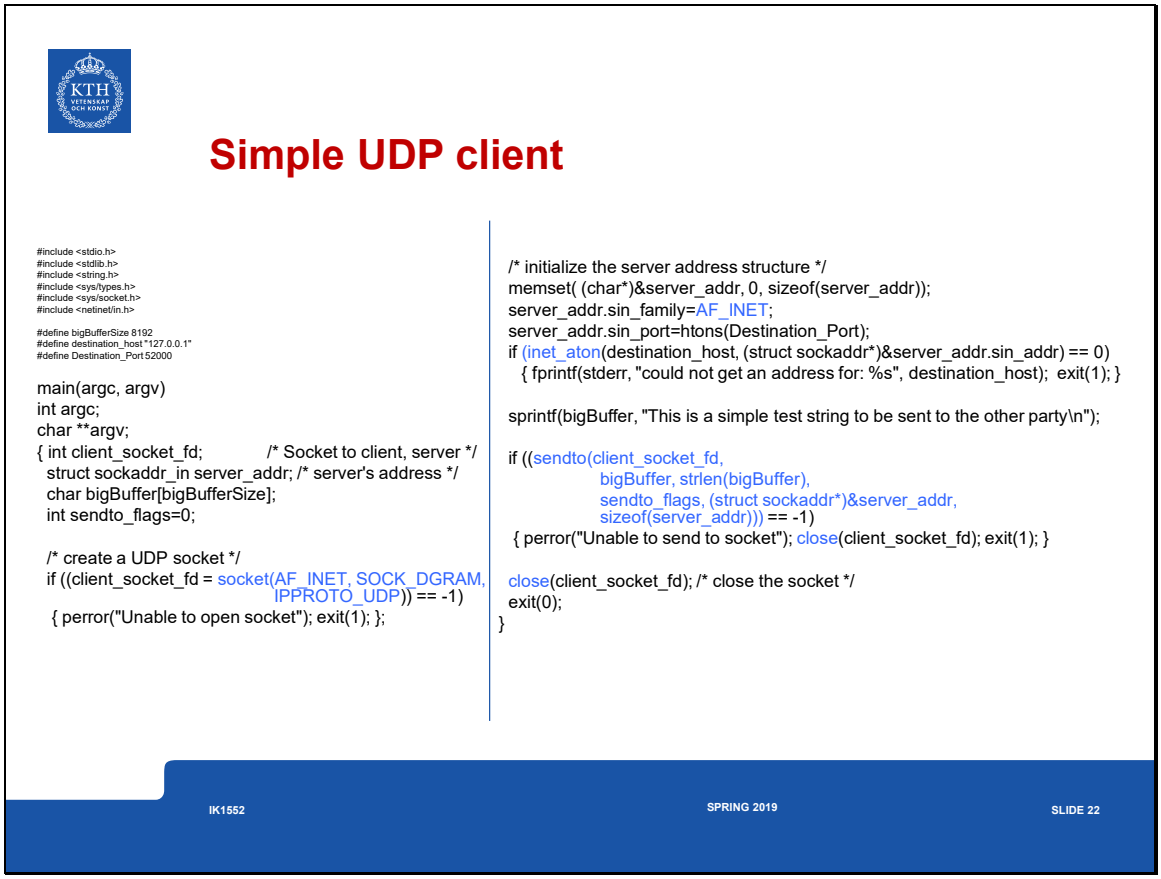

Windows Socket API
Have to start up the Windows TCP/IP stack:
int WSAStartup(_In_ WORD wVersionRequested,
Out LPWSADATA lpWSAData);
To cleanly shutdown the stack:
WSACleanup();
There are additional WSA functions, for example for asynchronous I/O
IK1552
SPRING 2019
SLIDE 20

Slide 21



Brian 'Beej' Hall, *Beej's Guide to Network Programming: Using Internet Sockets*. Jorgensen Publishing, 2011, ISBN: ASIN: B002AD9SNK [Online]. Available: http://beej.us/guide/bgnet/

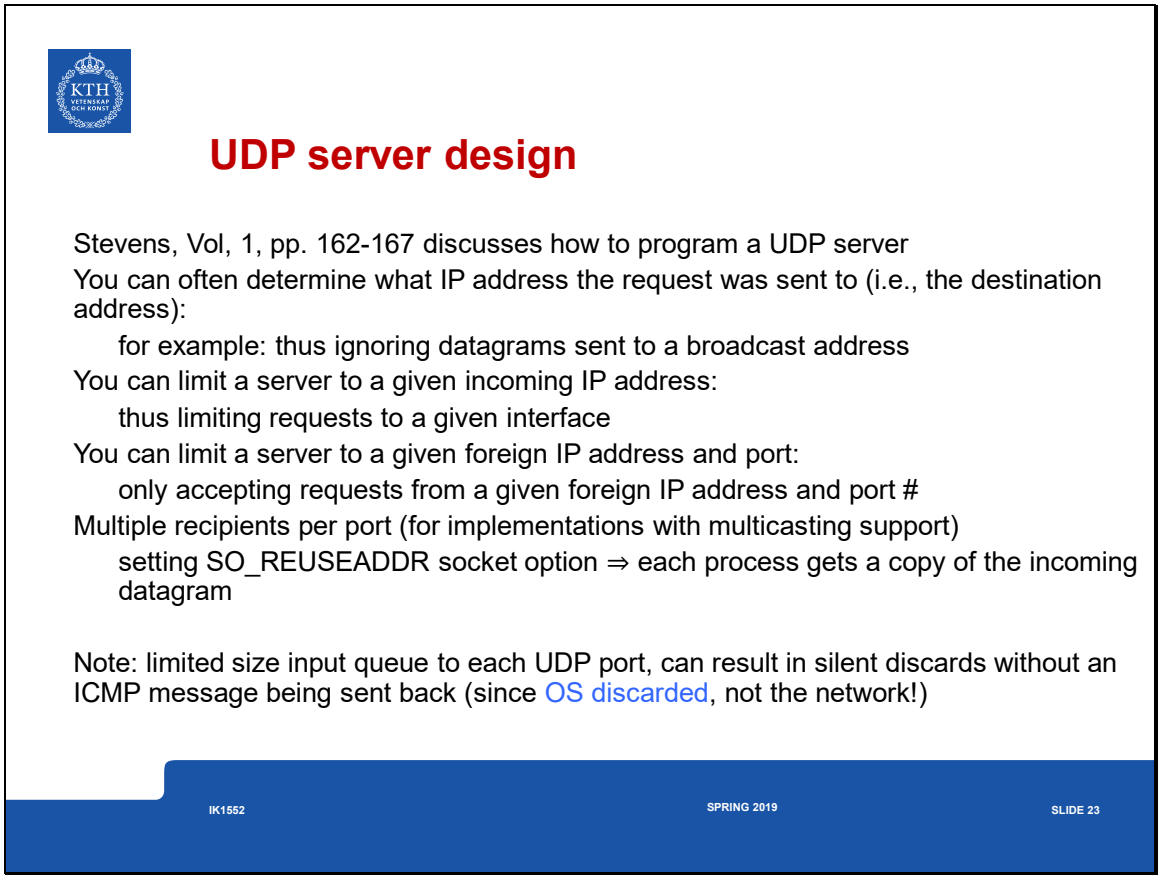
Slide 22



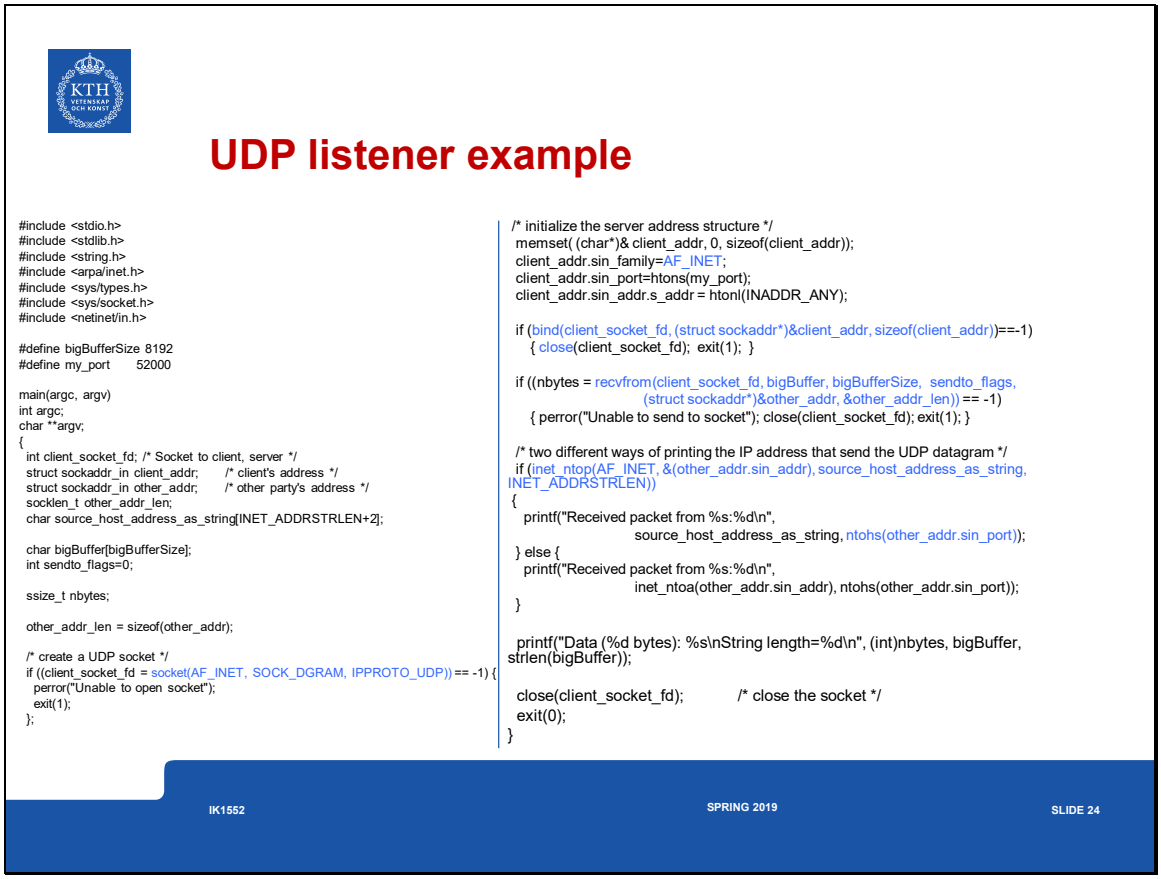
Simple UDP client

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define bigBufferSize 8192
#define destination_host "127.0.0.1"
#define Destination_Port 52000

main(argc, argv)
int argc;
char **argv;
{ int client_socket_fd;            /* Socket to client, server */
  struct sockaddr_in server_addr; /* server's address */
  char bigBuffer[bigBufferSize];
  int sendto_flags=0;

  /* create a UDP socket */
  if ((client_socket_fd = socket(AF_INET, SOCK_DGRAM,
                                 IPPROTO_UDP)) == -1)
   { perror("Unable to open socket"); exit(1); };

  /* initialize the server address structure */
  memset( (char*)&server_addr, 0, sizeof(server_addr));
  server_addr.sin_family=AF_INET;
  server_addr.sin_port=htons(Destination_Port);
  if (inet_aton(destination_host, (struct sockaddr*)&server_addr.sin_addr) == 0)
    { fprintf(stderr, "could not get an address for: %s", destination_host);  exit(1); }

  sprintf(bigBuffer, "This is a simple test string to be sent to the other party\n");

  if ((sendto(client_socket_fd,
              bigBuffer, strlen(bigBuffer),
              sendto_flags, (struct sockaddr*)&server_addr,
              sizeof(server_addr))) == -1)
   { perror("Unable to send to socket"); close(client_socket_fd); exit(1); }

  close(client_socket_fd); /* close the socket */
  exit(0);
}
```

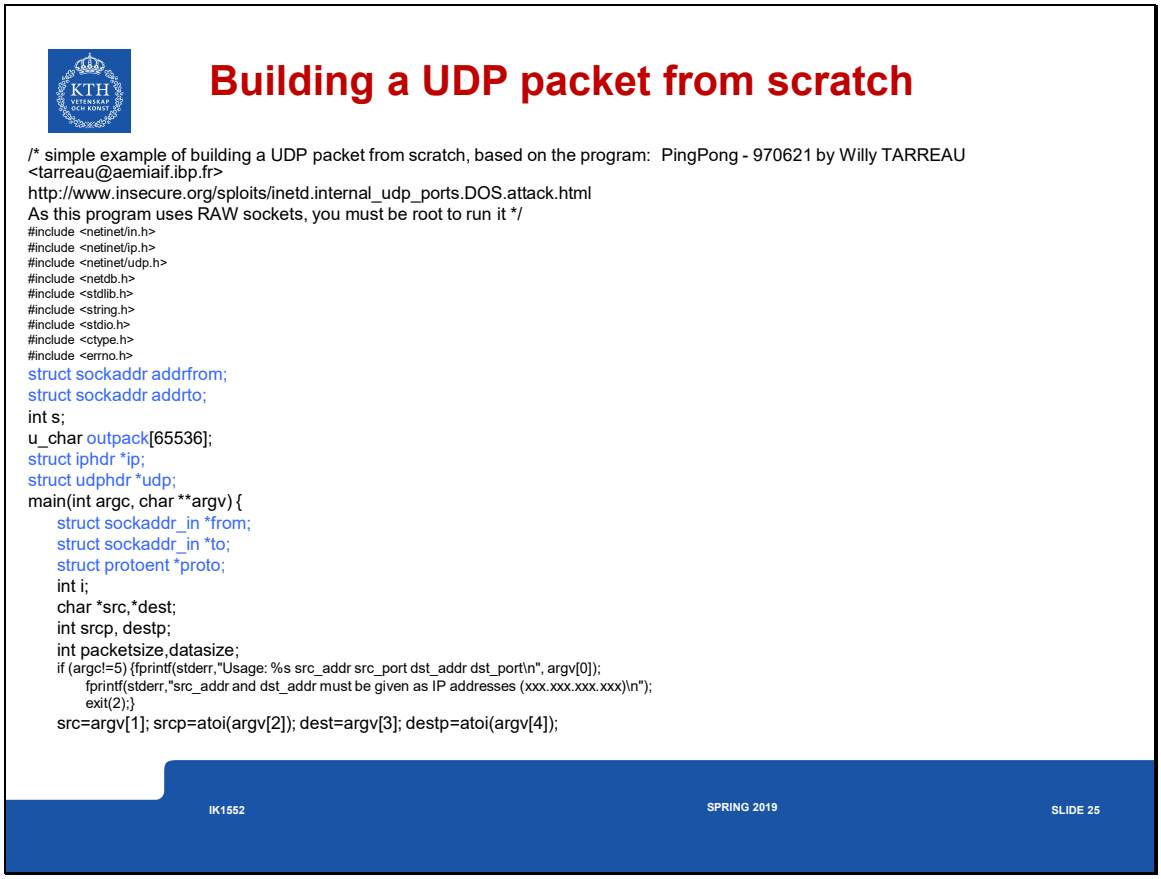
Slide 23


UDP server design
Stevens, Vol, 1, pp. 162-167 discusses how to program a UDP server
You can often determine what IP address the request was sent to (i.e., the destination address):
for example: thus ignoring datagrams sent to a broadcast address
You can limit a server to a given incoming IP address:
thus limiting requests to a given interface
You can limit a server to a given foreign IP address and port:
only accepting requests from a given foreign IP address and port #
Multiple recipients per port (for implementations with multicasting support)
setting SO_REUSEADDR socket option ⇒ each process gets a copy of the incoming datagram
Note: limited size input queue to each UDP port, can result in silent discards without an ICMP message being sent back (since OS discarded, not the network!)
IK1552
SPRING 2019
SLIDE 23

Slide 24


UDP listener example
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#define bigBufferSize 8192
#define my_port 52000
main(argc, argv)
int argc;
char **argv;
{
int client_socket_fd; /* Socket to client, server */
struct sockaddr_in client_addr; /* client's address */
struct sockaddr_in other_addr; /* other party's address */
socklen_t other_addr_len;
char source_host_address_as_string[INET_ADDRSTRLEN+2];
char bigBuffer[bigBufferSize];
int sendto_flags=0;
ssize_t nbytes;
other_addr_len = sizeof(other_addr);
/* create a UDP socket */
if ((client_socket_fd = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1) {
perror("Unable to open socket");
exit(1);
};
/* initialize the server address structure */
memset((char*)& client_addr, 0, sizeof(client_addr));
client_addr.sin_family=AF_INET;
client_addr.sin_port=htons(my_port);
client_addr.sin_addr.s_addr = htonl(INADDR_ANY);
if (bind(client_socket_fd, (struct sockaddr*)&client_addr, sizeof(client_addr))==-1)
{ close(client_socket_fd); exit(1); }
if ((nbytes = recvfrom(client_socket_fd, bigBuffer, bigBufferSize, sendto_flags,
(struct sockaddr*)&other_addr, &other_addr_len)) == -1)
{ perror("Unable to send to socket"); close(client_socket_fd); exit(1); }
/* two different ways of printing the IP address that send the UDP datagram */
if (inet_ntop(AF_INET, &(other_addr.sin_addr), source_host_address_as_string,
INET_ADDRSTRLEN))
{
printf("Received packet from %s:%d\n",
source_host_address_as_string, ntohs(other_addr.sin_port));
} else {
printf("Received packet from %s:%d\n",
inet_ntoa(other_addr.sin_addr), ntohs(other_addr.sin_port));
}
printf("Data (%d bytes): %s\nString length=%d\n", (int)nbytes, bigBuffer,
strlen(bigBuffer));
close(client_socket_fd); /* close the socket */
exit(0);
}
IK1552
SPRING 2019
SLIDE 24

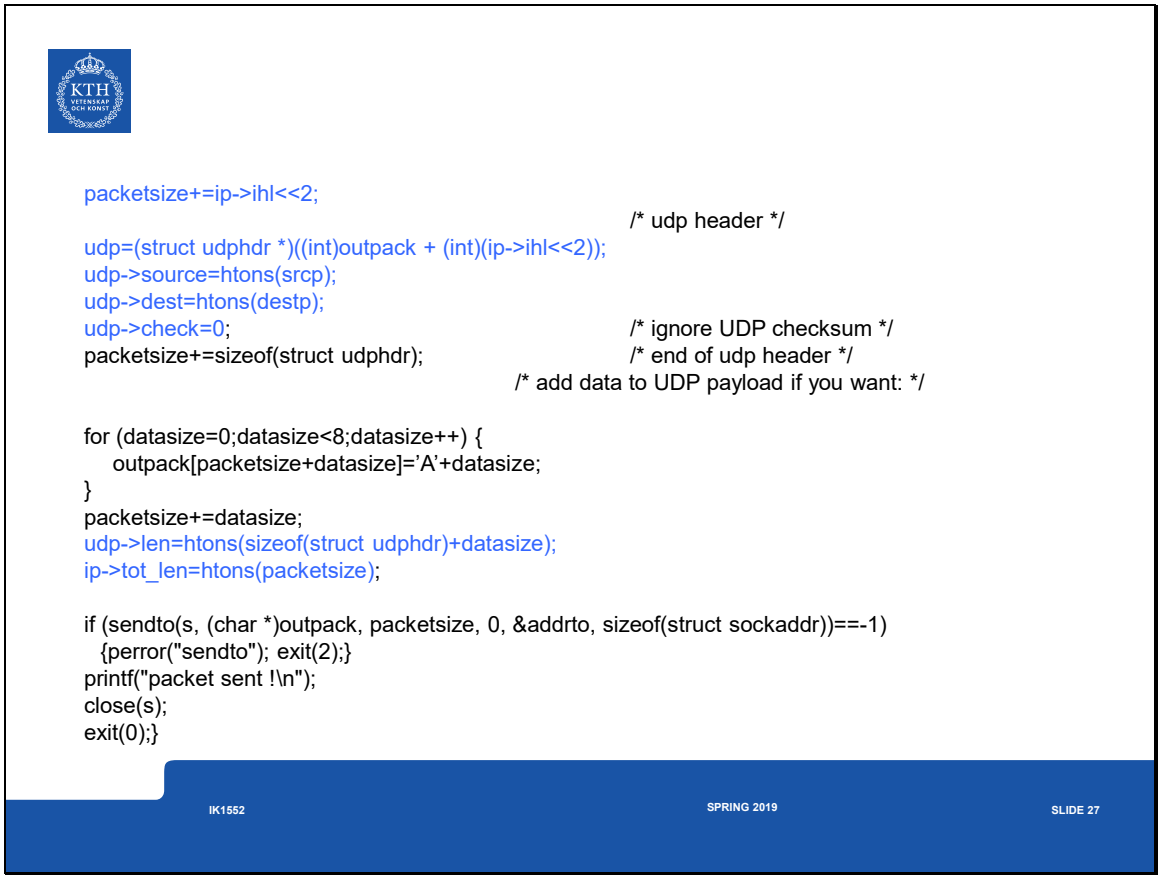
Slide 25


Building a UDP packet from scratch
/* simple example of building a UDP packet from scratch, based on the program: PingPong - 970621 by Willy TARREAU <tarreau@aemiaif.ibp.fr>
http://www.insecure.org/sploits/inetd.internal_udp_ports.DOS.attack.html
As this program uses RAW sockets, you must be root to run it */
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/udp.h>
#include <netdb.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <ctype.h>
#include <errno.h>
struct sockaddr addrfrom;
struct sockaddr addrto;
int s;
u_char outpack[65536];
struct iphdr *ip;
struct udphdr *udp;
main(int argc, char **argv) {
struct sockaddr_in *from;
struct sockaddr_in *to;
struct protoent *proto;
int i;
char *src,*dest;
int srcp, destp;
int packetsize,datasize;
if (argc!=5) {fprintf(stderr,"Usage: %s src_addr src_port dst_addr dst_port\n", argv[0]);
fprintf(stderr,"src_addr and dst_addr must be given as IP addresses (xxx.xxx.xxx.xxx)\n");
exit(2);}
src=argv[1]; srcp=atoi(argv[2]); dest=argv[3]; destp=atoi(argv[4]);
IK1552
SPRING 2019
SLIDE 25

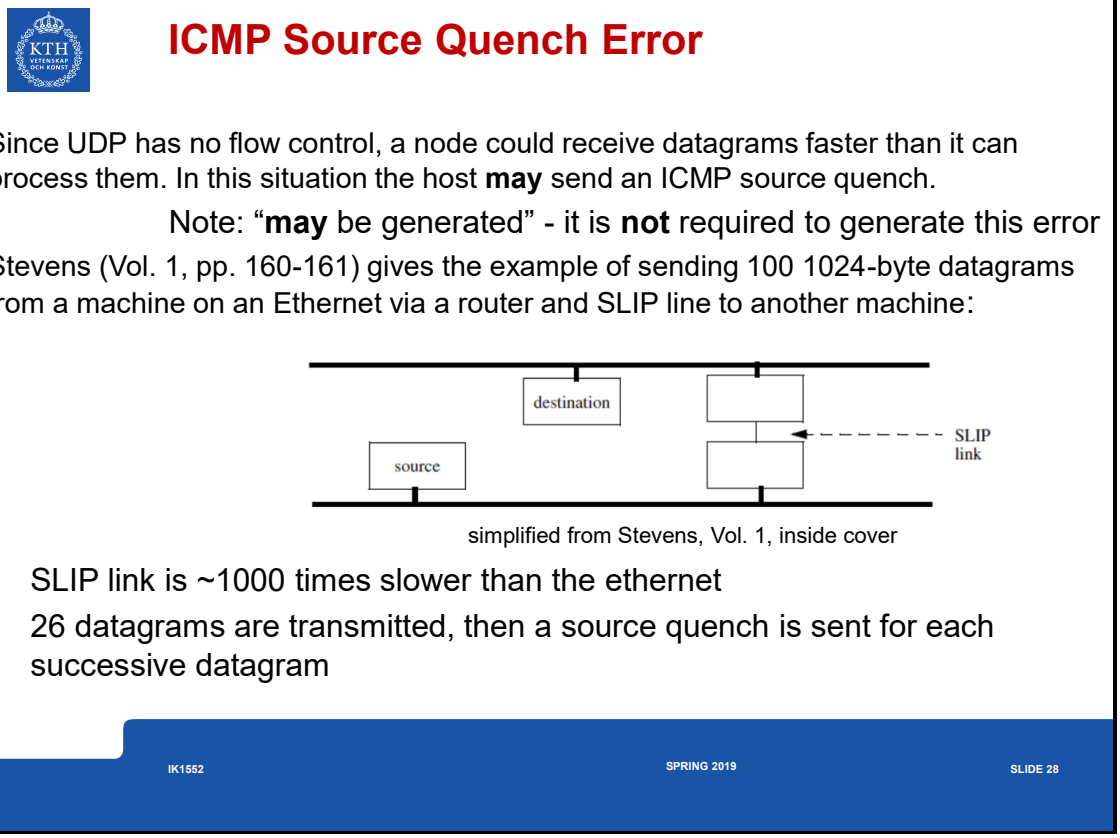
Slide 26

```
if (!(proto = getprotobyname("raw"))) {perror("getprotobyname(raw)");exit(2);}

if ((s = socket(AF_INET, SOCK_RAW, proto->p_proto)) < 0) {perror("socket");exit(2);}
memset(&addrfrom, 0, sizeof(struct sockaddr));
from = (struct sockaddr_in *)&addrfrom;
from->sin_family = AF_INET;
from->sin_port=htons(srcp);
if (!inet_aton(src, &from->sin_addr)) {fprintf(stderr,"Incorrect address for 'from': %s\n",src);exit(2); }

memset(&addrto, 0, sizeof(struct sockaddr));
to = (struct sockaddr_in *)&addrto;
to->sin_family = AF_INET;
to->sin_port=htons(destp);
f (!inet_aton(dest, &to->sin_addr)) {fprintf(stderr,"Incorrect address for 'to': %s\n",dest);exit(2); }

packetsize=0;                                   /* build a UDP packet from scratch */
ip=(struct iphdr *)outpack;
ip->version=4;                                  /* IPv4 */
ip->ihl=5;                                      /* IP header length: 5 words */
ip->tos=0;                                      /* no special type of service */
ip->id=0;                                       /* no ID */
ip->frag_off=0;                                 /* not a fragment - so there is no offset */
ip->ttl=0x40;                                   /* TTL = 64 */
If (!(proto = getprotobyname("udp"))) {perror("getprotobyname(udp)"); exit(2);}
ip->protocol=proto->p_proto;
ip->check=0;                                    /* null checksum, will be automatically computed by the kernel */
ip->saddr=from->sin_addr.s_addr;                /* set source and destination addresses */
ip->daddr=to->sin_addr.s_addr;                  /* end of ip header */
```

Slide 27



```
packetsize+=ip->ihl<<2;
                                                    /* udp header */
udp=(struct udphdr *)((int)outpack + (int)(ip->ihl<<2));
udp->source=htons(srcp);
udp->dest=htons(destp);
udp->check=0;                                       /* ignore UDP checksum */
packetsize+=sizeof(struct udphdr);                  /* end of udp header */
                                        /* add data to UDP payload if you want: */

for (datasize=0;datasize<8;datasize++) {
    outpack[packetsize+datasize]='A'+datasize;
}
packetsize+=datasize;
udp->len=htons(sizeof(struct udphdr)+datasize);
ip->tot_len=htons(packetsize);

if (sendto(s, (char *)outpack, packetsize, 0, &addrto, sizeof(struct sockaddr))==-1)
  {perror("sendto"); exit(2);}
printf("packet sent !\n");
close(s);
exit(0);}
```



ICMP Source Quench Error

Since UDP has no flow control, a node could receive datagrams faster than it can process them. In this situation the host **may** send an ICMP source quench.

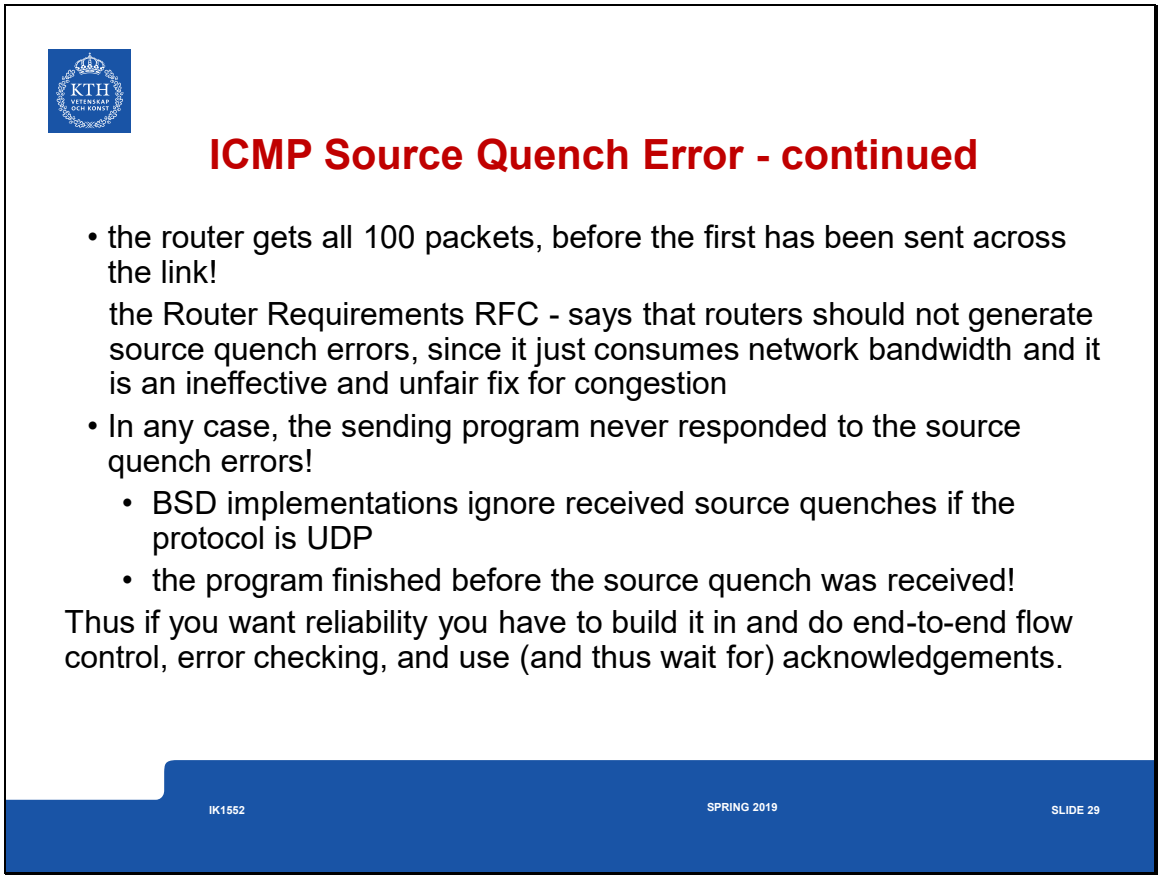
Note: "**may** be generated" - it is **not** required to generate this error

Stevens (Vol. 1, pp. 160-161) gives the example of sending 100 1024-byte datagrams from a machine on an Ethernet via a router and SLIP line to another machine:

simplified from Stevens, Vol. 1, inside cover

- SLIP link is ~1000 times slower than the ethernet
- 26 datagrams are transmitted, then a source quench is sent for each successive datagram

Slide 29



ICMP Source Quench Error - continued

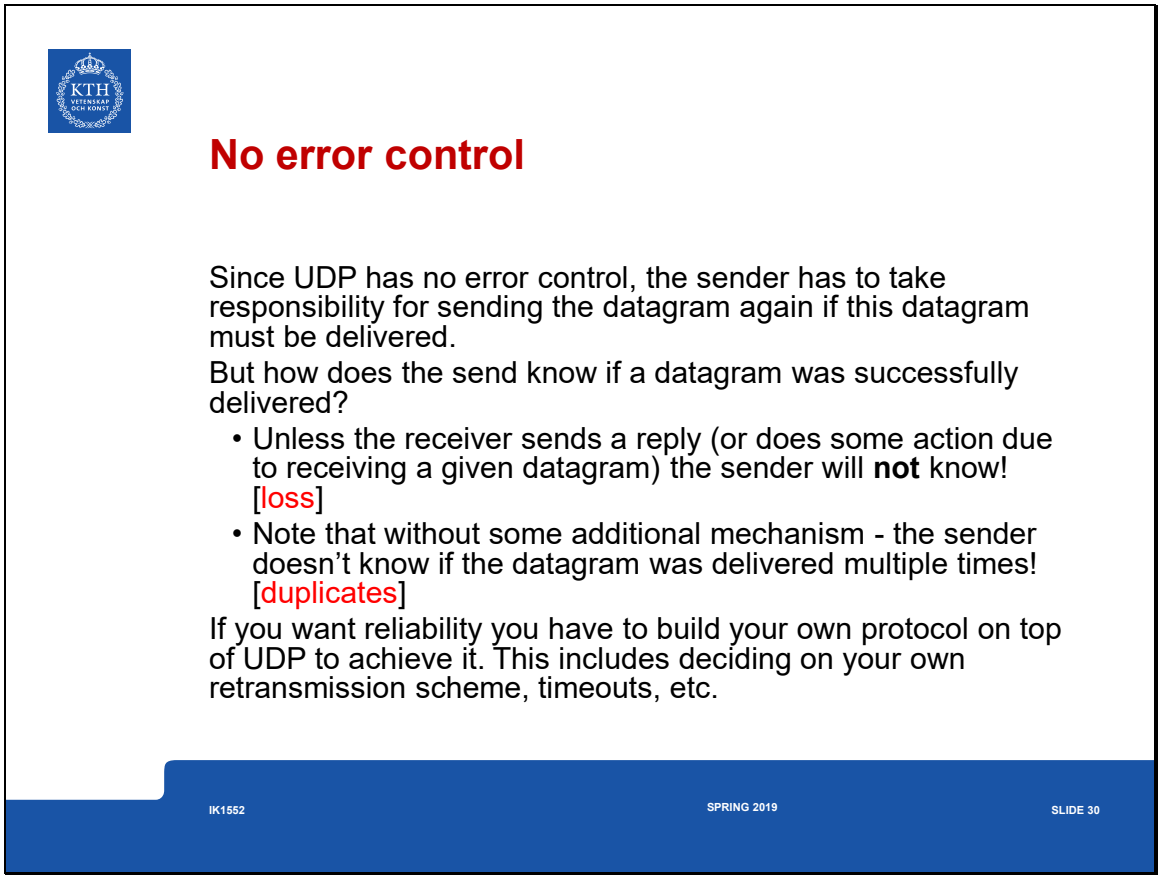
- the router gets all 100 packets, before the first has been sent across the link!

  the Router Requirements RFC - says that routers should not generate source quench errors, since it just consumes network bandwidth and it is an ineffective and unfair fix for congestion
- In any case, the sending program never responded to the source quench errors!
  - BSD implementations ignore received source quenches if the protocol is UDP
  - the program finished before the source quench was received!

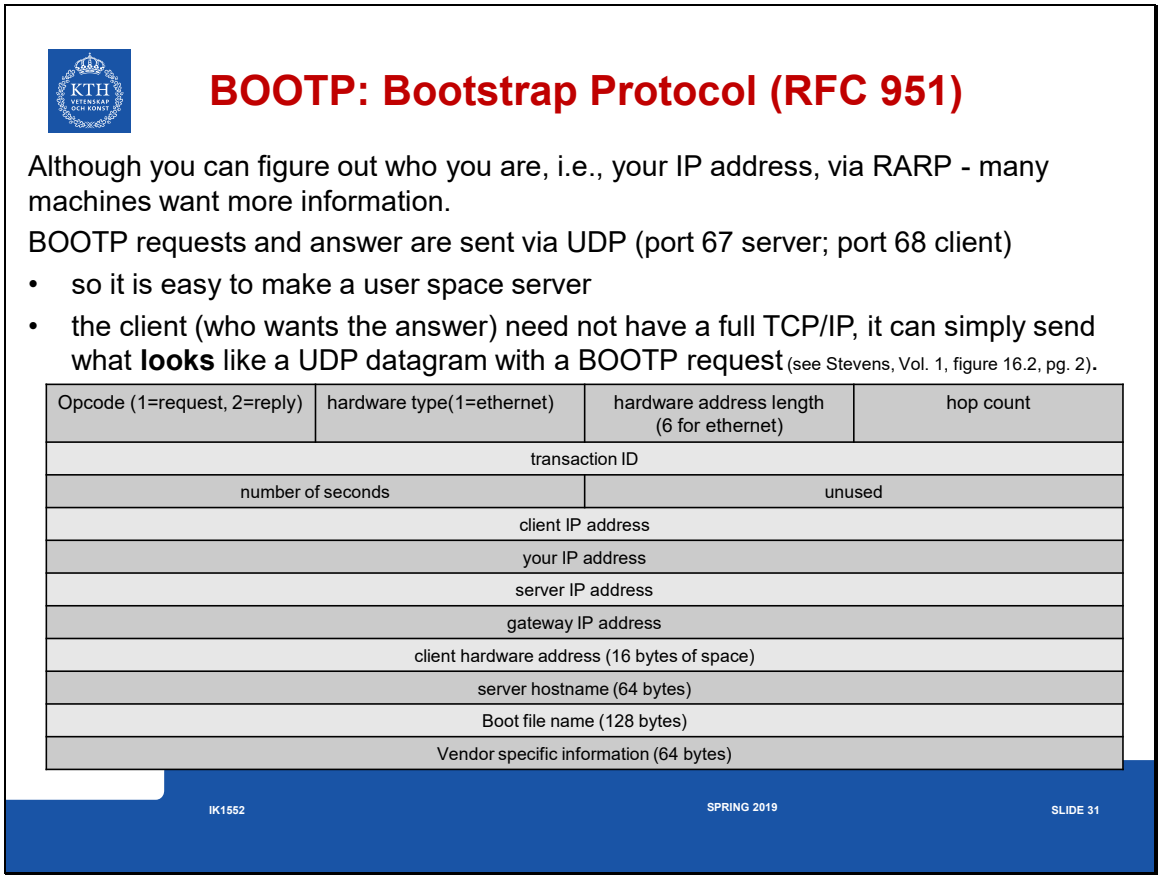
Thus if you want reliability you have to build it in and do end-to-end flow control, error checking, and use (and thus wait for) acknowledgements.

IK1552 SPRING 2019 SLIDE 29

Slide 30

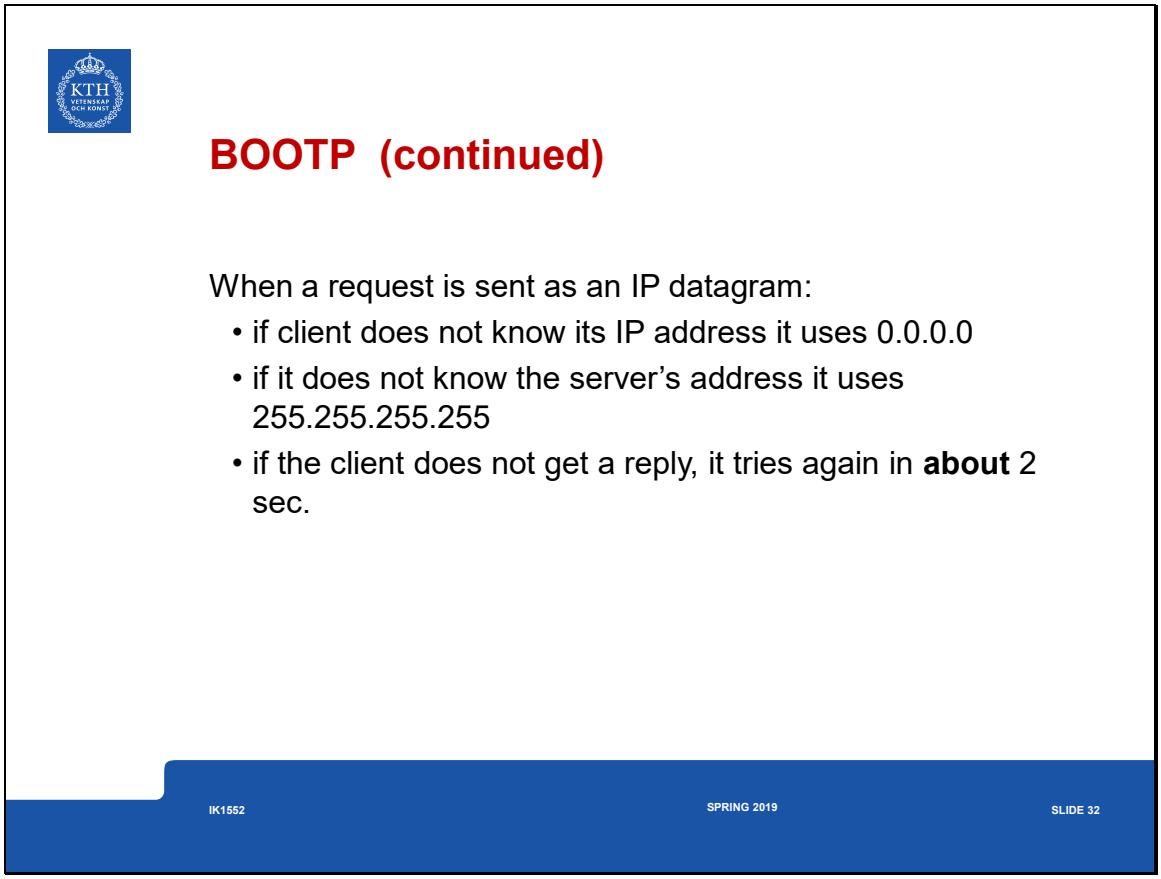

No error control
Since UDP has no error control, the sender has to take responsibility for sending the datagram again if this datagram must be delivered.
But how does the send know if a datagram was successfully delivered?
• Unless the receiver sends a reply (or does some action due to receiving a given datagram) the sender will not know! [loss]
• Note that without some additional mechanism - the sender doesn't know if the datagram was delivered multiple times! [duplicates]
If you want reliability you have to build your own protocol on top of UDP to achieve it. This includes deciding on your own retransmission scheme, timeouts, etc.
IK1552
SPRING 2019
SLIDE 30

Slide 31

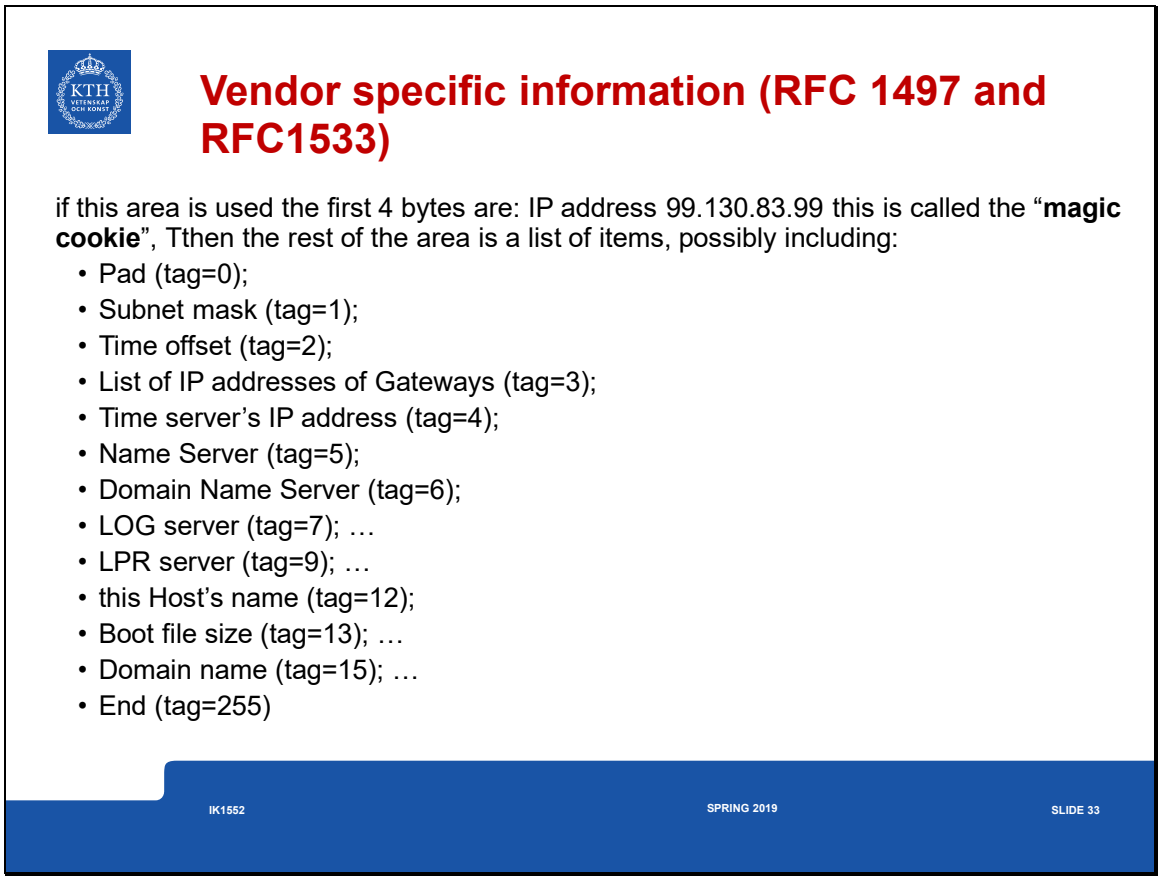


W. J. Croft and J. Gilmore, 'Bootstrap Protocol', *Internet Request for Comments*, vol. RFC 951 (Draft Standard), Sep. 1985 [Online]. Available: http://www.rfc-editor.org/rfc/rfc951.txt

Slide 32

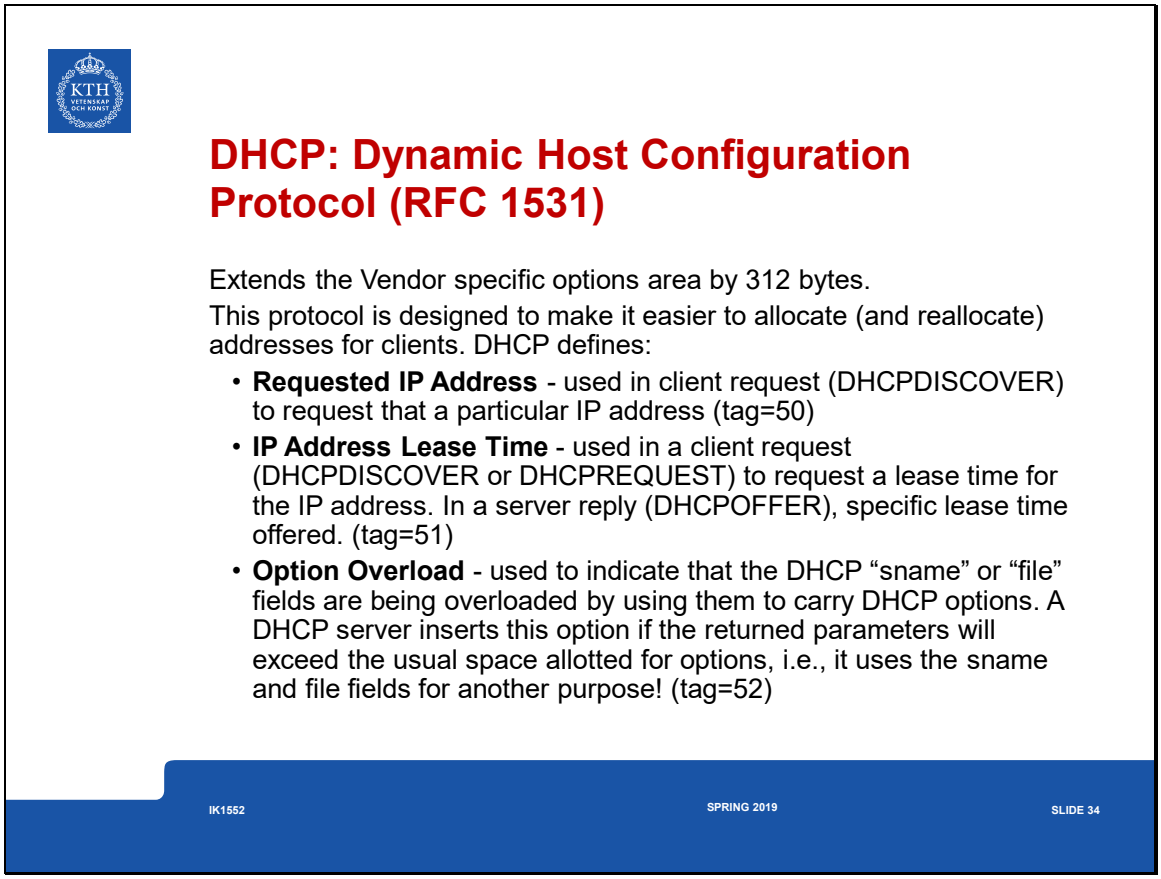

BOOTP (continued)
When a request is sent as an IP datagram:
• if client does not know its IP address it uses 0.0.0.0
• if it does not know the server's address it uses 255.255.255.255
• if the client does not get a reply, it tries again in about 2 sec.
IK1552
SPRING 2019
SLIDE 32

Slide 33



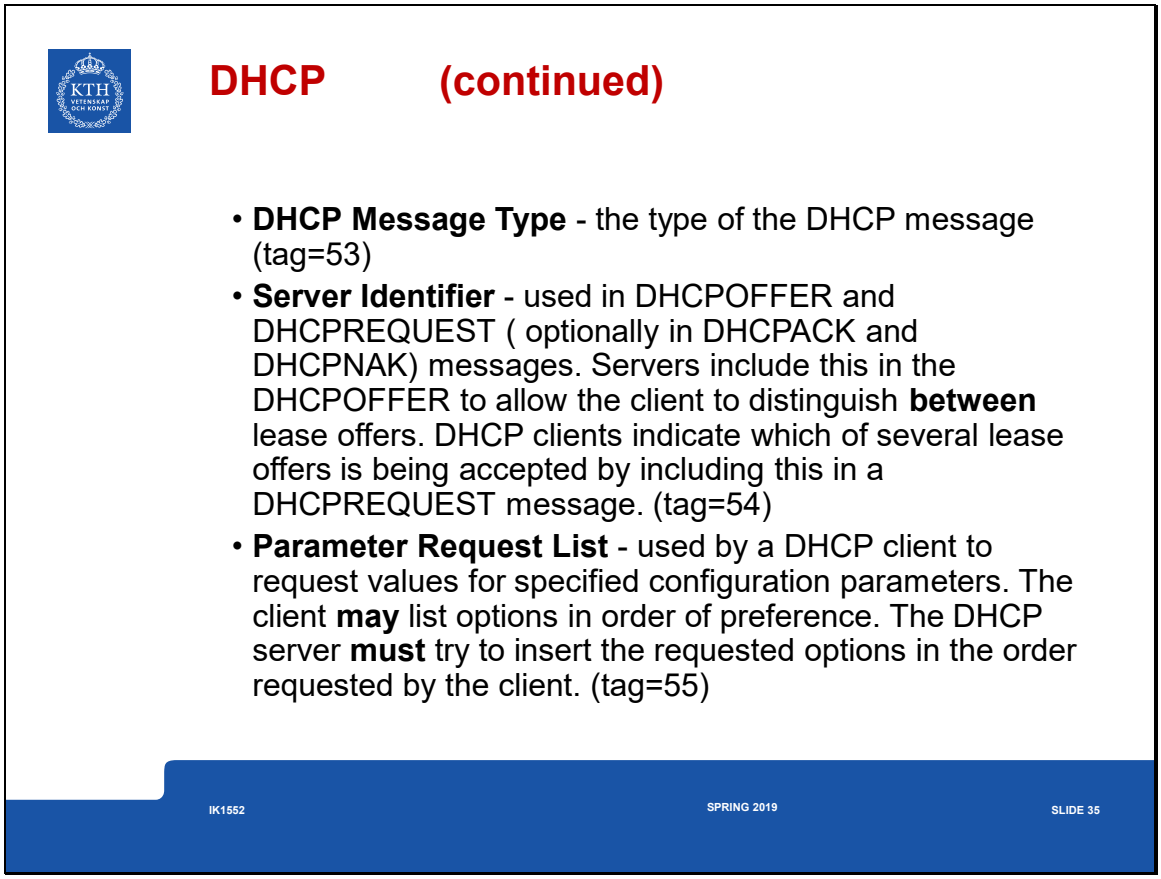
W. J. Croft and J. Gilmore, 'Bootstrap Protocol', *Internet Request for Comments*, vol. RFC 951 (Draft Standard), Sep. 1985 [Online]. Available: http://www.rfc-editor.org/rfc/rfc951.txt
P. A. Prindeville, 'BOOTP vendor information extensions', *Internet Request for Comments*, vol. RFC 1048, Feb. 1988 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1048.txt
J. K. Reynolds, 'BOOTP vendor information extensions', *Internet Request for Comments*, vol. RFC 1084, Dec. 1988 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1084.txt
J. Reynolds, 'BOOTP Vendor Information Extensions', *Internet Request for Comments*, vol. RFC 1395 (Draft Standard), Jan. 1993 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1395.txt
J. Reynolds, 'BOOTP Vendor Information Extensions', *Internet Request for Comments*, vol. RFC 1497 (Draft Standard), Aug. 1993 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1497.txt
S. Alexander and R. Droms, 'DHCP Options and BOOTP Vendor Extensions', *Internet Request for Comments*, vol. RFC 1533 (Proposed Standard), Oct. 1993 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1533.txt

Slide 34

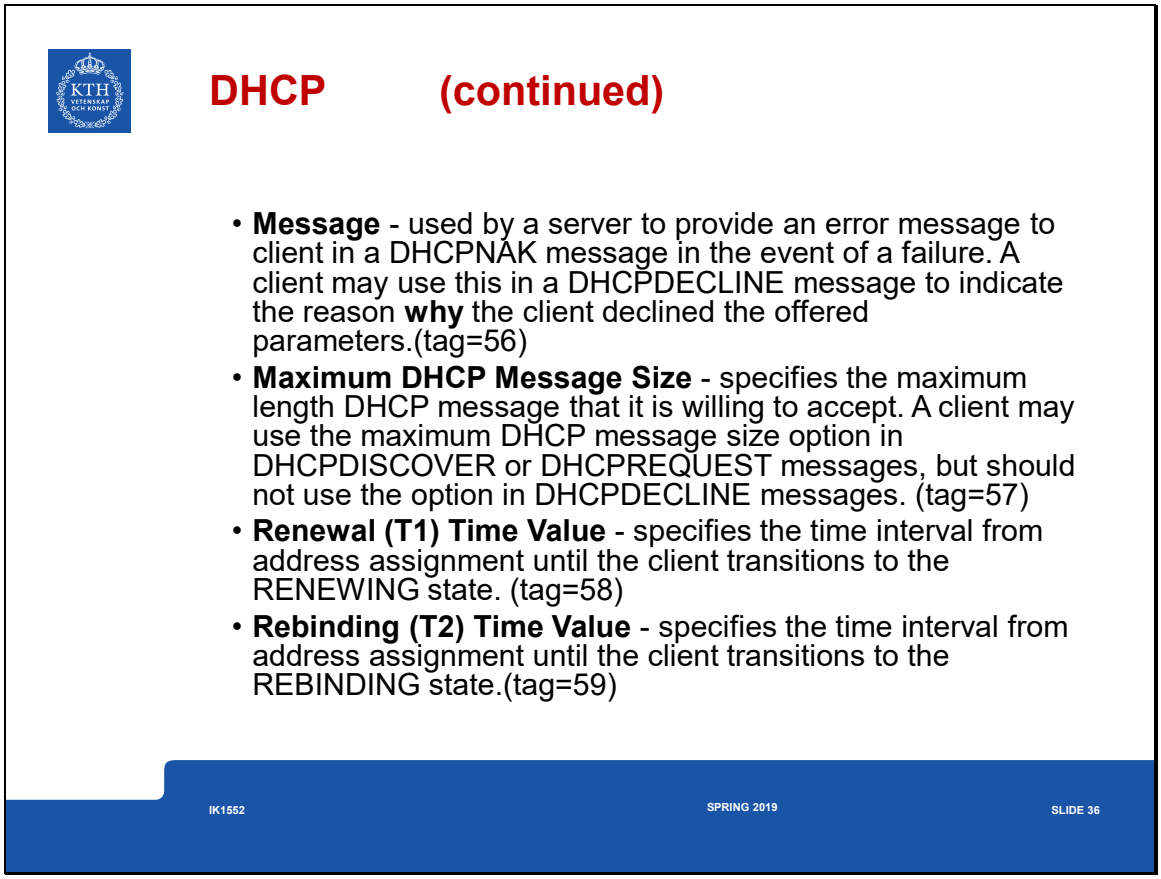


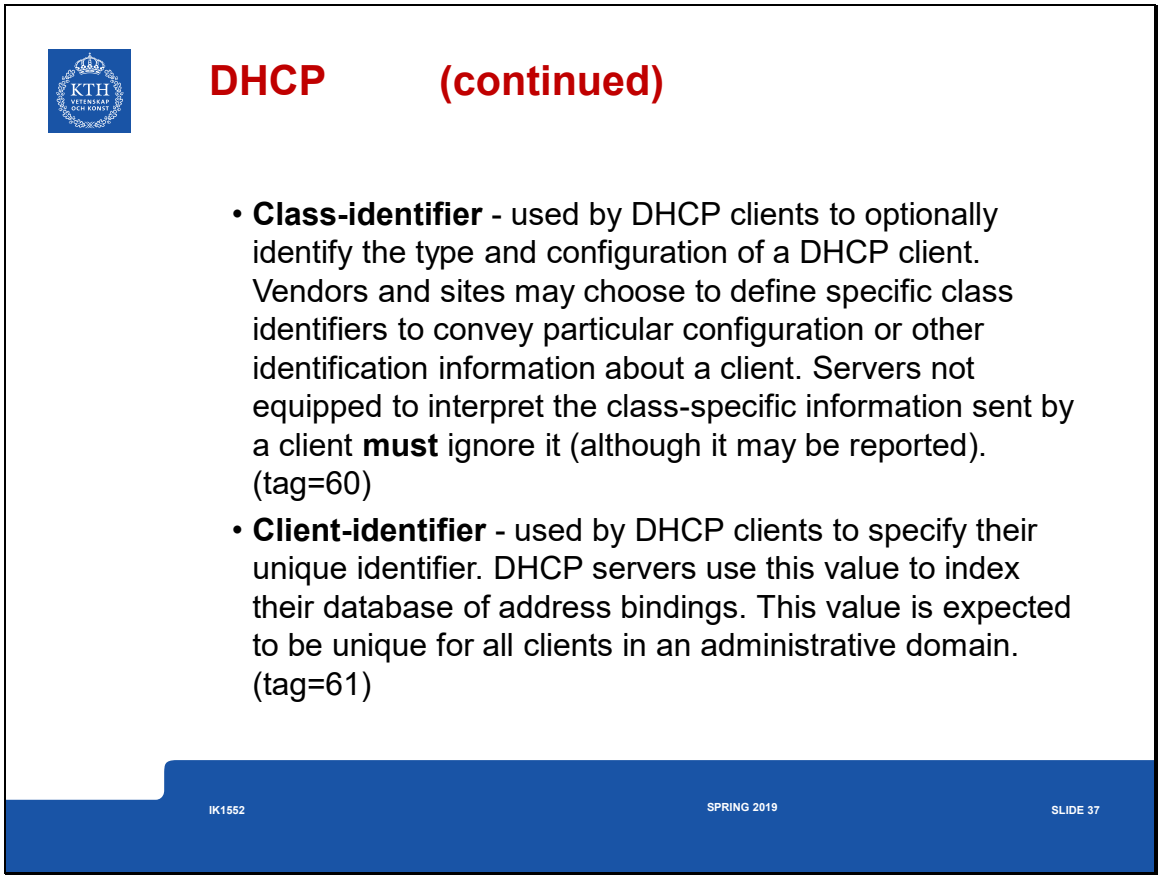
R. Droms, 'Dynamic Host Configuration Protocol', *Internet Request for Comments*, vol. RFC 1531 (Proposed Standard), Oct. 1993 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1531.txt

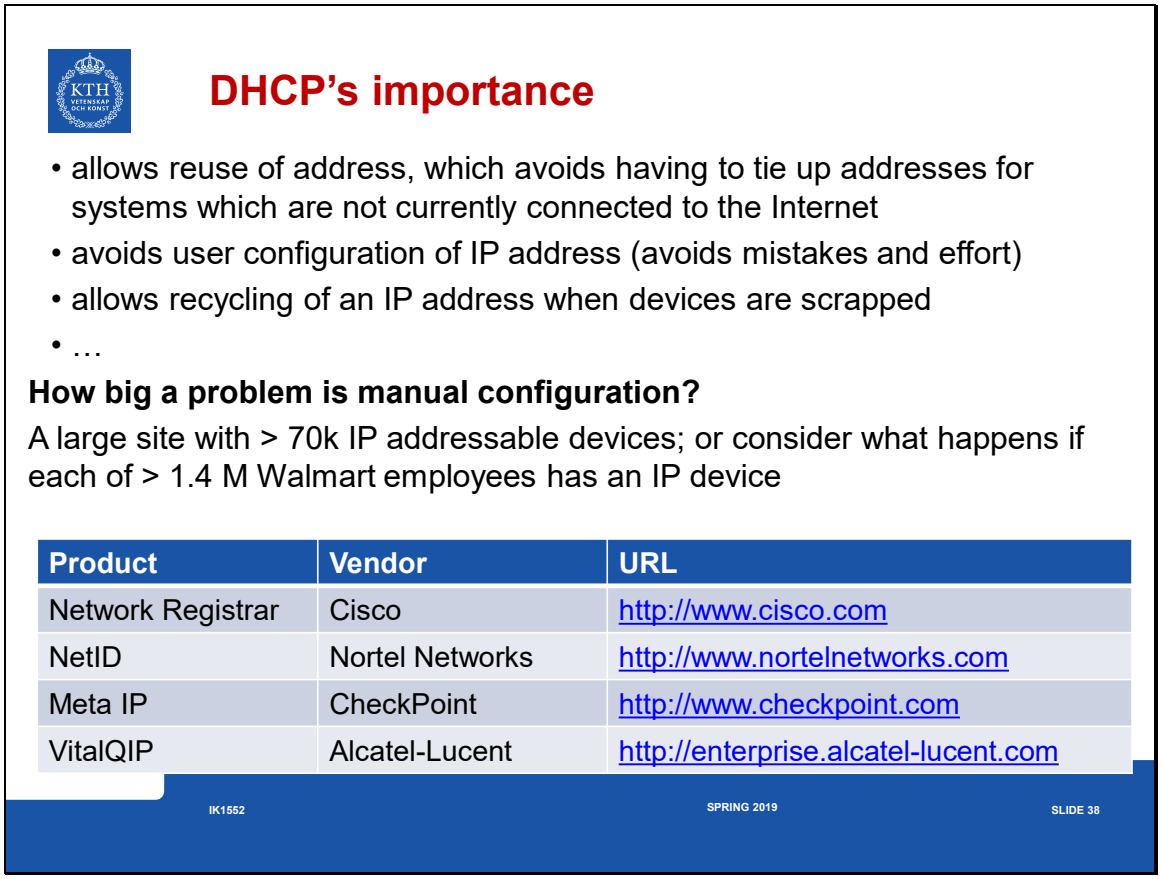
Slide 35


DHCP (continued)
• DHCP Message Type - the type of the DHCP message (tag=53)
• Server Identifier - used in DHCPOFFER and DHCPREQUEST (optionally in DHCPACK and DHCPNAK) messages. Servers include this in the DHCPOFFER to allow the client to distinguish between lease offers. DHCP clients indicate which of several lease offers is being accepted by including this in a DHCPREQUEST message. (tag=54)
• Parameter Request List - used by a DHCP client to request values for specified configuration parameters. The client may list options in order of preference. The DHCP server must try to insert the requested options in the order requested by the client. (tag=55)
IK1552
SPRING 2019
SLIDE 35

Slide 36


DHCP (continued)
• Message - used by a server to provide an error message to client in a DHCPNAK message in the event of a failure. A client may use this in a DHCPDECLINE message to indicate the reason why the client declined the offered parameters.(tag=56)
• Maximum DHCP Message Size - specifies the maximum length DHCP message that it is willing to accept. A client may use the maximum DHCP message size option in DHCPDISCOVER or DHCPREQUEST messages, but should not use the option in DHCPDECLINE messages. (tag=57)
• Renewal (T1) Time Value - specifies the time interval from address assignment until the client transitions to the RENEWING state. (tag=58)
• Rebinding (T2) Time Value - specifies the time interval from address assignment until the client transitions to the REBINDING state.(tag=59)
IK1552
SPRING 2019
SLIDE 36


DHCP (continued)
• Class-identifier - used by DHCP clients to optionally identify the type and configuration of a DHCP client. Vendors and sites may choose to define specific class identifiers to convey particular configuration or other identification information about a client. Servers not equipped to interpret the class-specific information sent by a client must ignore it (although it may be reported). (tag=60)
• Client-identifier - used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain. (tag=61)
IK1552
SPRING 2019
SLIDE 37



DHCP's importance

- allows reuse of address, which avoids having to tie up addresses for systems which are not currently connected to the Internet
- avoids user configuration of IP address (avoids mistakes and effort)
- allows recycling of an IP address when devices are scrapped
- …

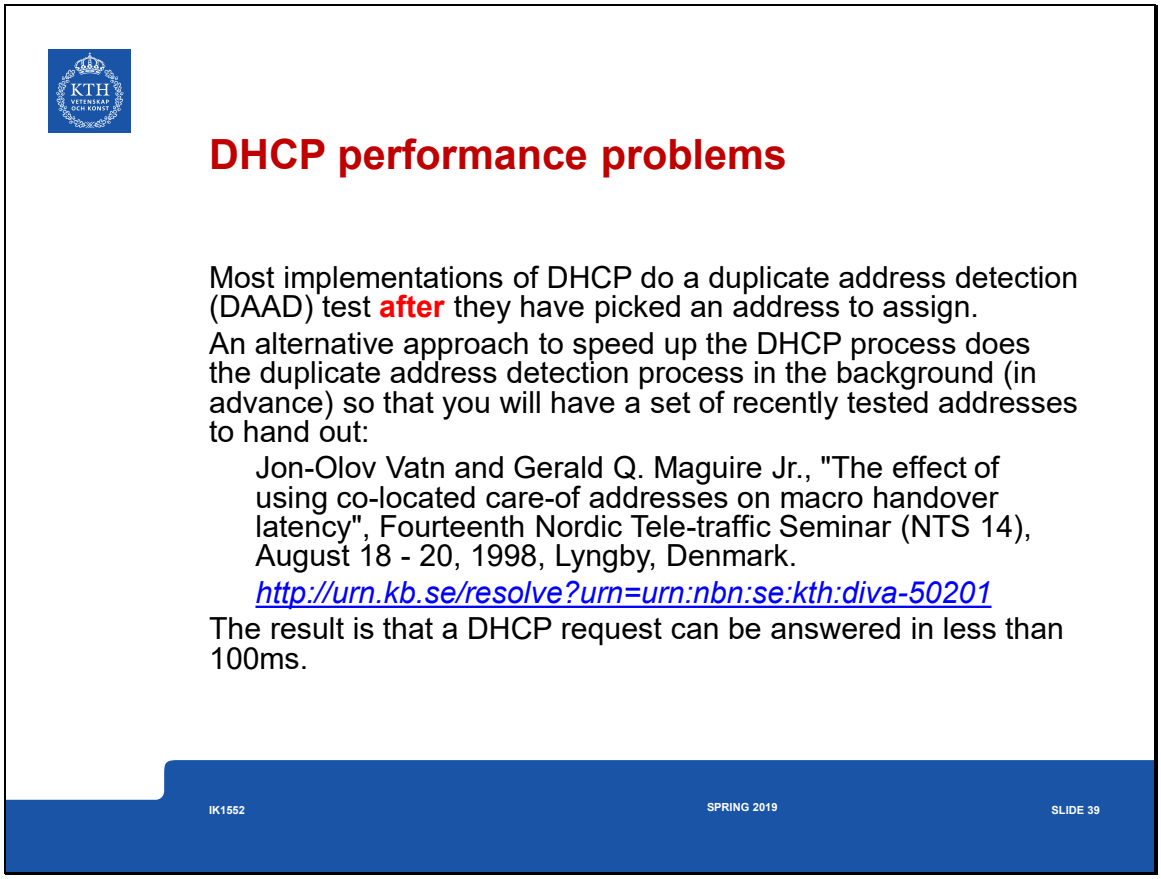
How big a problem is manual configuration?

A large site with > 70k IP addressable devices; or consider what happens if each of > 1.4 M Walmart employees has an IP device

Product	Vendor	URL
Network Registrar	Cisco	http://www.cisco.com
NetID	Nortel Networks	http://www.nortelnetworks.com
Meta IP	CheckPoint	http://www.checkpoint.com
VitalQIP	Alcatel-Lucent	http://enterprise.alcatel-lucent.com

IK1552 SPRING 2019 SLIDE 38

Slide 39



DHCP performance problems

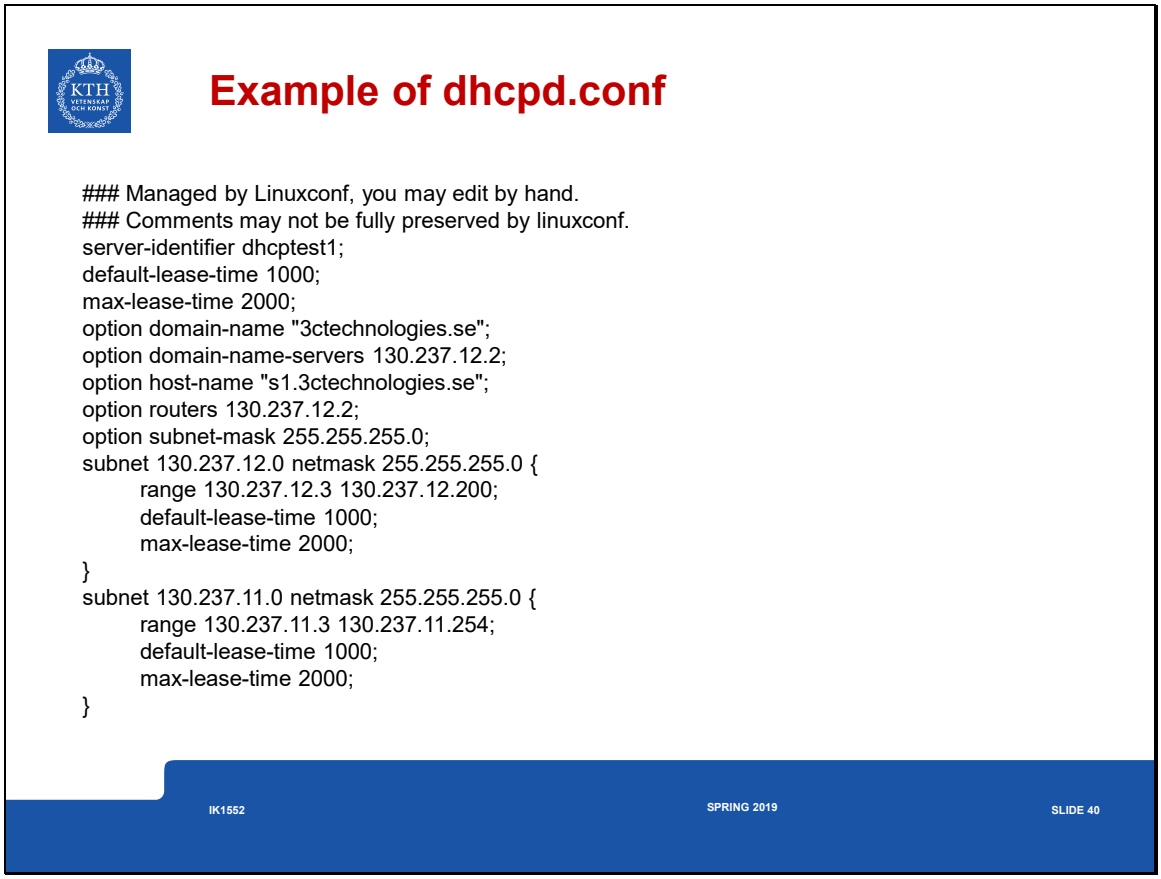
Most implementations of DHCP do a duplicate address detection (DAAD) test **after** they have picked an address to assign.

An alternative approach to speed up the DHCP process does the duplicate address detection process in the background (in advance) so that you will have a set of recently tested addresses to hand out:

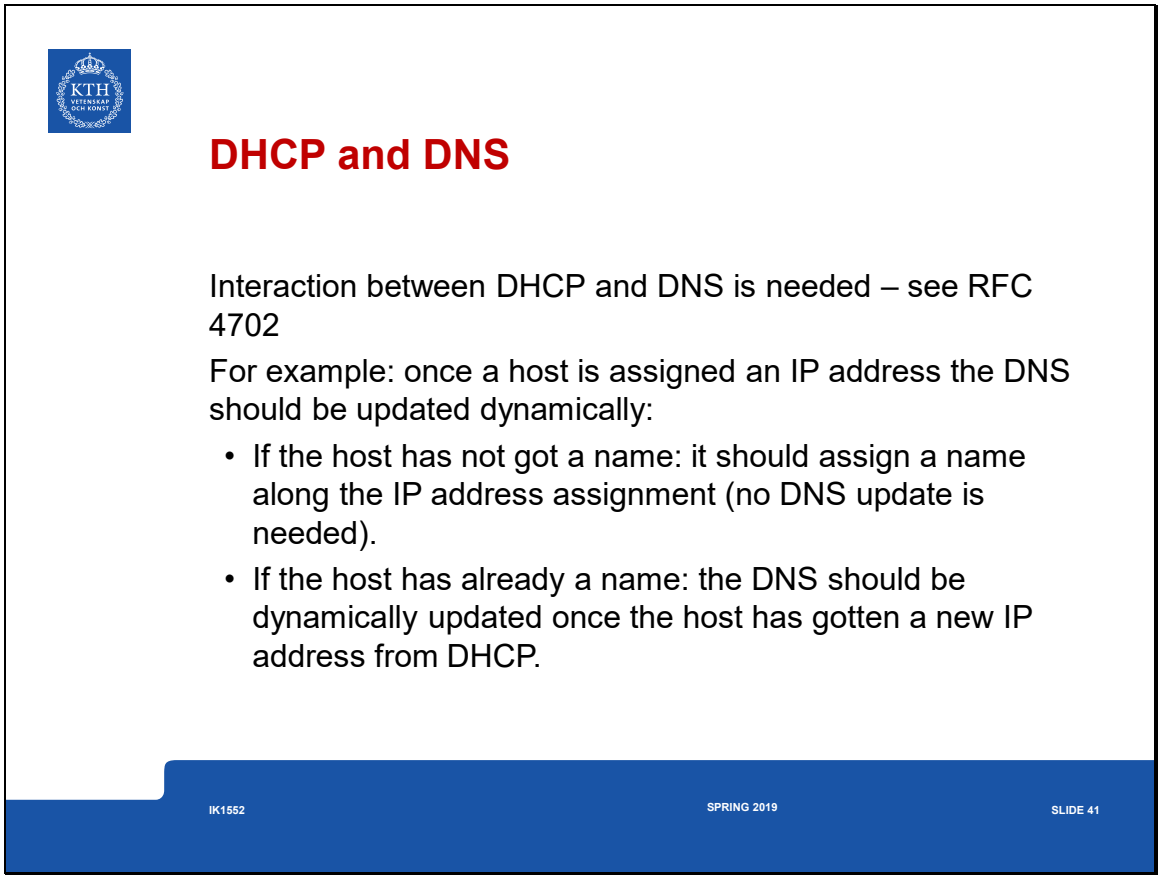
> Jon-Olov Vatn and Gerald Q. Maguire Jr., "The effect of using co-located care-of addresses on macro handover latency", Fourteenth Nordic Tele-traffic Seminar (NTS 14), August 18 - 20, 1998, Lyngby, Denmark.
>
> *http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-50201*

The result is that a DHCP request can be answered in less than 100ms.

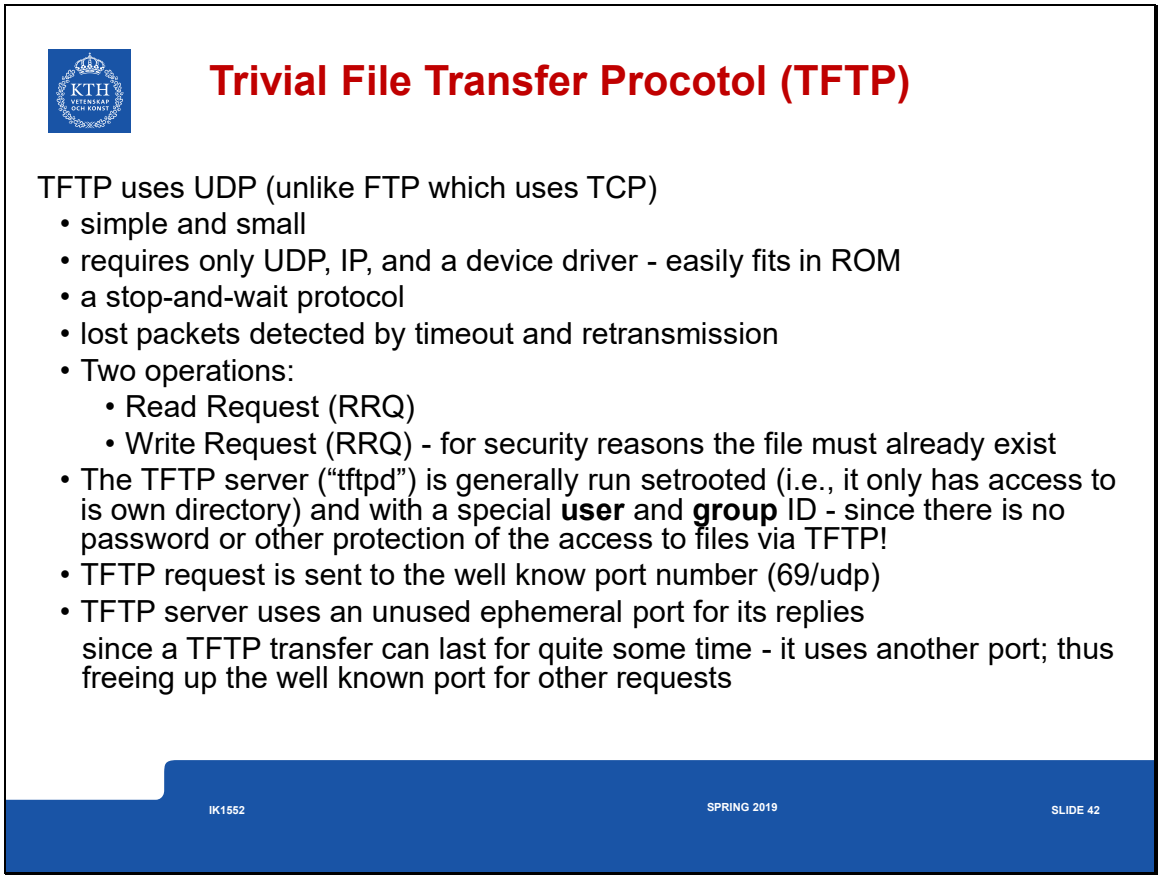
Slide 40


Example of dhcpd.conf
Managed by Linuxconf, you may edit by hand.
Comments may not be fully preserved by linuxconf.
server-identifier dhcptest1;
default-lease-time 1000;
max-lease-time 2000;
option domain-name "3ctechnologies.se";
option domain-name-servers 130.237.12.2;
option host-name "s1.3ctechnologies.se";
option routers 130.237.12.2;
option subnet-mask 255.255.255.0;
subnet 130.237.12.0 netmask 255.255.255.0 {
range 130.237.12.3 130.237.12.200;
default-lease-time 1000;
max-lease-time 2000;
}
subnet 130.237.11.0 netmask 255.255.255.0 {
range 130.237.11.3 130.237.11.254;
default-lease-time 1000;
max-lease-time 2000;
}
IK1552
SPRING 2019
SLIDE 40

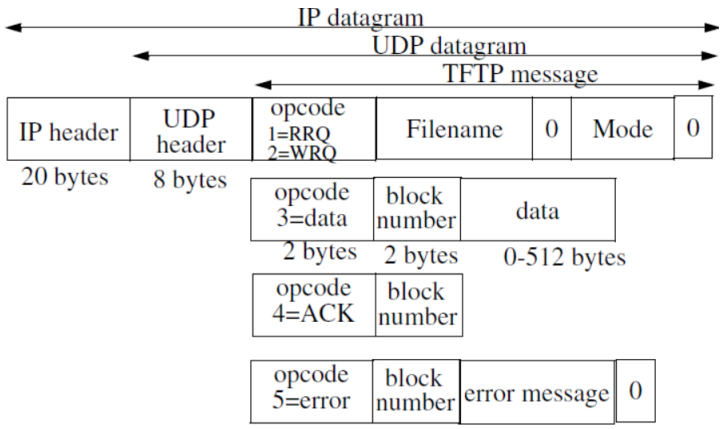
Slide 41



[RFC 4702] M. Stapp, B. Volz, and Y. Rekhter, 'The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option', *Internet Request for Comments*, vol. RFC 4702 (Proposed Standard), Oct. 2006 [Online]. Available: http://www.rfc-editor.org/rfc/rfc4702.txt


Trivial File Transfer Procotol (TFTP)
TFTP uses UDP (unlike FTP which uses TCP)
• simple and small
• requires only UDP, IP, and a device driver - easily fits in ROM
• a stop-and-wait protocol
• lost packets detected by timeout and retransmission
• Two operations:
• Read Request (RRQ)
• Write Request (RRQ) - for security reasons the file must already exist
• The TFTP server (“tftpd”) is generally run setrooted (i.e., it only has access to is own directory) and with a special user and group ID - since there is no password or other protection of the access to files via TFTP!
• TFTP request is sent to the well know port number (69/udp)
• TFTP server uses an unused ephemeral port for its replies
since a TFTP transfer can last for quite some time - it uses another port; thus freeing up the well known port for other requests
IK1552
SPRING 2019
SLIDE 42

TFTP messaages

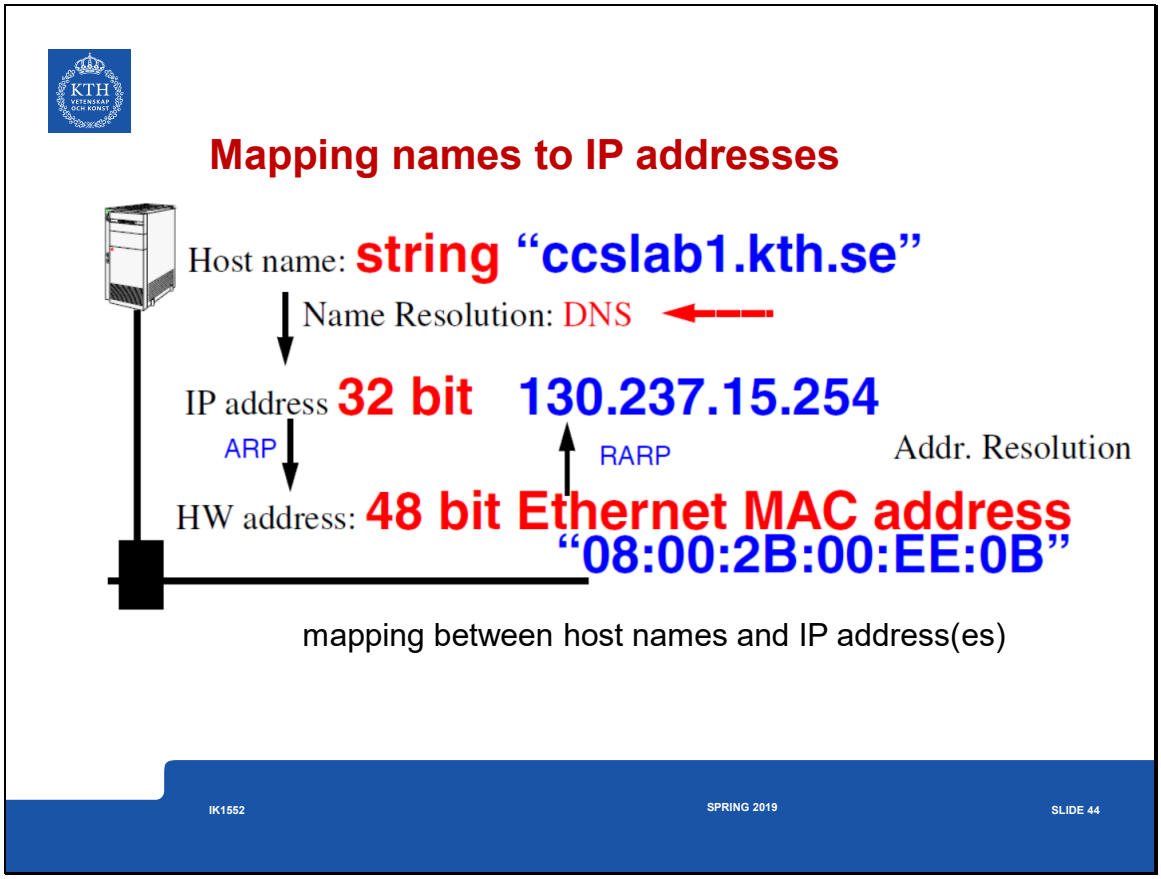


TFTP messages (see Stevens, Vol. 1, figure 15.1, pg. 210)

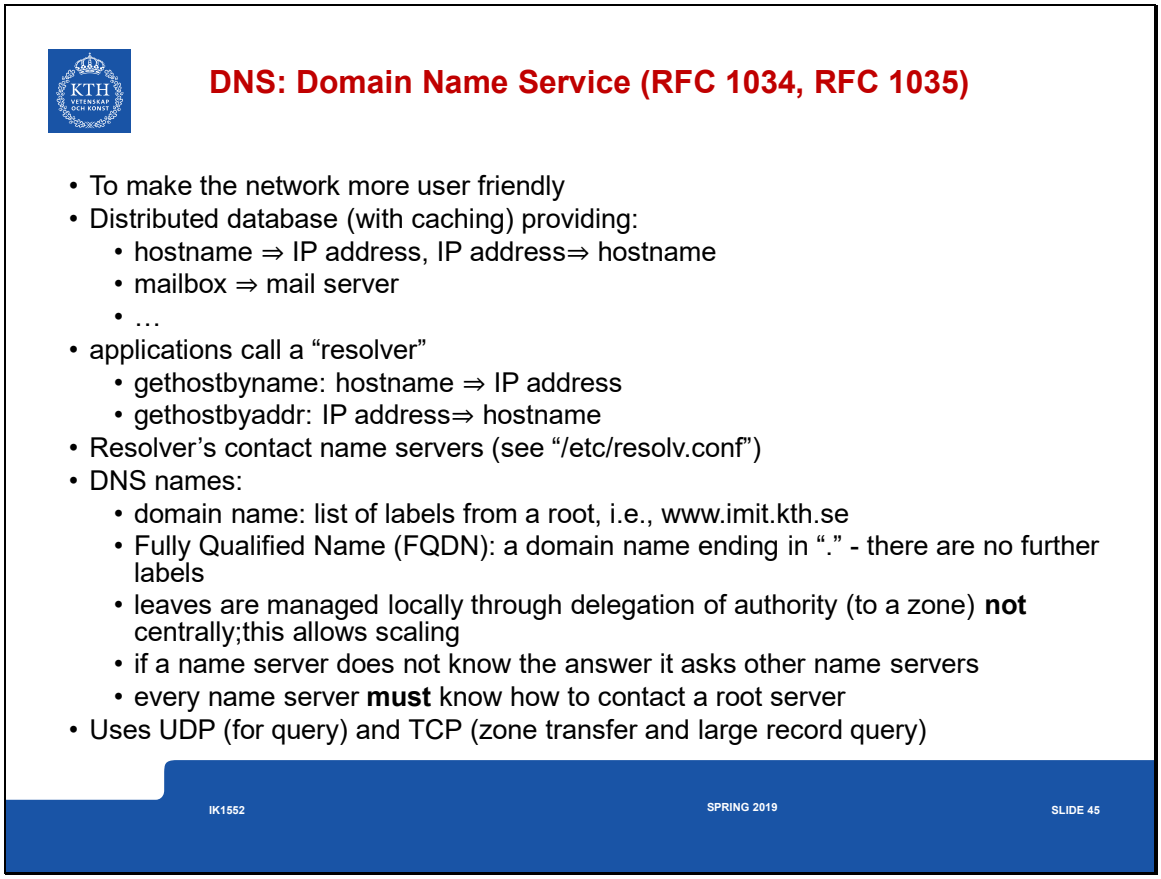
Filename and Mode (“netascii” or “octet”) are both N bytes sequences terminated by a null byte.

Widely used for bootstrapping diskless systems (such as X terminals) and for dumping the configuration of routers (this is where the write request is used)

Slide 44


Mapping names to IP addresses
Host name: string "ccslab1.kth.se"
Name Resolution: DNS
IP address 32 bit 130.237.15.254
ARP
RARP
Addr. Resolution
HW address: 48 bit Ethernet MAC address
"08:00:2B:00:EE:0B"
mapping between host names and IP address(es)
IK1552
SPRING 2019
SLIDE 44

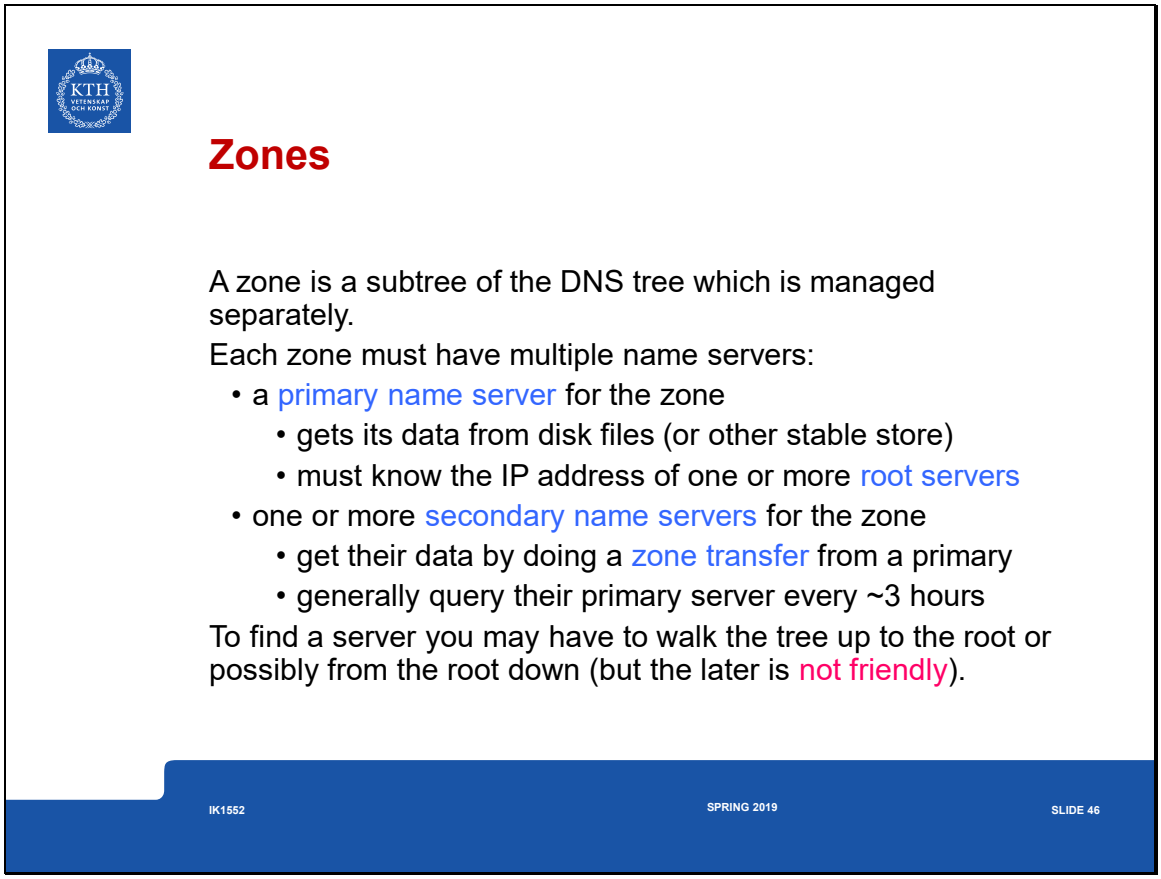
Slide 45



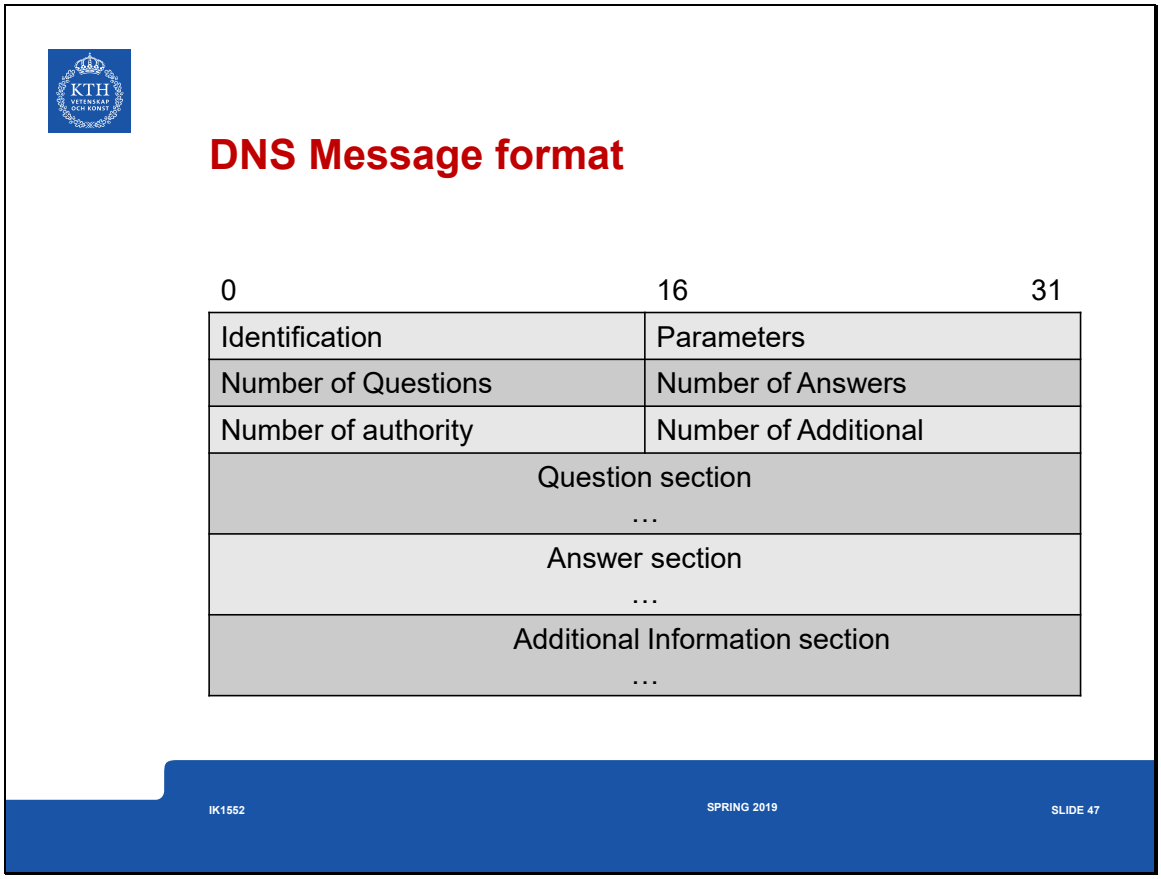
P. V. Mockapetris, ‘Domain names - concepts and facilities’, *Internet Request for Comments*, vol. RFC 1034 (INTERNET STANDARD), Nov. 1987 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1034.txt

P. V. Mockapetris, ‘Domain names - implementation and specification’, *Internet Request for Comments*, vol. RFC 1035 (INTERNET STANDARD), Nov. 1987 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1035.txt

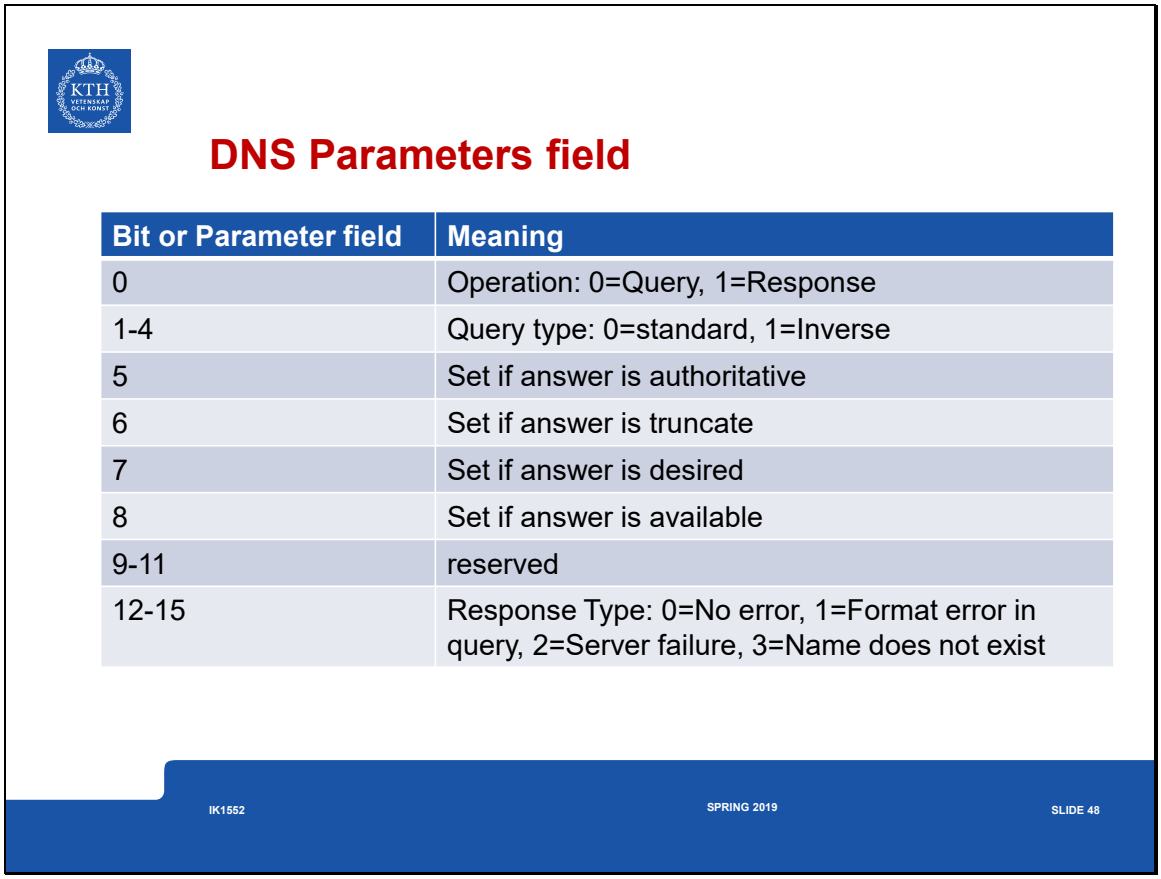
Slide 46


Zones
A zone is a subtree of the DNS tree which is managed separately.
Each zone must have multiple name servers:
• a primary name server for the zone
• gets its data from disk files (or other stable store)
• must know the IP address of one or more root servers
• one or more secondary name servers for the zone
• get their data by doing a zone transfer from a primary
• generally query their primary server every ~3 hours
To find a server you may have to walk the tree up to the root or possibly from the root down (but the later is not friendly).
IK1552
SPRING 2019
SLIDE 46

Slide 47


DNS Message format
0
16
31
Identification
Parameters
Number of Questions
Number of Answers
Number of authority
Number of Additional
Question section
…
Answer section
…
Additional Information section
…
IK1552
SPRING 2019
SLIDE 47

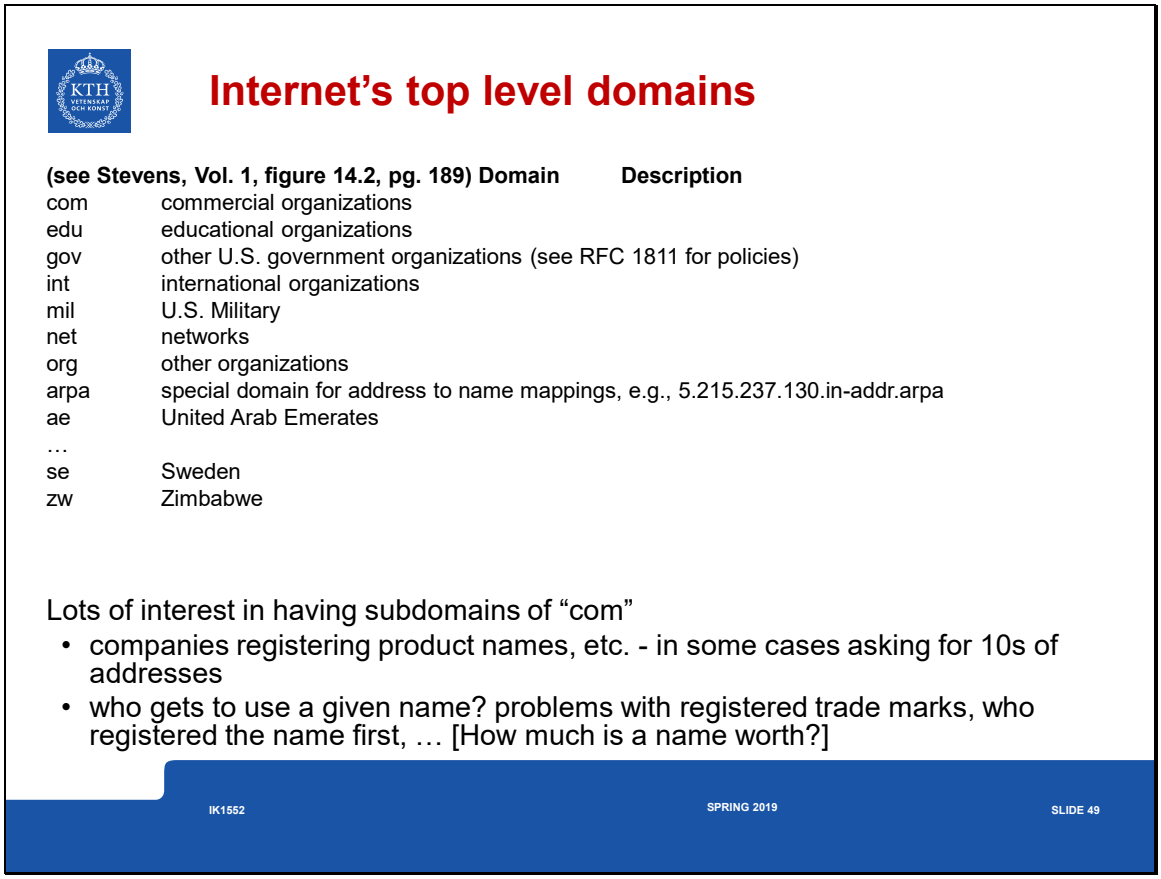
Slide 48



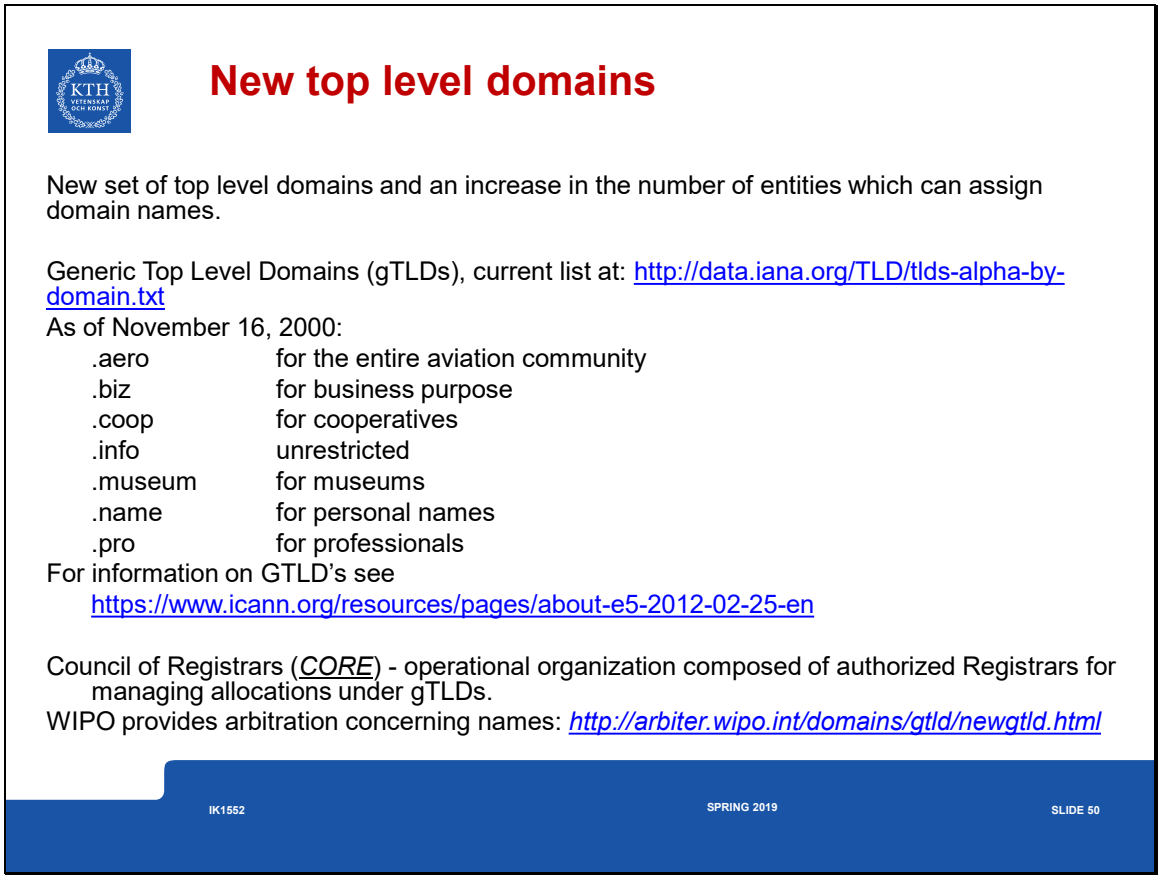
DNS Parameters field

Bit or Parameter field	Meaning
0	Operation: 0=Query, 1=Response
1-4	Query type: 0=standard, 1=Inverse
5	Set if answer is authoritative
6	Set if answer is truncate
7	Set if answer is desired
8	Set if answer is available
9-11	reserved
12-15	Response Type: 0=No error, 1=Format error in query, 2=Server failure, 3=Name does not exist

Slide 49


Internet's top level domains
(see Stevens, Vol. 1, figure 14.2, pg. 189) Domain Description
com commercial organizations
edu educational organizations
gov other U.S. government organizations (see RFC 1811 for policies)
int international organizations
mil U.S. Military
net networks
org other organizations
arpa special domain for address to name mappings, e.g., 5.215.237.130.in-addr.arpa
ae United Arab Emerates
…
se Sweden
zw Zimbabwe
Lots of interest in having subdomains of "com"
• companies registering product names, etc. - in some cases asking for 10s of addresses
• who gets to use a given name? problems with registered trade marks, who registered the name first, … [How much is a name worth?]
IK1552
SPRING 2019
SLIDE 49

Slide 50



New top level domains

New set of top level domains and an increase in the number of entities which can assign domain names.

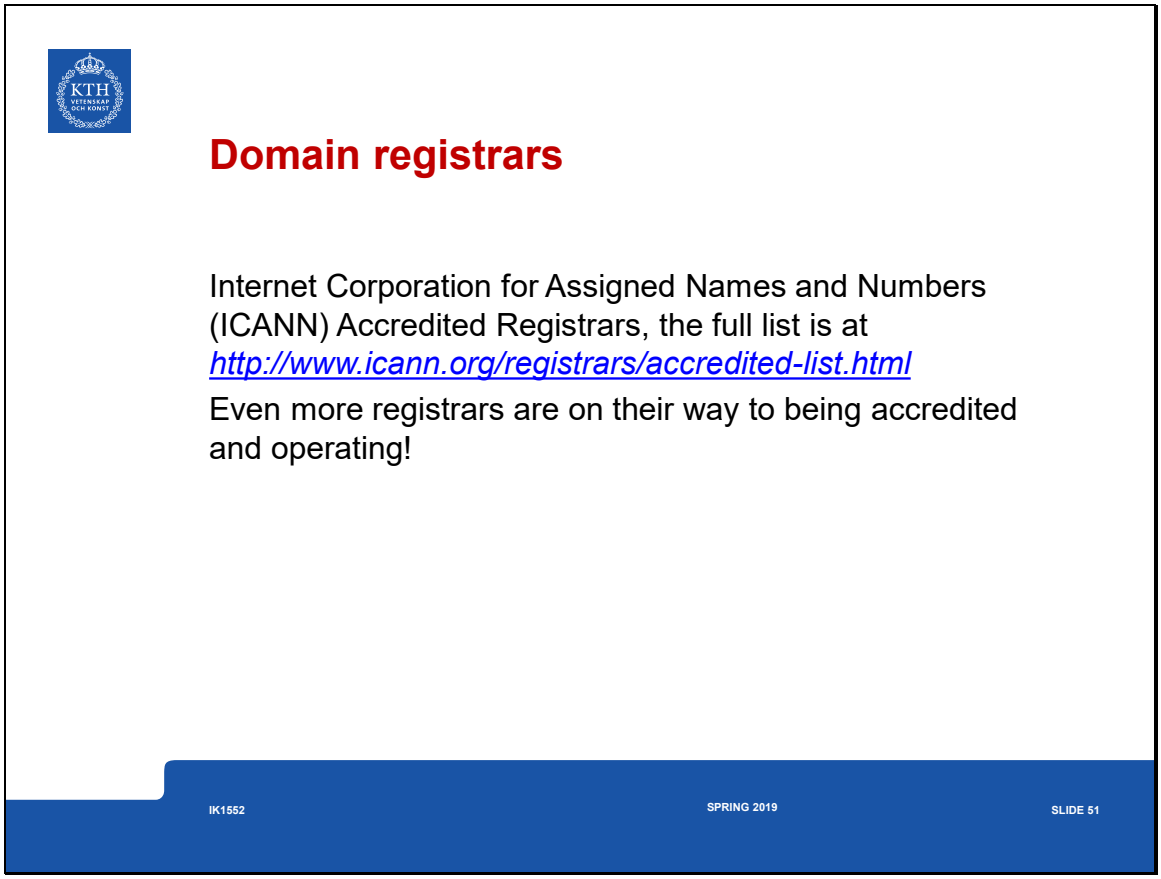
Generic Top Level Domains (gTLDs), current list at: http://data.iana.org/TLD/tlds-alpha-by-domain.txt
As of November 16, 2000:

.aero	for the entire aviation community
.biz	for business purpose
.coop	for cooperatives
.info	unrestricted
.museum	for museums
.name	for personal names
.pro	for professionals

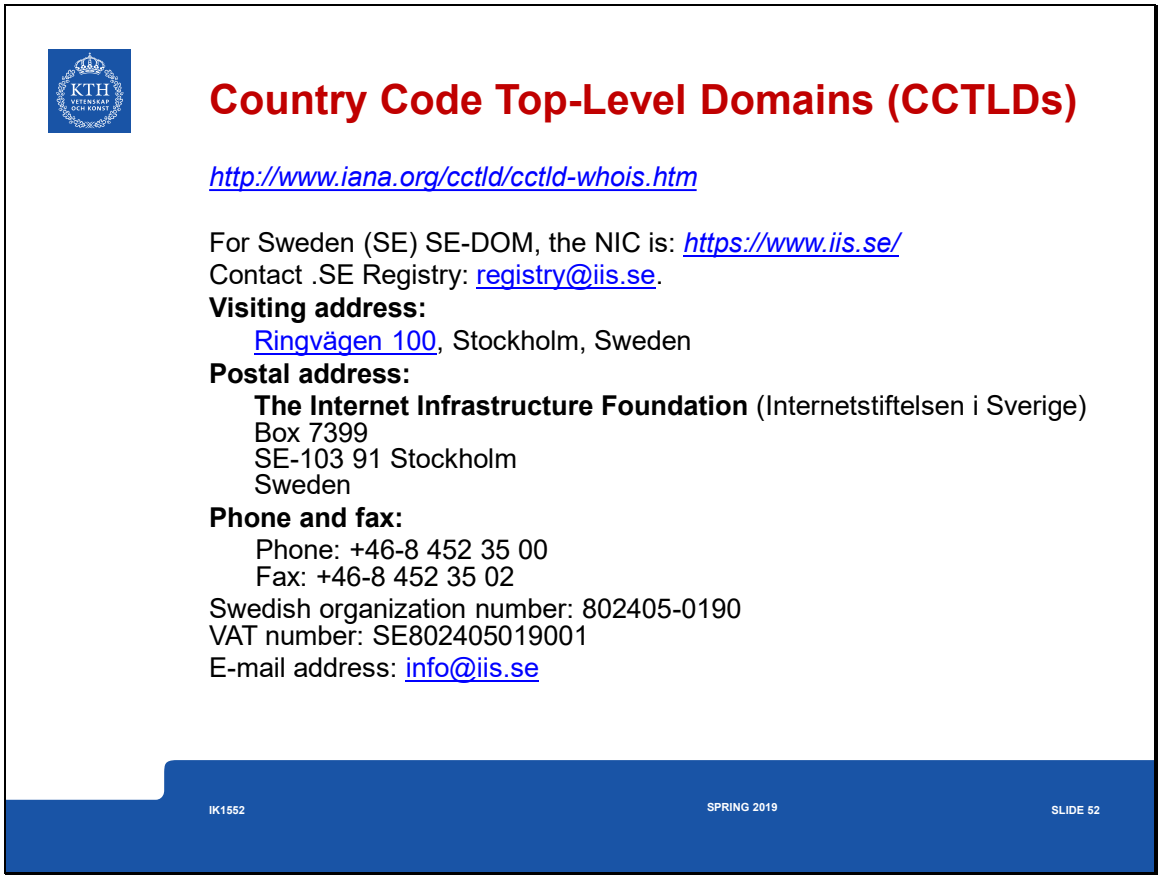
For information on GTLD's see
https://www.icann.org/resources/pages/about-e5-2012-02-25-en

Council of Registrars (*CORE*) - operational organization composed of authorized Registrars for managing allocations under gTLDs.
WIPO provides arbitration concerning names: *http://arbiter.wipo.int/domains/gtld/newgtld.html*

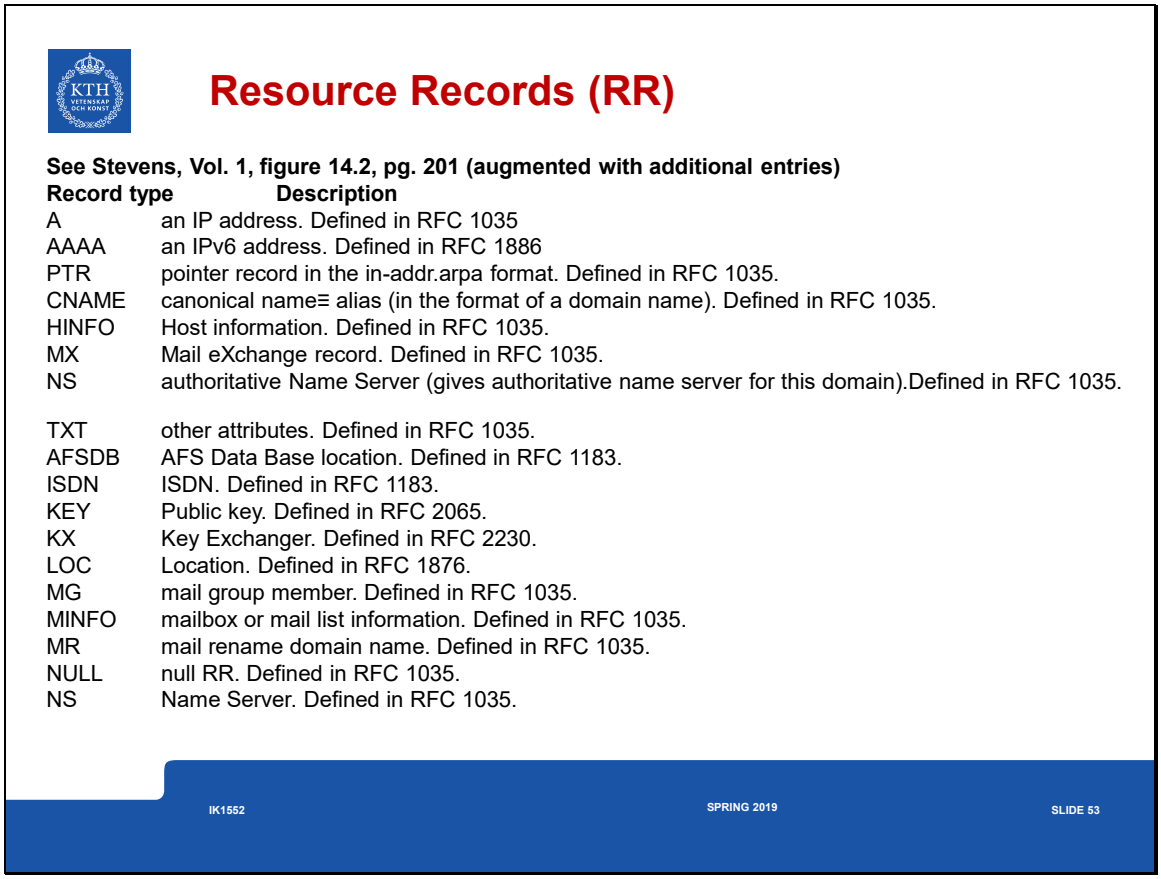
Slide 51


Domain registrars
Internet Corporation for Assigned Names and Numbers (ICANN) Accredited Registrars, the full list is at http://www.icann.org/registrars/accredited-list.html
Even more registrars are on their way to being accredited and operating!
IK1552
SPRING 2019
SLIDE 51

Slide 52


Country Code Top-Level Domains (CCTLDs)
http://www.iana.org/cctld/cctld-whois.htm
For Sweden (SE) SE-DOM, the NIC is: https://www.iis.se/
Contact .SE Registry: registry@iis.se.
Visiting address:
Ringvägen 100, Stockholm, Sweden
Postal address:
The Internet Infrastructure Foundation (Internetstiftelsen i Sverige)
Box 7399
SE-103 91 Stockholm
Sweden
Phone and fax:
Phone: +46-8 452 35 00
Fax: +46-8 452 35 02
Swedish organization number: 802405-0190
VAT number: SE802405019001
E-mail address: info@iis.se
IK1552
SPRING 2019
SLIDE 52

Slide 53



Resource Records (RR)

See Stevens, Vol. 1, figure 14.2, pg. 201 (augmented with additional entries)

Record type	Description
A	an IP address. Defined in RFC 1035
AAAA	an IPv6 address. Defined in RFC 1886
PTR	pointer record in the in-addr.arpa format. Defined in RFC 1035.
CNAME	canonical name≡ alias (in the format of a domain name). Defined in RFC 1035.
HINFO	Host information. Defined in RFC 1035.
MX	Mail eXchange record. Defined in RFC 1035.
NS	authoritative Name Server (gives authoritative name server for this domain).Defined in RFC 1035.
TXT	other attributes. Defined in RFC 1035.
AFSDB	AFS Data Base location. Defined in RFC 1183.
ISDN	ISDN. Defined in RFC 1183.
KEY	Public key. Defined in RFC 2065.
KX	Key Exchanger. Defined in RFC 2230.
LOC	Location. Defined in RFC 1876.
MG	mail group member. Defined in RFC 1035.
MINFO	mailbox or mail list information. Defined in RFC 1035.
MR	mail rename domain name. Defined in RFC 1035.
NULL	null RR. Defined in RFC 1035.
NS	Name Server. Defined in RFC 1035.

P. V. Mockapetris, ‘Domain names - implementation and specification’, *Internet Request for Comments*, vol. RFC 1035 (INTERNET STANDARD), Nov. 1987 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1035.txt
S. Thomson and C. Huitema, ‘DNS Extensions to support IP version 6’, *Internet Request for Comments*, vol. RFC 1886 (Proposed Standard), Dec. 1995 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1886.txt
C. F. Everhart, L. A. Mamakos, R. Ullmann, and P. V. Mockapetris, ‘New DNS RR Definitions’, *Internet Request for Comments*, vol. RFC 1183 (Experimental), Oct. 1990 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1183.txt
R. Allbery, ‘DNS SRV Resource Records for AFS’, *Internet Request for Comments*, vol. RFC 5864 (Proposed Standard), Apr. 2010 [Online]. Available: http://www.rfc-editor.org/rfc/rfc5864.txt
D. Eastlake 3rd, ‘Domain Name System (DNS) IANA Considerations’, *Internet Request for Comments*, vol. RFC 6895 (Best Current Practice), Apr. 2013 [Online]. Available: http://www.rfc-editor.org/rfc/rfc6895.txt
D. Eastlake 3rd and C. Kaufman, ‘Domain Name System Security Extensions’, *Internet Request for Comments*, vol. RFC 2065 (Proposed Standard), Jan. 1997 [Online]. Available: http://www.rfc-editor.org/rfc/rfc2065.txt

D. Eastlake 3rd, 'Domain Name System Security Extensions', *Internet Request for Comments*, vol. RFC 2535 (Proposed Standard), Mar. 1999 [Online]. Available: http://www.rfc-editor.org/rfc/rfc2535.txt
R. Atkinson, 'Key Exchange Delegation Record for the DNS', *Internet Request for Comments*, vol. RFC 2230 (Informational), Nov. 1997 [Online]. Available: http://www.rfc-editor.org/rfc/rfc2230.txt
C. Davis, P. Vixie, T. Goodwin, and I. Dickinson, 'A Means for Expressing Location Information in the Domain Name System', *Internet Request for Comments*, vol. RFC 1876 (Experimental), Jan. 1996 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1876.txt

Slide 54

See Stevens, Vol. 1, figure 14.2, pg . 201 (augmented with additional entries)

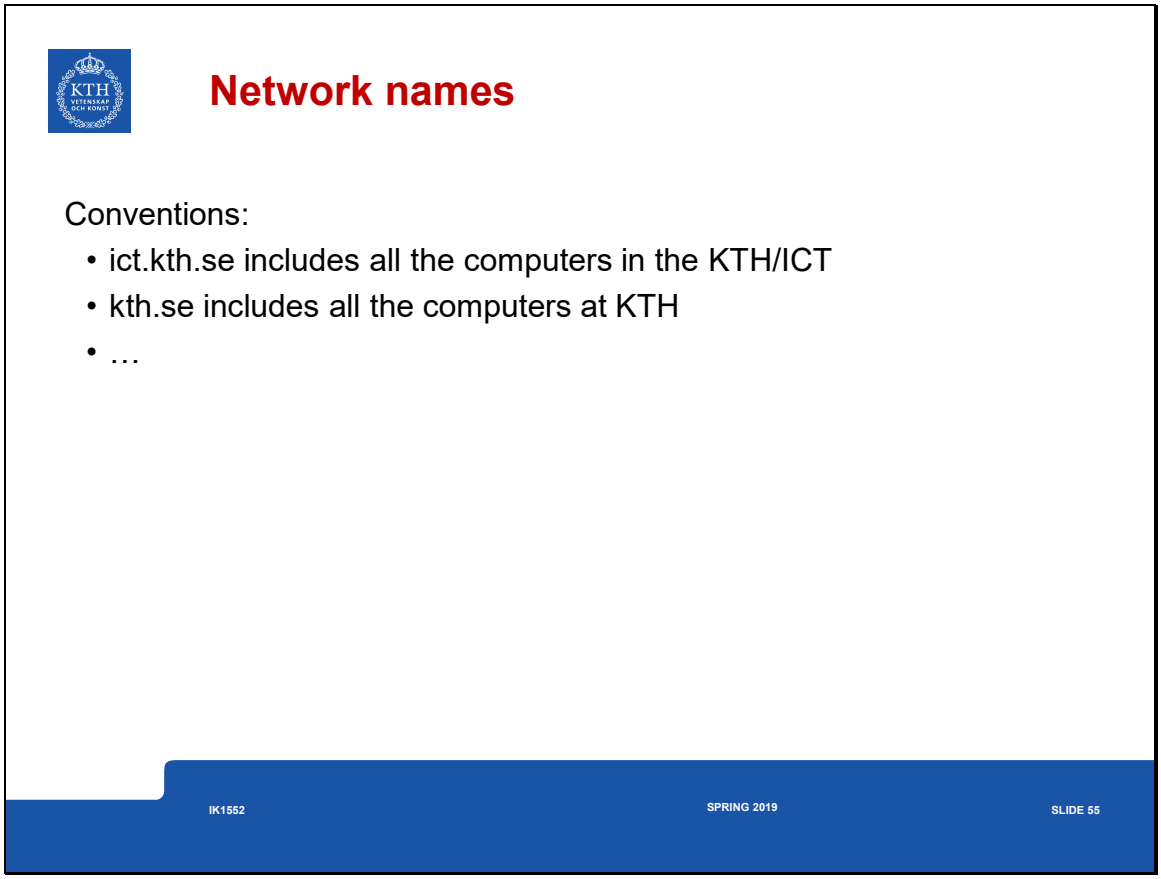
Record type	**Description**
NSAP	Network service access point address. Defined in RFC 1348. Redefined in RFC 1637. Redefined in RFC 1706.
NXT	Next. Defined in RFC 2065.
PX	Pointer to X.400/RFC822 information. Defined in RFC 1664.
RP	Responsible Person. Defined in RFC 1183.
RT	Route Through. Defined in RFC 1183.
SIG	Cryptographic signature. Defined in RFC 2065.
SOA	Start Of Authority. Defined in RFC 1035.
SRV	Server. DNS Server resource record -- RFC 2052, for use with DDNS.
TXT	Text. Defined in RFC 1035.
WKS	Well-Known Service. Defined in RFC 1035.
X25	X25. Defined in RFC 1183.

Note that an number of the RR types above are for experimental use.

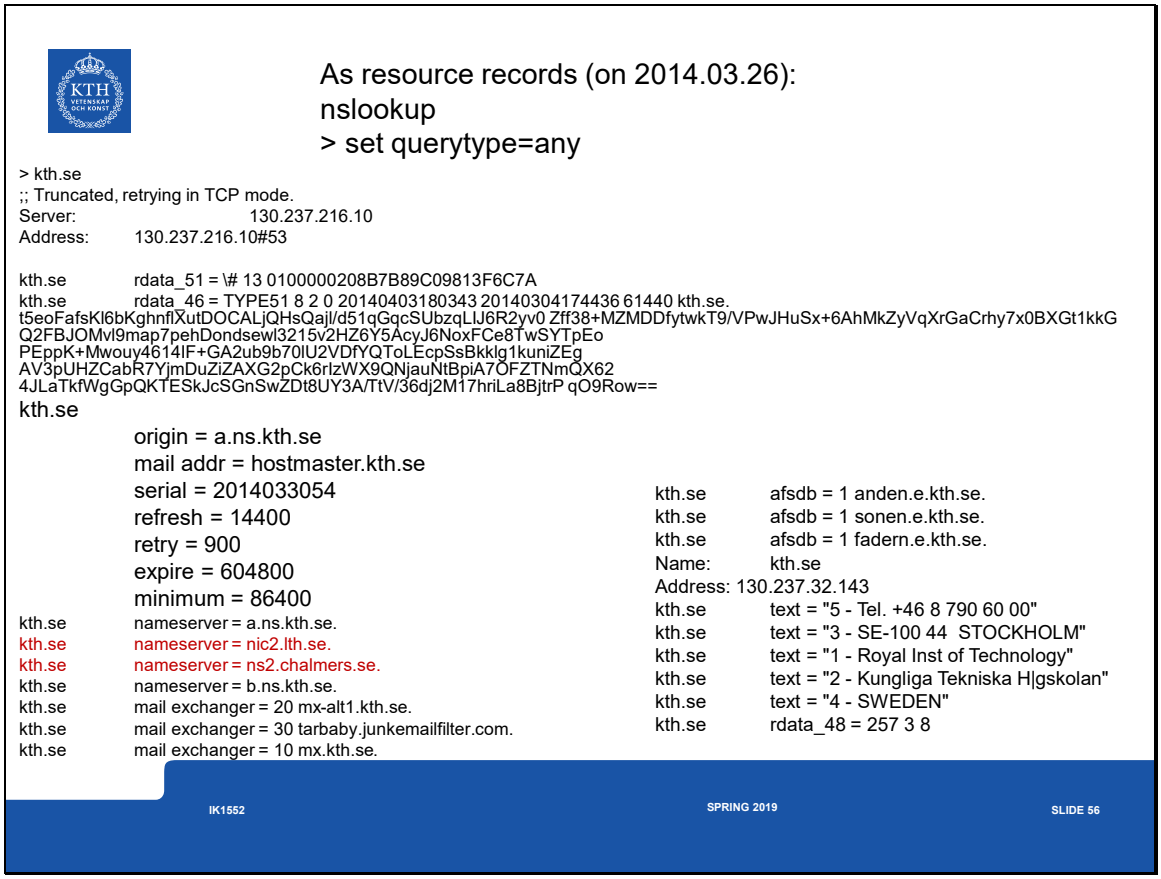
Name of an organization:

ISI.EDU. PTR 0.0.9.128.IN-ADDR.ARPA.

IK1552 SPRING 2019 SLIDE 54


Network names
Conventions:
• ict.kth.se includes all the computers in the KTH/ICT
• kth.se includes all the computers at KTH
• …
IK1552
SPRING 2019
SLIDE 55

Slide 56



As resource records (on 2014.03.26):
nslookup
> set querytype=any

> kth.se
;; Truncated, retrying in TCP mode.
Server: 130.237.216.10
Address: 130.237.216.10#53

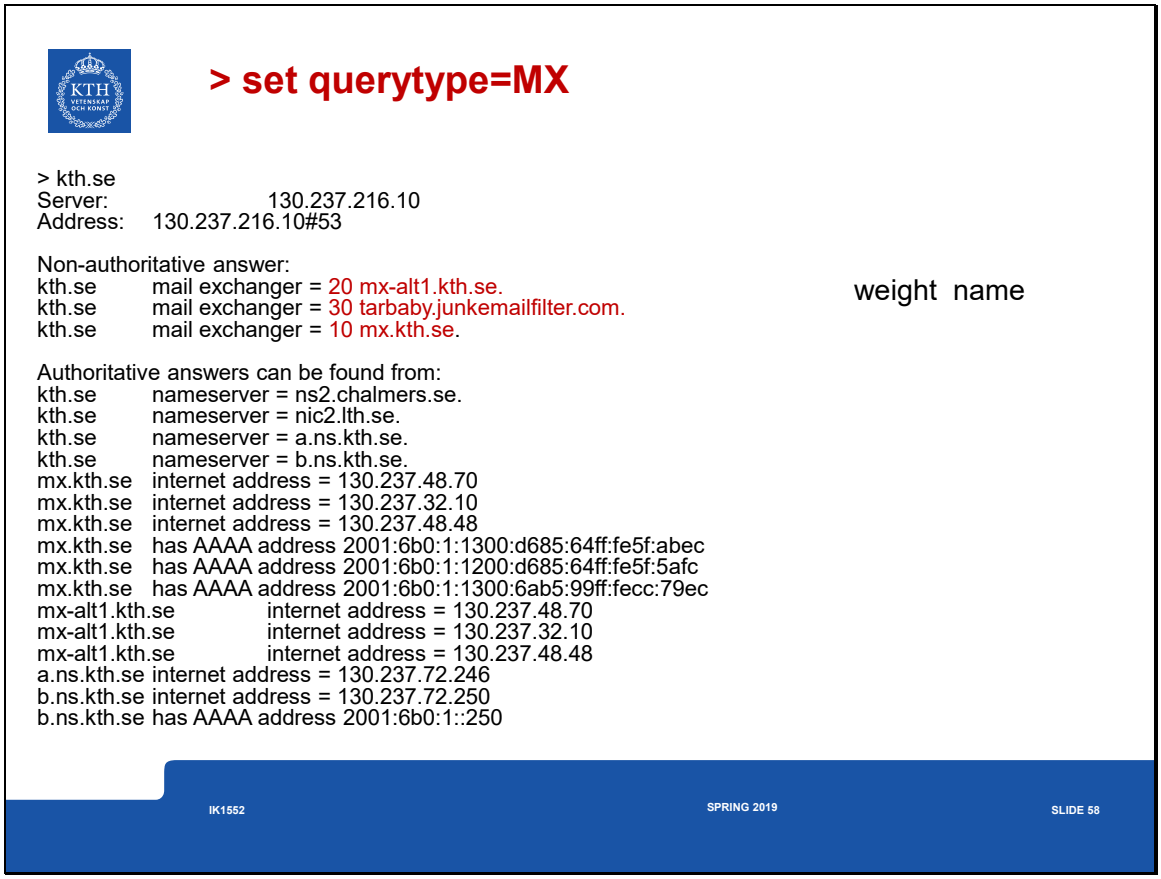
kth.se rdata_51 = \# 13 0100000208B7B89C09813F6C7A
kth.se rdata_46 = TYPE51 8 2 0 20140403180343 20140304174436 61440 kth.se.
t5eoFafsKl6bKghnflXutDOCALjQHsQajl/d51qGqcSUbzqLIJ6R2yv0 Zff38+MZMDDfytwkT9/VPwJHuSx+6AhMkZyVqXrGaCrhy7x0BXGt1kkG
Q2FBJOMvl9map7pehDondsewl3215v2HZ6Y5AcyJ6NoxFCe8TwSYTpEo
PEppK+Mwouy4614lF+GA2ub9b70lU2VDfYQToLEcpSsBkklg1kuniZEg
AV3pUHZCabR7YjmDuZiZAXG2pCk6rIzWX9QNjauNtBpiA7OFZTNmQX62
4JLaTkfWgGpQKTESkJcSGnSwZDt8UY3A/TtV/36dj2M17hriLa8BjtrP qO9Row==
kth.se
origin = a.ns.kth.se
mail addr = hostmaster.kth.se
serial = 2014033054
refresh = 14400
retry = 900
expire = 604800
minimum = 86400
kth.se nameserver = a.ns.kth.se.
kth.se nameserver = nic2.lth.se.
kth.se nameserver = ns2.chalmers.se.
kth.se nameserver = b.ns.kth.se.
kth.se mail exchanger = 20 mx-alt1.kth.se.
kth.se mail exchanger = 30 tarbaby.junkemailfilter.com.
kth.se mail exchanger = 10 mx.kth.se.
kth.se afsdb = 1 anden.e.kth.se.
kth.se afsdb = 1 sonen.e.kth.se.
kth.se afsdb = 1 fadern.e.kth.se.
Name: kth.se
Address: 130.237.32.143
kth.se text = "5 - Tel. +46 8 790 60 00"
kth.se text = "3 - SE-100 44 STOCKHOLM"
kth.se text = "1 - Royal Inst of Technology"
kth.se text = "2 - Kungliga Tekniska H|gskolan"
kth.se text = "4 - SWEDEN"
kth.se rdata_48 = 257 3 8

IK1552 SPRING 2019 SLIDE 56

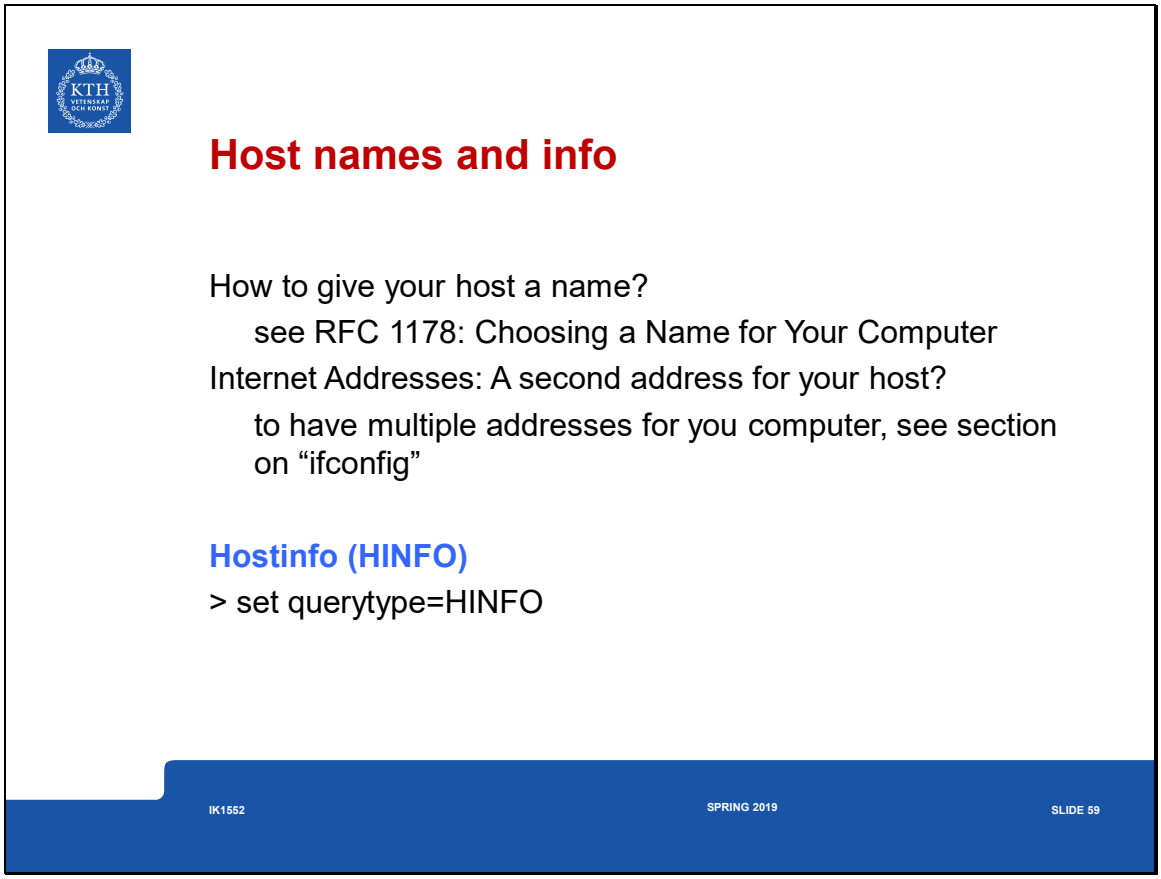
Slide 57


kth.se (continued) – for DNSSEC
kth.se rdata_48 = 257 3 8 AwEAAeqLuvtNBVEmjzUQCkRLHcDbmOl83rOl8wmwj+wIvQ3l+Tw2MsMJ
9o1J1U6p6yRJyHis3V4gC5ltXRUKuU4waOqCgFSXMReq+BvFk80HjEN+
/PRpKgRCU537mQcDgZhBukOa4cYydXBMQ8FLMh2yfmP2DdK1iBNTXOPS
IJCPIZyeJ6/hh+fURZeFed+t9CrcZpvsn19uHhJ+YBi5UZtY0WmsnlSX
ltgmt0m3YYFVu5NRXD7HQadjxL5D8gQPDwdEXDcCB0BhvVkZB7Bt1+7q
1pCucNUFFWnnxSgmKoFPg832ELvuuGxafAH+S/Z5GJXPbrVgKiBDDFLr OmaqRkDciJM=
kth.se rdata_48 = 256 3 8 AwEAAelwmjXgoN8dIcWEy2QN40Vu42hdcjliLY5dMDvimwup/9NlkYlo
nnwRWo8gyvr53Ol+K/TlindQK0og4PQRKr9uyLHV42BwMsump9kcVhzy
Hy+hSNsqffWyFxFlK+3GpfodClxTSTD1uMlgUxF9vA4WmvejKCd/U/R
D0LB7TB+Zcl8VqNhpV4VLyc0UokgMHSdVO6lKNVIlXKUiYkDK8Fi8JK1
XSPv45MSQ7zcHNv33DZeO0it9anVxq/rAF8fdn3yb8dfiKud18jrm1wG
bvwT56sjIdDMFh0Skf53VJBxsGCZF/jdefy0U002zTjoDx9Hpz/nLX3V s19lBgtf8+s=
kth.se rdata_65534 = \# 5 0856710001
kth.se rdata_46 = DNSKEY 8 2 7200 20140414140000 20140315130000 22129 kth.se.
Nca6AROKLx2TU+bywWVBJHVbyNx/VXGbaDOAxx5C+V+O/4ItikDZd2Ri
CRgrBF/sXRXzL8lgyZJ1mdOtPOsS0WbE3fNzbjMEenfRHDCFbXuWa9t2
l83y31DB1ClkVHvhg8A3tmmjlTclqPbatBqEcqQwFTBSfEYFxguy3+qG
uRvnqod05PNnQrynf/4O9TEETUmF2qECaXOP22xUQcZTMEmOMrKv6iD7
XKL85ucyTN6dxxU/Nv7Q6yqwMumY0M8+uhYlcfYw/bfY74wXCmB6ffi9
1q+l8xNbzljdHARRn4h/dE4PGfk8I1ciun2m3fEPps7zvK5FkJZHqsFM UMHS3g==
kth.se rdata_46 = DNSKEY 8 2 7200 20140414140000 20140315130000 61440 kth.se. UuZly1y4vnKPCd+YesOCF7OatayU3dMJ54QCGN7fA/7H+JJrMV8Yd7 AlOr66/nj6D4AsMOgPmG1lROq6lRnFhh/B7EJibi1YQthnYC5lnNE8Y4 AaUG8FMxlEzrliTCORlCIUEqK2ZEBF26VtHTMNzxisszOck8tudBGZW/d hU3tAlm5vvwCCl0guOe1Y7otTKaiYVlqf06tYHKK8GZ/E2HeWwcb5J Vkp+g54IS630tNsnRJ1aKd9nkSxEF/HyhAkHMEkk7G13rKQAsxmV3wJ 2ihwrF53c5cMQDID10rOyRPm2n/Vxo919CpplkCZC0fWoecIHQcgn5uU aTrC3g==
kth.se rdata_46 = TYPE65534 8 2 0 20140414130840 20140315130005 61440 kth.se. wyQr9s9InhTLF0sAKoPAXcgOgZtI32VXPmDiFdyUtOve2hReV7Remhw bhsTEzSC7mtjhGYJR7E0cXo/9xqQmLiT+8z+Zm3lVjlmkkl80Rl5o8xG 9laW3PubjTppa/JrKuov1MCirowaNsFgvUHZKkoOo4ZXXn5z69cHa3j OLDUsGl0CazwDJkJTP3aJWbhlzADCaPWUNTfhhBx57wuTkU4SGkkkl0M gKCX3cTiNliCim794QTcPYnJEZXKk336SidzkU4O8JLKe/pJoFOJ0V9 zzzmNKLQtFstn8hSYLUExX42qXWIITziNoE4R/dzs6oPYmH/3gL90hh F/gZ0Q==
kth.se rdata_46 = SOA 8 2 1800 20140425045132 20140326035132 61440 kth.se. Hp4TnWOJHT/kvG66VPfGpDJZHsrzak4/OG/oYejxza8tNxGs527ye4ZE 0SFhtFOJFQr4/U+GqmJL+fMQJUxMKV1xE6M0CvxE+kagbgYlgIPxg9ZW O0KQNbIEUq2+AtzotTgy1CyjdM5tFg5WTlGUlBmHNw+dNrLLvcJgcLlc 7jqFUdjlsbgQuBhOVBH26jNvF8MVcT4cmHu5fRmS0Yi0tSKHYS9g1NRW KMJiNoHDgyzOLU8/q4czcvOnqL4dcxV51tkYEW6B+23lexG9eNqpbeIP Medbr0GnwP/cRG1WRKoSVOZCDRaZ0JSMao2QJ/CdynCLFaH75RJNp3WKGdDg==
kth.se rdata_46 = TXT 8 2 1800 20140424045412 20140325045214 61440 kth.se. CzcQaHZjNBl20ZlKTb3tf/cRXJwtAtssdAGAgT0WqL1HKAzbShxdlU93 o4Wj0RPBWwUp+l3LmNyg+S9cA55vz9dtmALV3Oovdm/0X4uQRD2KjR96 acBcMKdYr3CpSal9fjdcuVRdMCahYrlrAtYEPX3Hb2CLFd+f776+4pX+ RPpu3KjPGLwgwtgUf3DK4MdBIlqvUFmfqgbdfYL1MbJDCfjRPZNKzNB ItoY1QF++haxzlfUmtp2QVJfkcof61tIBnqp2peZfA3YNiywnsHaeyNF kRXkWT1gW7UQhTFWpXedqH+kb4bshHGBs9MTZXuzTaXI7wK9CmDMxyML fzVe7g==
kth.se rdata_46 = A 8 2 60 20140419103918 20140320094725 61440 kth.se. t78lDsk8mG8lVS8nfUCu6oVM5GMlIT03EegJR6pEqDjCAQR32Eew1W8x uh+DvY3S6ni1PbFS/H3sZarTeaUvVWUlfsRvslLs4Zo0q68MEjH0OJUp mQhqW3EGFF0F85QOldb4ljQAWpul+cKVQlD2RaPp6PuhOEz4tAQFVPQ+ 0MZo4esylH7nwsGXioIRjm58CQWjUUlcYH/Kzk+daeKgUoivtdSUbCKE Z9M63/k1zkQl5HDH7LDC3m52wp06FtXYfnRdXV5Mh1X1fMEZstWoysZz N2qGarNpY1bP9V0nVZMh5sYUm9l1AfD0lJkuWSke7SLnxqnj9ZBegv7u EAwBSQ==
kth.se rdata_46 = AFSDB 8 2 1800 20140424060154 20140325050711 61440 kth.se. csOHtHfau8sDCH0rbktEEkCCRS3Cmk4QtS7zBwJB8LNPkdW6tciHHtJ gQv0S/PkBf1ehord4rcyDC9a+3ke4nJR22T3Cc771rTy9iILJ7xBR/4f Clrs/lbyHqqQiHRoFyo9VAhfD8jCMqlNBLS59Hr5CnfrliHcwrHhe/Yv 8fiudCjHtnJzgdAi5g3nsgkkY/t/o4NsBhpy5hn8HZ+5OWWnpMT6m6T5 cmRDJLOfjSaQHj3NvUzUe00GYxefyMQaLf4o8v+KxgAN0KUOCqOrvlVu sTgEW1AwZ0RqmnHnmMJ+G0etWCxWncmzI7DMmWHLI3UBNSCXJiiP6p9a/T1TQ==
kth.se rdata_46 = MX 8 2 60 20140419103918 20140320094725 61440 kth.se. mNp5okAwUQ4gMHBj9yANinao2vnXz3F1+nxEDyvQTlHj5LtUHyxOB0js IDnM6O0YohtLEacc5SQxRFlcgUxD/TXkJxQC005p108l5K26FTjCr1iw sOTMCW7tkh6GBuI5S6iTjjisduMbtWXKD8j0tfMdVToUUKA7psSMEAGD zQZZSDOER0fDCa6OMdg7fgzvHy62i75LpsGgwXNtm/NiofirkUYoiJc dHAl5eWSz6hpj5SWDFVML81NGAnLVk+r1qQgGyemKKGxxFwSdLxaw/Xf W4Kr8A3FpQzMUjXfRPmwdQADrS/E6KbsFfLu2rCuf6i/0Wg6dj1IJ7+ S/qQSw==
kth.se rdata_46 = NS 8 2 1800 20140403201036 20140304193741 61440 kth.se. S25mGKBUAMFO5G+xYn5rHMunjHJwZV5GSL2mmPrGRFQ94oJ3bmj5+oTq kxthQ7xlW4ypLu9rXntLgU6xN9601mg+h5ZjjNhbxPfv8lUGqJuOoj4D jJWVX3uRDTCJach9GqE5UgrrrfJkJr3nEKbX9Cco5JkdMh74sycxVTY 8sl+QhNASgZvs3QUuUCX7teEM6c2qeu/5kFq35quK5vRW+J16hs910V 4hDbKY36TdC++B4q5lhtchdkJZNGO3Z+Mv66GyCyCFS/AVOng/QFHwEZ D9zm7n89PRekueGHAfFV9gjm9lldTjT2neyDRVtFAkdwWzx+N3r743XZ KU3gCA==
kth.se has AAAA address 2001:6b0:1:1200::82ed:208f
kth.se rdata_46 = AAAA 8 2 60 20140404010055 20140305000109 61440 kth.se. vacJ1G1acyEA3ThlFVwI5dhoTeckvghGgMYib7Bq2hp6AP6dsltsyLcg ng29sacnQoHOA0xll/zqpTmh06T9h9Jd4BBDvD+0EN4eREj/jvAhu19 QmWU5RFpjeWWQ7ICR2AphpXaYGGzE1hGTRyUAGEHJvjBsHWG5+YQmE+pH2edJYVMyUAXZZcEVWXbBsIpiEN8ZingT9ESPC+FgyAcy dUKzWmbqSt7057MLDPpTymHpshC3ssTL5MQp5u0xBdEcA1Te8VQxb 1TvasmkoI0FfqNOFRQ/SowcP1qHkFQDHuiLfgIV+2MvicQ1LFC3nLOLU 3VicjQ==

Slide 58

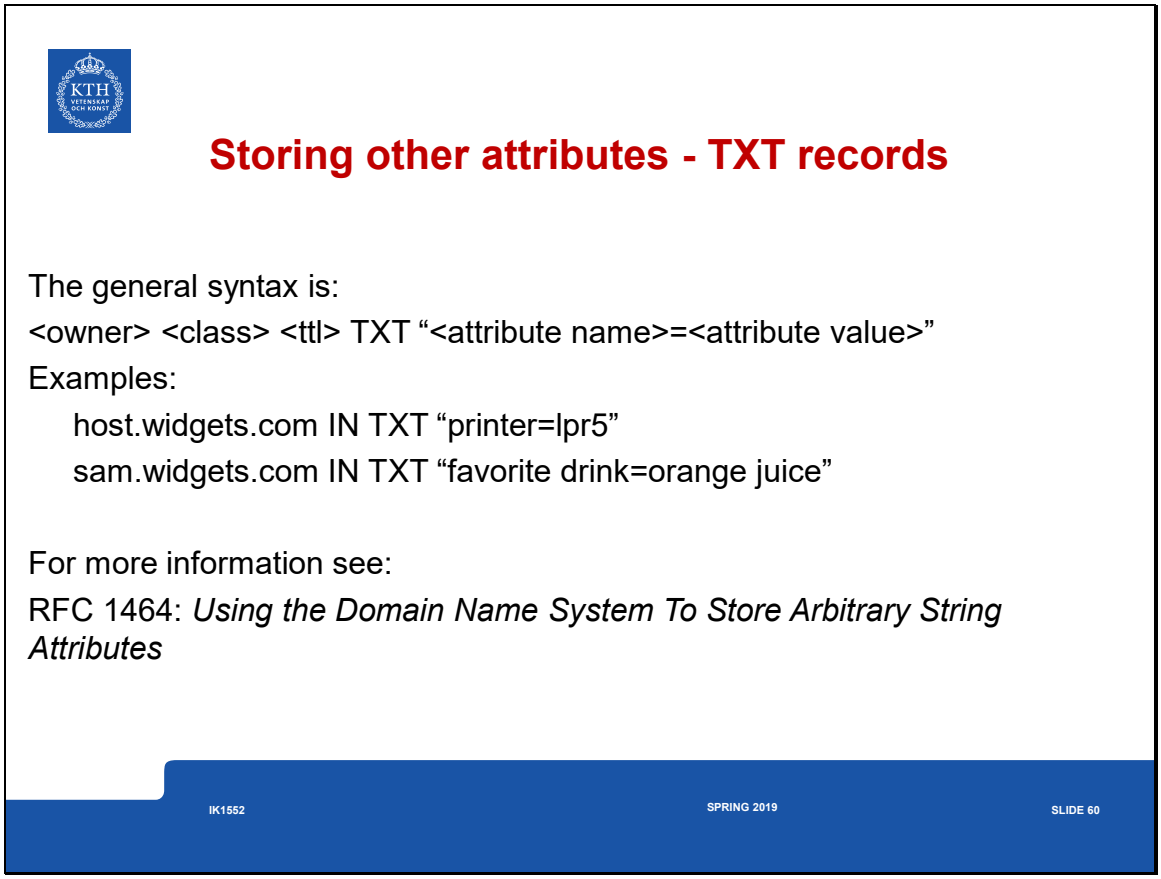

> set querytype=MX
> kth.se
Server: 130.237.216.10
Address: 130.237.216.10#53
Non-authoritative answer:
kth.se mail exchanger = 20 mx-alt1.kth.se.
kth.se mail exchanger = 30 tarbaby.junkemailfilter.com.
kth.se mail exchanger = 10 mx.kth.se.
weight name
Authoritative answers can be found from:
kth.se nameserver = ns2.chalmers.se.
kth.se nameserver = nic2.lth.se.
kth.se nameserver = a.ns.kth.se.
kth.se nameserver = b.ns.kth.se.
mx.kth.se internet address = 130.237.48.70
mx.kth.se internet address = 130.237.32.10
mx.kth.se internet address = 130.237.48.48
mx.kth.se has AAAA address 2001:6b0:1:1300:d685:64ff:fe5f:abec
mx.kth.se has AAAA address 2001:6b0:1:1200:d685:64ff:fe5f:5afc
mx.kth.se has AAAA address 2001:6b0:1:1300:6ab5:99ff:fecc:79ec
mx-alt1.kth.se internet address = 130.237.48.70
mx-alt1.kth.se internet address = 130.237.32.10
mx-alt1.kth.se internet address = 130.237.48.48
a.ns.kth.se internet address = 130.237.72.246
b.ns.kth.se internet address = 130.237.72.250
b.ns.kth.se has AAAA address 2001:6b0:1::250
IK1552
SPRING 2019
SLIDE 58

Slide 59



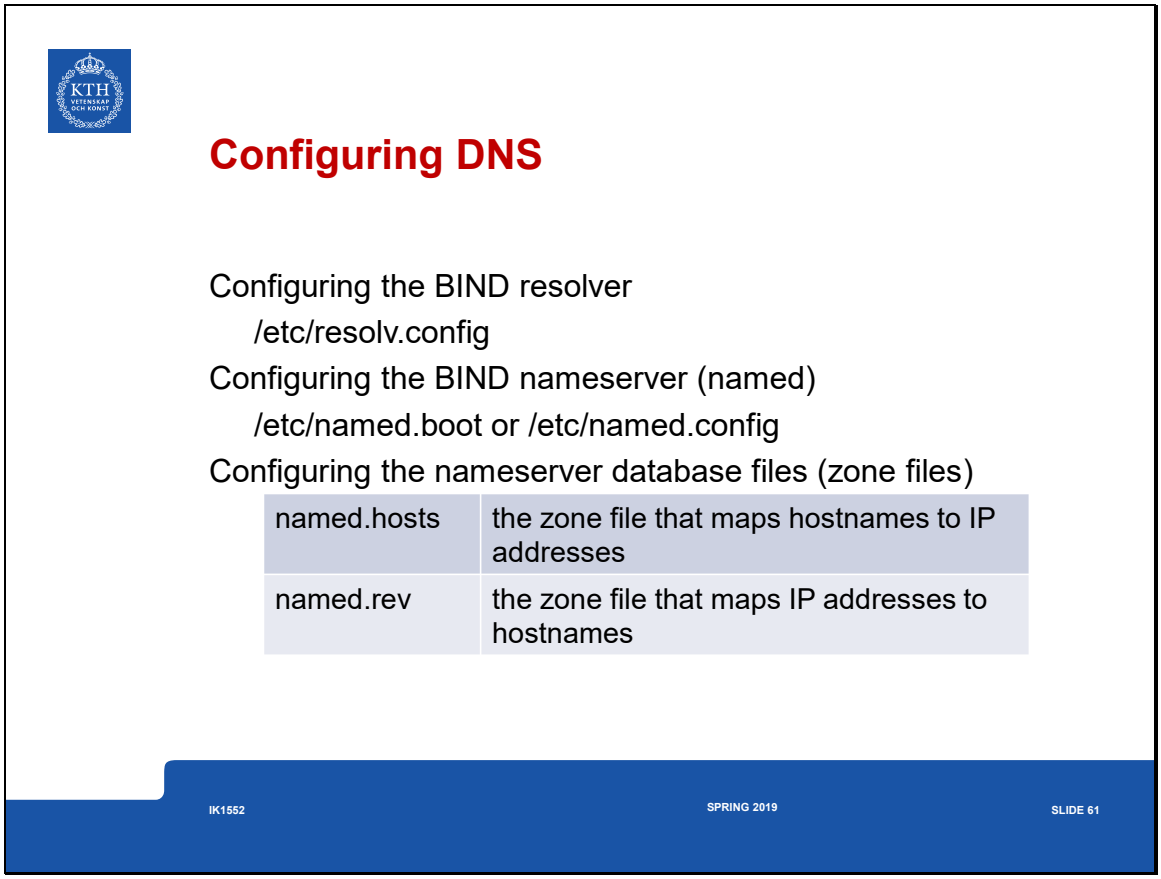
D. Libes, ‘Choosing a name for your computer’, *Internet Request for Comments*, vol. RFC 1178 (Informational), Aug. 1990 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1178.txt

Slide 60



R. Rosenbaum, 'Using the Domain Name System To Store Arbitrary String Attributes', *Internet Request for Comments*, vol. RFC 1464 (Experimental), May 1993 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1464.txt

Slide 61



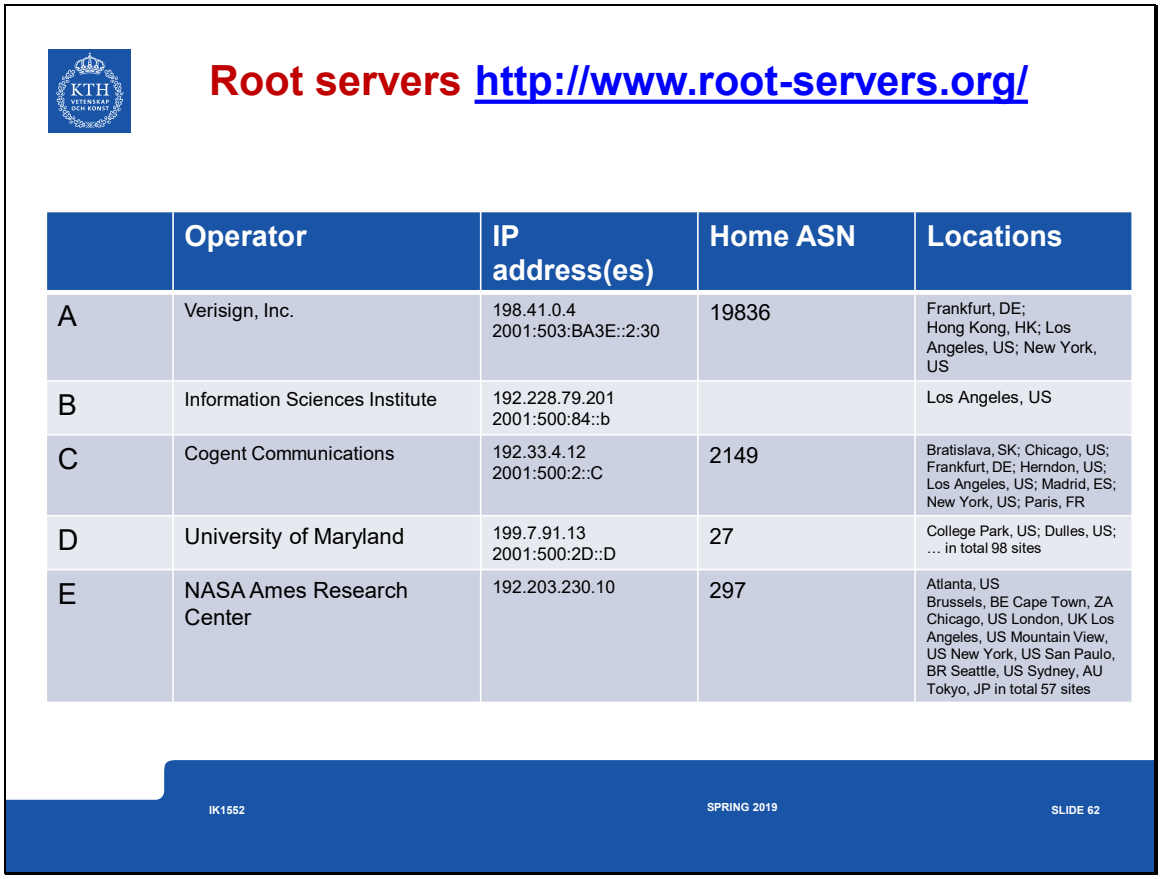
Configuring DNS

Configuring the BIND resolver
/etc/resolv.config
Configuring the BIND nameserver (named)
/etc/named.boot or /etc/named.config
Configuring the nameserver database files (zone files)

named.hosts	the zone file that maps hostnames to IP addresses
named.rev	the zone file that maps IP addresses to hostnames

IK1552 SPRING 2019 SLIDE 61

Slide 62



Root servers http://www.root-servers.org/

	Operator	IP address(es)	Home ASN	Locations
A	Verisign, Inc.	198.41.0.4 2001:503:BA3E::2:30	19836	Frankfurt, DE; Hong Kong, HK; Los Angeles, US; New York, US
B	Information Sciences Institute	192.228.79.201 2001:500:84::b		Los Angeles, US
C	Cogent Communications	192.33.4.12 2001:500:2::C	2149	Bratislava, SK; Chicago, US; Frankfurt, DE; Herndon, US; Los Angeles, US; Madrid, ES; New York, US; Paris, FR
D	University of Maryland	199.7.91.13 2001:500:2D::D	27	College Park, US; Dulles, US; ... in total 98 sites
E	NASA Ames Research Center	192.203.230.10	297	Atlanta, US Brussels, BE Cape Town, ZA Chicago, US London, UK Los Angeles, US Mountain View, US New York, US San Paulo, BR Seattle, US Sydney, AU Tokyo, JP in total 57 sites

IK1552 SPRING 2019 SLIDE 62

Slide 63

Root servers (continued)

	Operator	IP address(es)	Home ASN	Locations
F	Internet Systems Consortium, Inc.	192.5.5.241 2001:500:2f::f	3557	total 58 sites

Slide 64

Root servers (continued)

	Operator	IP address(es)	Home ASN	Locations
G	U.S. DOD Network Information Center	192.112.36.4	5927	Columbus, US; Fussa, JP; Honolulu, US; Naples, IT; San Antonio, US; Stuttgart-Vaihingen, DE
H	U.S. Army Research Lab	128.63.2.53 2001:500:1::803f:235	13	Aberdeen Proving Ground, US; San Diego, US
I	Netnod (formerly Autonomica)	192.36.148.17 2001:7fe::53	29216	In total 50 sites (see below)

Amsterdam, NL · Ankara, TR · Ashburn, US · Bangkok, TH · Beijing, CN · Belgrade, RS · Brussels, BE
Bucharest, RO · Chicago, US · Doha, QA · Dubai, AE · Frankfurt, DE · Geneva, CH · Helsinki, FI · HongKong, CN
Jakarta, ID · Johannesburg, ZA · Karachi, PK · Kathmandu, NP · Kiev, UA · Kigali, RW · Kuala Lumpur, MY
London, UK · Lulea, SE · Luxembourg City, LU · Manama, BH · Manila, PH · Miami, US · Milan, IT · Mumbai, IN
Oslo, NO · Paris, FR · Perth, AU · Port Vila, VU · Porto Alegre, BR · San Francisco, US · Singapore, SG
St Petersburg, RU · Stockholm, SE · Taipei, TW · Tallinn, EE · Thimphu, BT · Tokyo, JP · Ulaanbaatar, MN
Washington DC, US · Wellington, NZ · Wien, AT · Yerevan, AM

IK1552 SPRING 2019 SLIDE 64

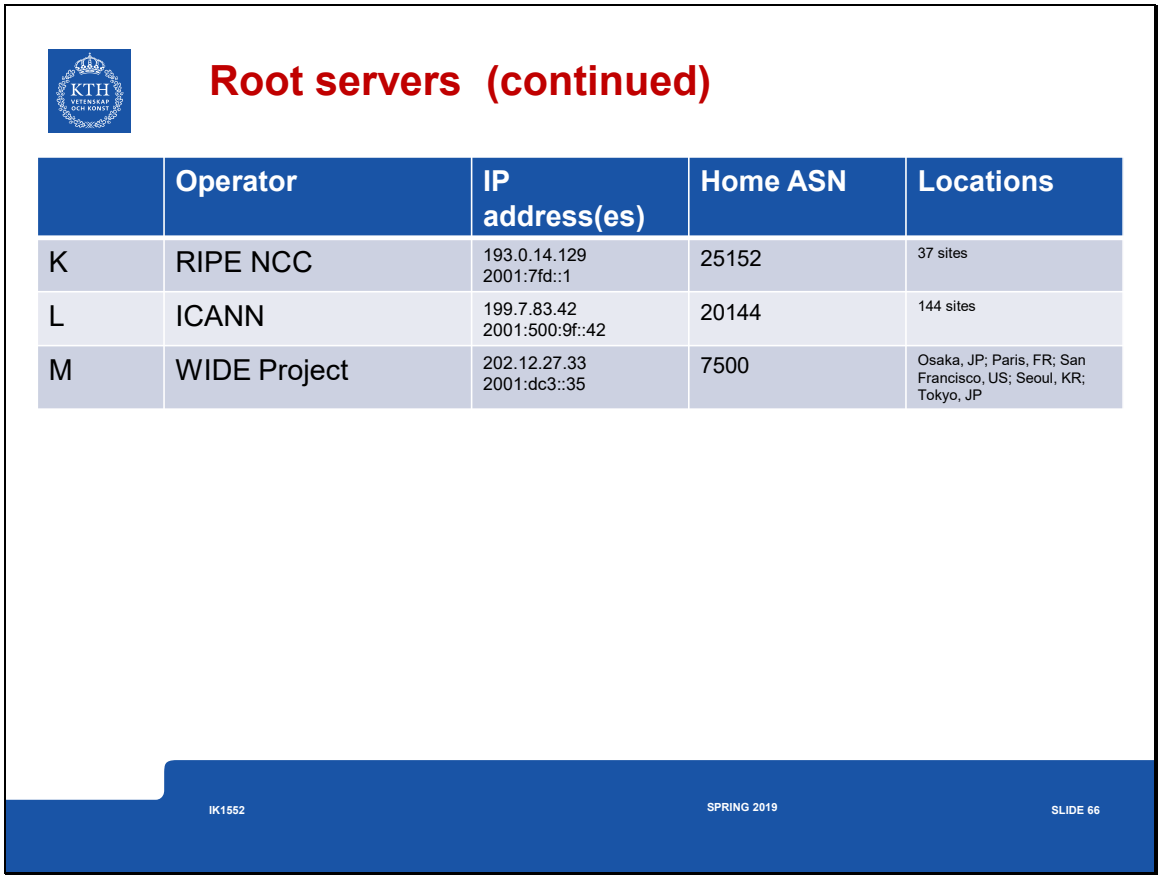
Slide 65

Root servers (continued)

	Operator	IP address(es)	Home ASN	Locations
J	Verisign, Inc.	192.58.128.30 2001:503:C27::2:30	26415	104 sites

Amsterdam, NL · Ashburn, US · Athens, GR · Atlanta, US · Bangalore, IN · Bangkok, TH · Bangkok, TH
Barcelona, ES · Beijing, CN · Belgrade, RS · Boston, US · Brasilia, BR · Bratislava, SK · Bucharest, RO
Buenos Aires, AR · Cairo, EG · Cape Town, ZA · Cebu City, PH · Centro Junco do Serido, BR · Chicago, US
Chicago, US · Chicago, US · Dakar, SN · Dallas, US · Dar es Salaam, TZ · Denver, US · Des Moines, US · Dhaka, BD
Dublin, IE · Geneva, CH · Guarabira, BR · Helsinki, FI · Hong Kong, HK · Honolulu, US · Johannesburg, ZA
Kalamazoo, US · Kaunas, LT · Kigali, RW · Klagenfurt, AT · Kuala Lumpur, MY · Lagoa Seca, BR · Lisbon, PT
Ljubljana, SI · Luxembourg City, LU · Lyon, FR · Madrid, ES · Manila, PH · Melbourne, AU · Miami, US · Milan, IT
Montreal, CA · Moscow, RU · Mumbai, IN · Mumbai, IN · Nairobi, KE · New Castle, US · New Dehli, IN · Oslo, NO
Palo Alto, US · Paris, FR · Perth, AU · Pirassununga, BR · Porto Alegre, BR · Prague, CZ · Quezon City, PH
Reno, US · Reykjavik, IS · Riga, LV · Rio de Janeiro, BR · Rome, IT · San Francisco, US · San José, CR
San Jose, US · San Juan, PR · Sao Paulo, BR · São João do Rio do Peixe, BR · Seattle, US · Seoul, KR · Singapore, SG
Sofia, BG · Stockholm, SE · Sydney, AU · Sydney, AU · Taipei, TW · Tallinn, EE · Tamuning, GU · Tel Aviv, IL
Tokyo, JP · Toronto, CA · Turin, IT · Vancouver, CA · Warsaw, PL · Warsaw, PL · Wellington, NZ · Willemstad, CW
Yerevan, AM · Zagreb, HR · Zurich, CH

Slide 66

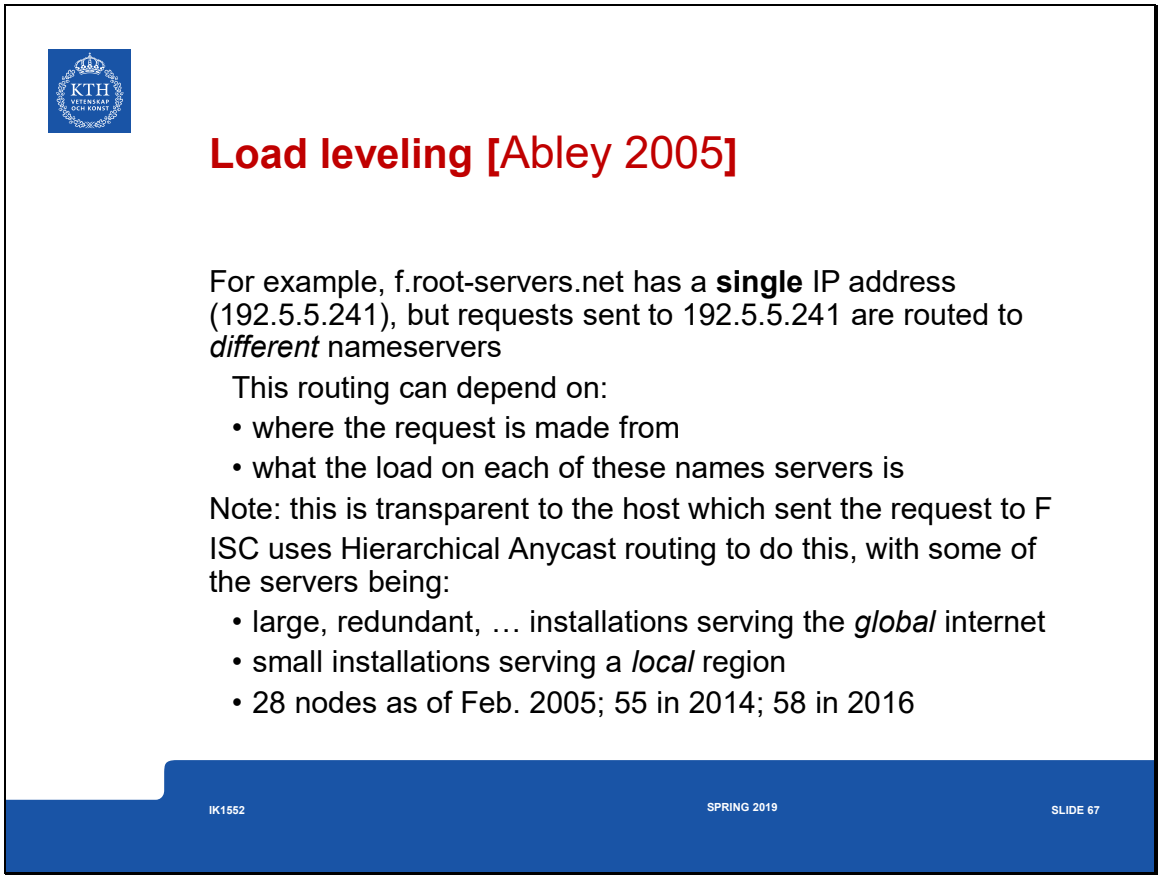


Root servers (continued)

	Operator	IP address(es)	Home ASN	Locations
K	RIPE NCC	193.0.14.129 2001:7fd::1	25152	37 sites
L	ICANN	199.7.83.42 2001:500:9f::42	20144	144 sites
M	WIDE Project	202.12.27.33 2001:dc3::35	7500	Osaka, JP; Paris, FR; San Francisco, US; Seoul, KR; Tokyo, JP

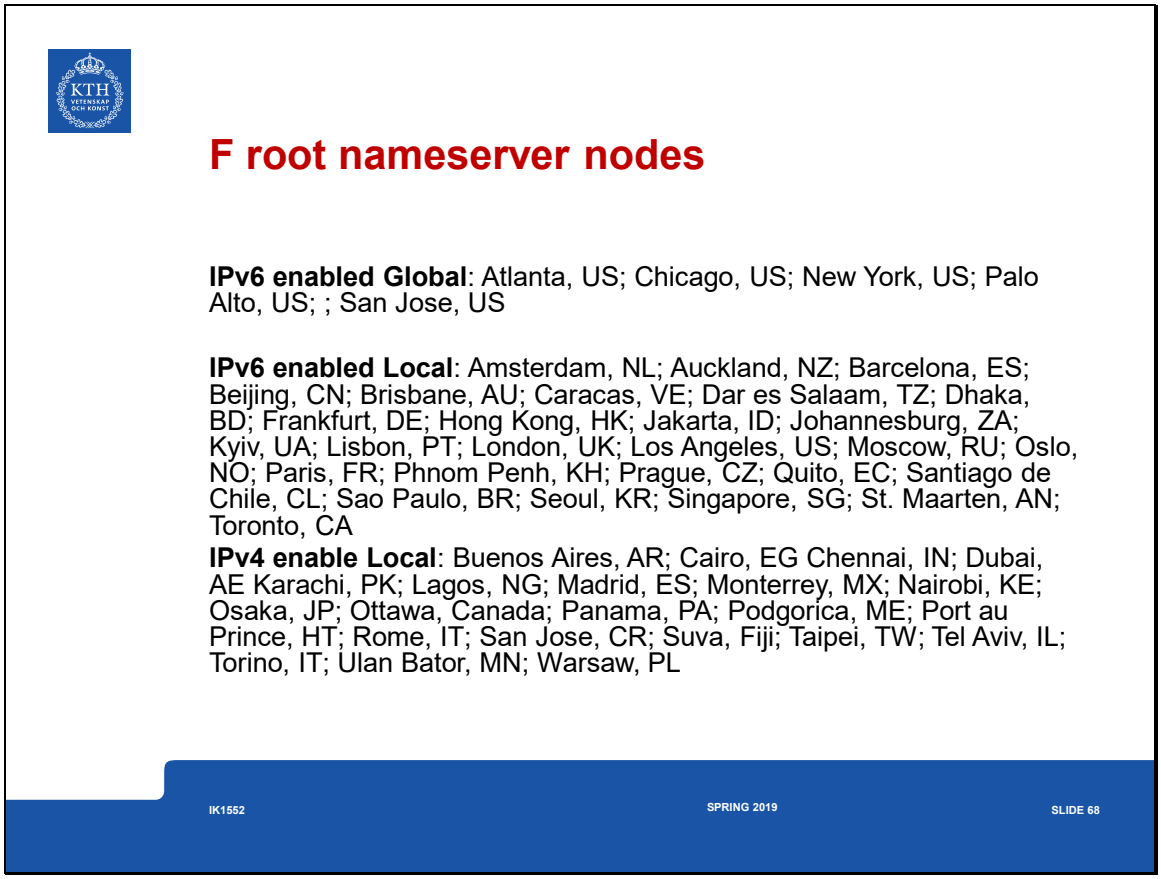
IK1552 SPRING 2019 SLIDE 66

Slide 67

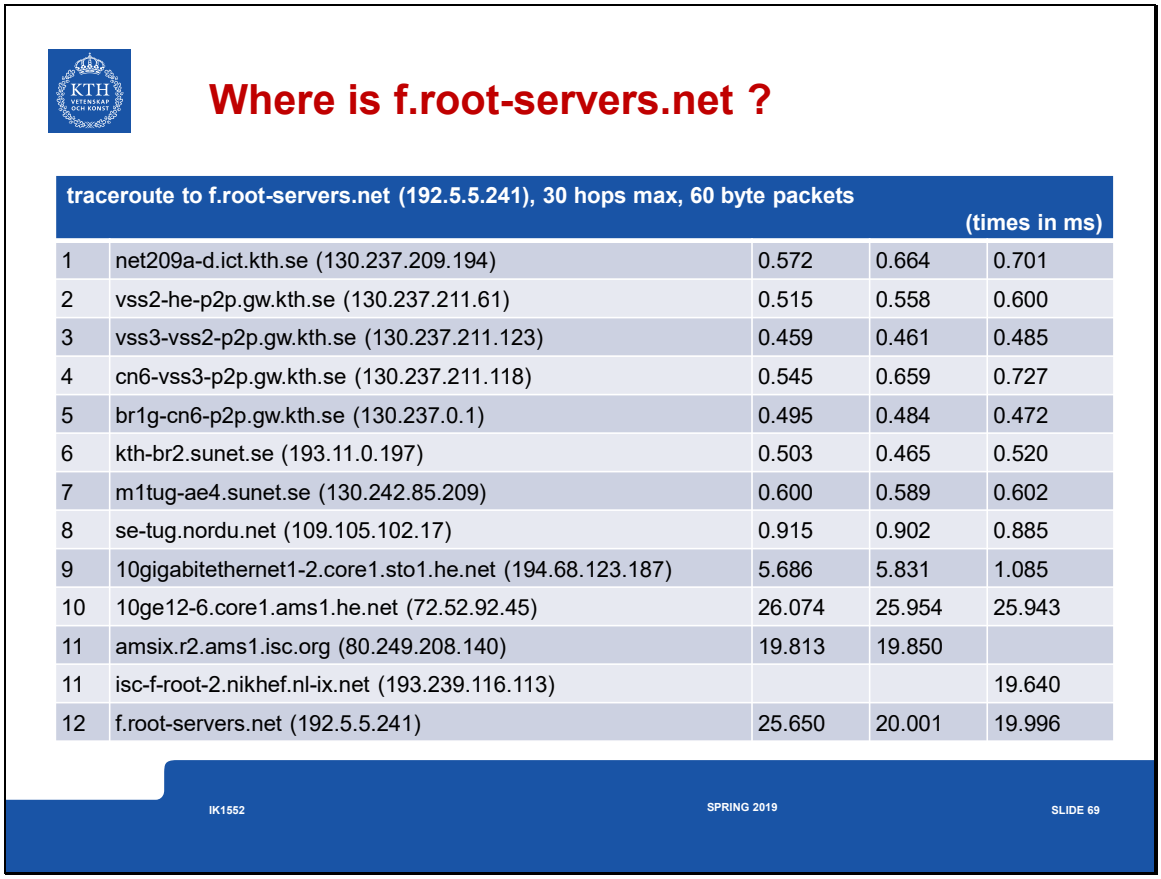


Joe Abley, f.root-servers.net, NZNOG 2005, February 2005, Hamilton, NZ
http://www.isc.org/pubs/pres/NZNOG/2005/F%20Root%20Server.pdf

Slide 68


F root nameserver nodes
IPv6 enabled Global: Atlanta, US; Chicago, US; New York, US; Palo Alto, US; ; San Jose, US
IPv6 enabled Local: Amsterdam, NL; Auckland, NZ; Barcelona, ES; Beijing, CN; Brisbane, AU; Caracas, VE; Dar es Salaam, TZ; Dhaka, BD; Frankfurt, DE; Hong Kong, HK; Jakarta, ID; Johannesburg, ZA; Kyiv, UA; Lisbon, PT; London, UK; Los Angeles, US; Moscow, RU; Oslo, NO; Paris, FR; Phnom Penh, KH; Prague, CZ; Quito, EC; Santiago de Chile, CL; Sao Paulo, BR; Seoul, KR; Singapore, SG; St. Maarten, AN; Toronto, CA
IPv4 enable Local: Buenos Aires, AR; Cairo, EG Chennai, IN; Dubai, AE Karachi, PK; Lagos, NG; Madrid, ES; Monterrey, MX; Nairobi, KE; Osaka, JP; Ottawa, Canada; Panama, PA; Podgorica, ME; Port au Prince, HT; Rome, IT; San Jose, CR; Suva, Fiji; Taipei, TW; Tel Aviv, IL; Torino, IT; Ulan Bator, MN; Warsaw, PL
IK1552
SPRING 2019
SLIDE 68

Slide 69

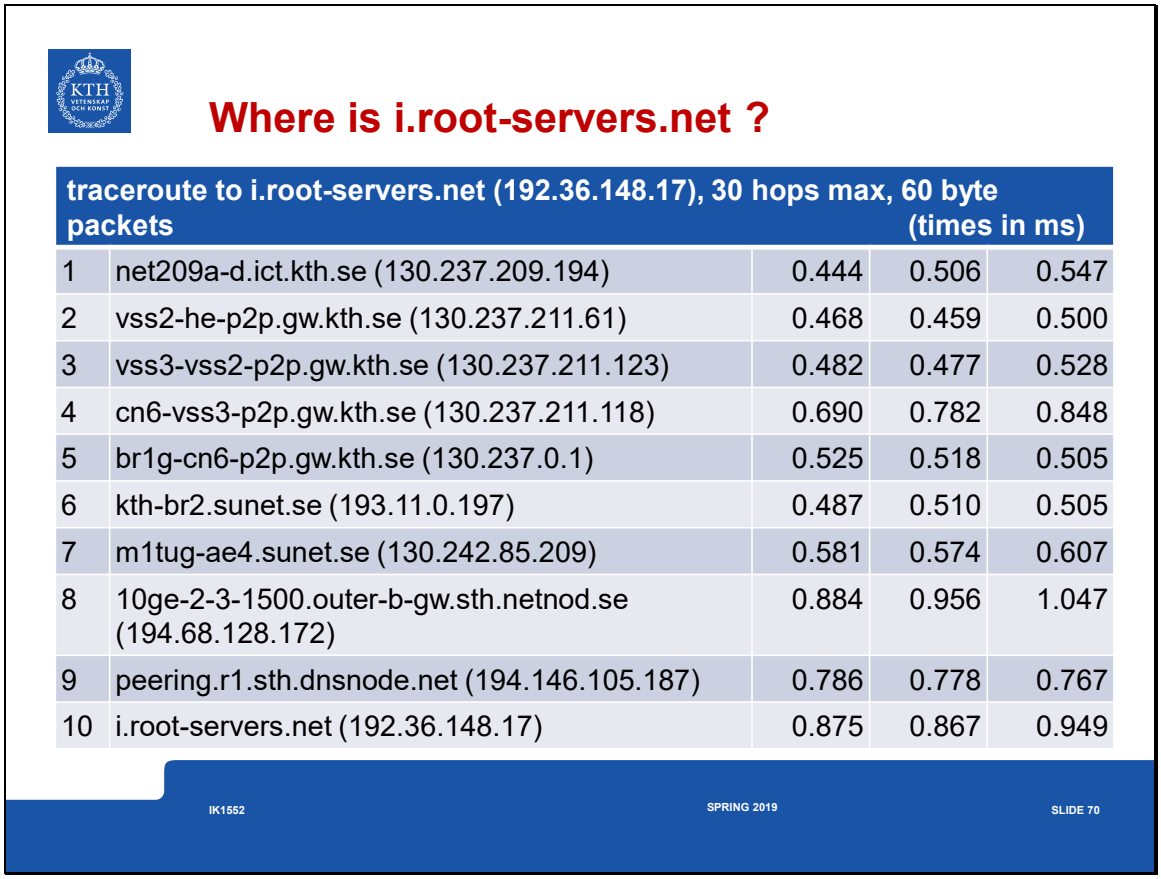


Where is f.root-servers.net ?

traceroute to f.root-servers.net (192.5.5.241), 30 hops max, 60 byte packets				(times in ms)
1	net209a-d.ict.kth.se (130.237.209.194)	0.572	0.664	0.701
2	vss2-he-p2p.gw.kth.se (130.237.211.61)	0.515	0.558	0.600
3	vss3-vss2-p2p.gw.kth.se (130.237.211.123)	0.459	0.461	0.485
4	cn6-vss3-p2p.gw.kth.se (130.237.211.118)	0.545	0.659	0.727
5	br1g-cn6-p2p.gw.kth.se (130.237.0.1)	0.495	0.484	0.472
6	kth-br2.sunet.se (193.11.0.197)	0.503	0.465	0.520
7	m1tug-ae4.sunet.se (130.242.85.209)	0.600	0.589	0.602
8	se-tug.nordu.net (109.105.102.17)	0.915	0.902	0.885
9	10gigabitethernet1-2.core1.sto1.he.net (194.68.123.187)	5.686	5.831	1.085
10	10ge12-6.core1.ams1.he.net (72.52.92.45)	26.074	25.954	25.943
11	amsix.r2.ams1.isc.org (80.249.208.140)	19.813	19.850	
11	isc-f-root-2.nikhef.nl-ix.net (193.239.116.113)			19.640
12	f.root-servers.net (192.5.5.241)	25.650	20.001	19.996

IK1552 SPRING 2019 SLIDE 69

Slide 70



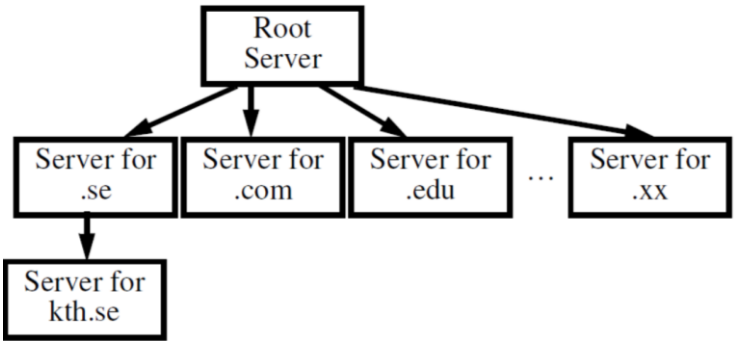
Where is i.root-servers.net ?

traceroute to i.root-servers.net (192.36.148.17), 30 hops max, 60 byte packets				(times in ms)
1	net209a-d.ict.kth.se (130.237.209.194)	0.444	0.506	0.547
2	vss2-he-p2p.gw.kth.se (130.237.211.61)	0.468	0.459	0.500
3	vss3-vss2-p2p.gw.kth.se (130.237.211.123)	0.482	0.477	0.528
4	cn6-vss3-p2p.gw.kth.se (130.237.211.118)	0.690	0.782	0.848
5	br1g-cn6-p2p.gw.kth.se (130.237.0.1)	0.525	0.518	0.505
6	kth-br2.sunet.se (193.11.0.197)	0.487	0.510	0.505
7	m1tug-ae4.sunet.se (130.242.85.209)	0.581	0.574	0.607
8	10ge-2-3-1500.outer-b-gw.sth.netnod.se (194.68.128.172)	0.884	0.956	1.047
9	peering.r1.sth.dnsnode.net (194.146.105.187)	0.786	0.778	0.767
10	i.root-servers.net (192.36.148.17)	0.875	0.867	0.949

IK1552 SPRING 2019 SLIDE 70

Slide 71

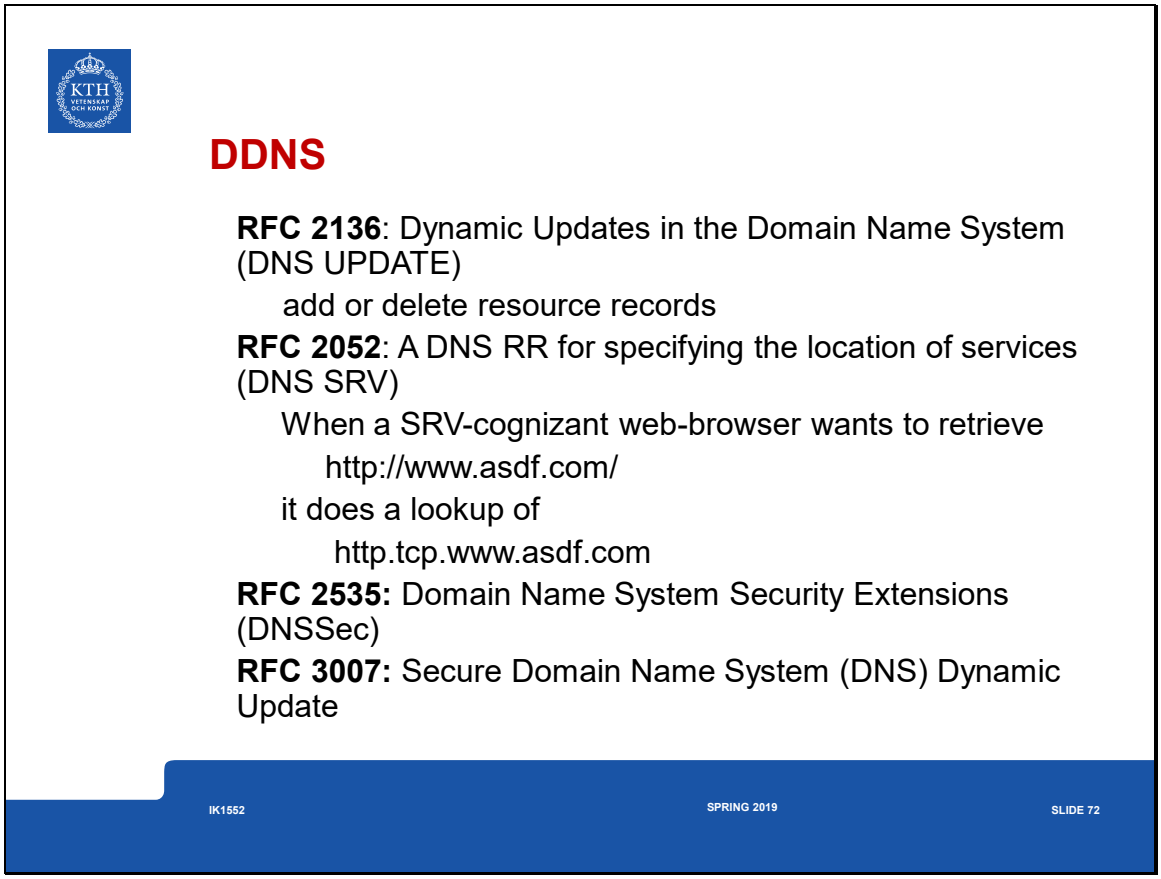
Dynamic Domain Name System (DDNS)



Host Name	IP-address
host_a	130.237.x.1
host_b	130.237.x.2
host_c	130.237.x.3
host_d	130.237.x.4
mobile1	c/o address <<< we can update this dynamically

IK1552 SPRING 2019 SLIDE 71

Slide 72



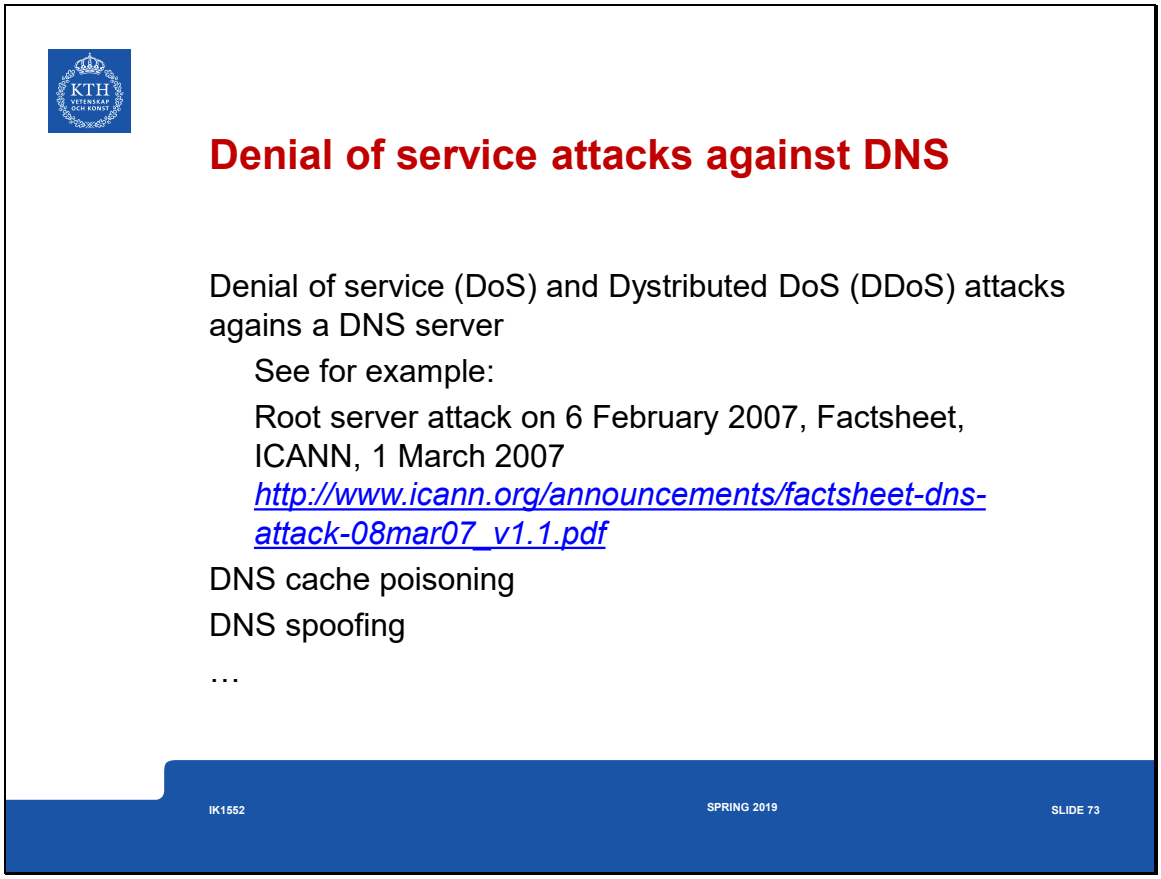
P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, 'Dynamic Updates in the Domain Name System (DNS UPDATE)', *Internet Request for Comments*, vol. RFC 2136 (Proposed Standard), Apr. 1997 [Online]. Available: http://www.rfc-editor.org/rfc/rfc2136.txt

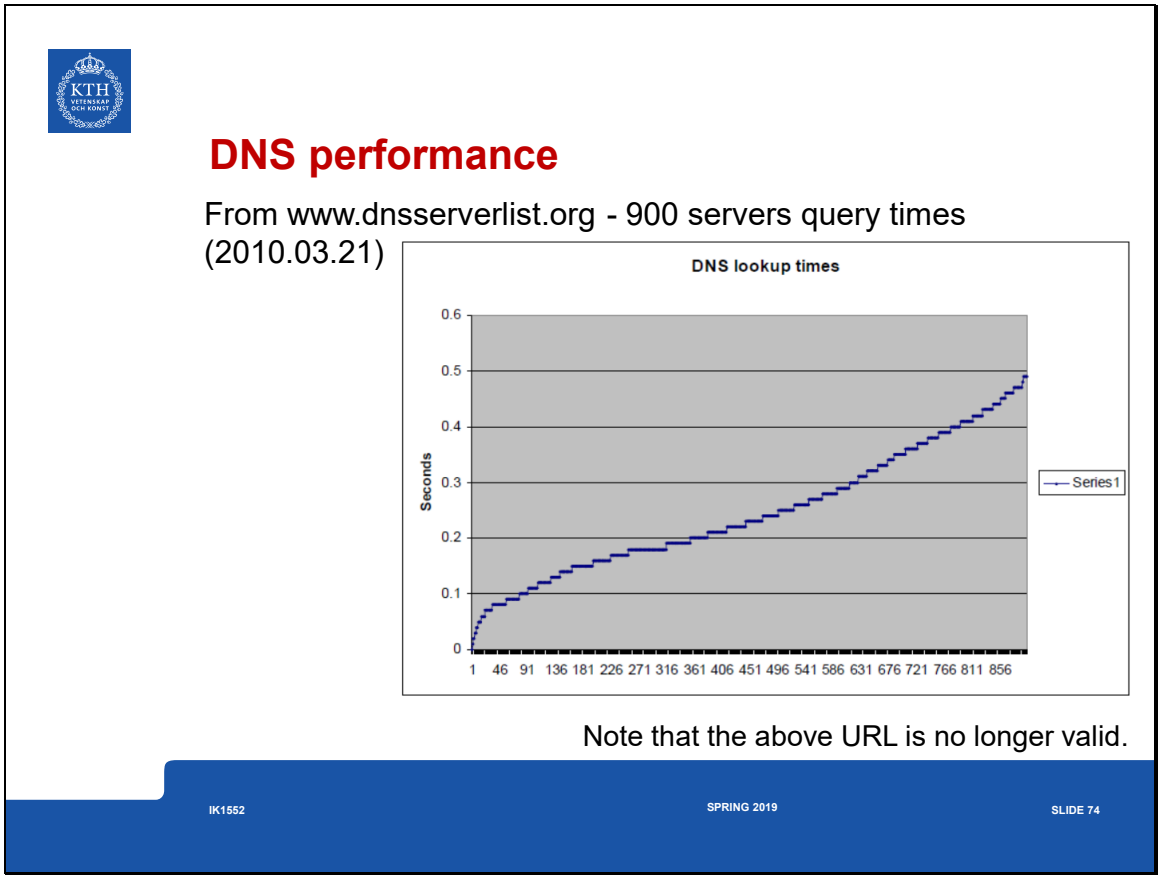
- Gulbrandsen and P. Vixie, 'A DNS RR for specifying the location of services (DNS SRV)', *Internet Request for Comments*, vol. RFC 2052 (Experimental), Oct. 1996 [Online]. Available: http://www.rfc-editor.org/rfc/rfc2052.txt

D. Eastlake 3rd, 'Domain Name System Security Extensions', *Internet Request for Comments*, vol. RFC 2535 (Proposed Standard), Mar. 1999 [Online]. Available: http://www.rfc-editor.org/rfc/rfc2535.txt

B. Wellington, 'Secure Domain Name System (DNS) Dynamic Update', *Internet Request for Comments*, vol. RFC 3007 (Proposed Standard), Nov. 2000 [Online]. Available: http://www.rfc-editor.org/rfc/rfc3007.txt

Slide 73


Denial of service attacks against DNS
Denial of service (DoS) and Dystributed DoS (DDoS) attacks agains a DNS server
See for example:
Root server attack on 6 February 2007, Factsheet, ICANN, 1 March 2007
http://www.icann.org/announcements/factsheet-dns-attack-08mar07_v1.1.pdf
DNS cache poisoning
DNS spoofing
…
IK1552
SPRING 2019
SLIDE 73



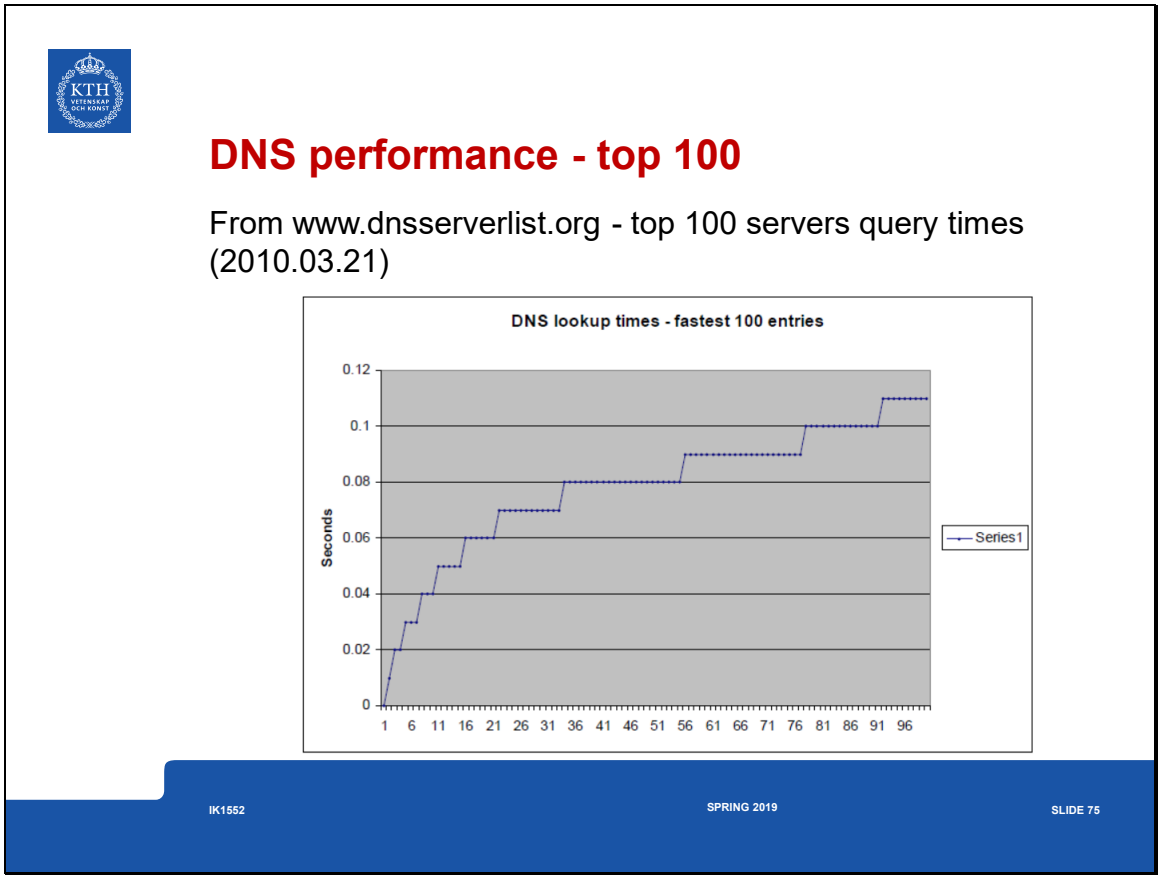
DNS performance

From www.dnsserverlist.org - 900 servers query times (2010.03.21)

Note that the above URL is no longer valid.

IK1552 SPRING 2019 SLIDE 74

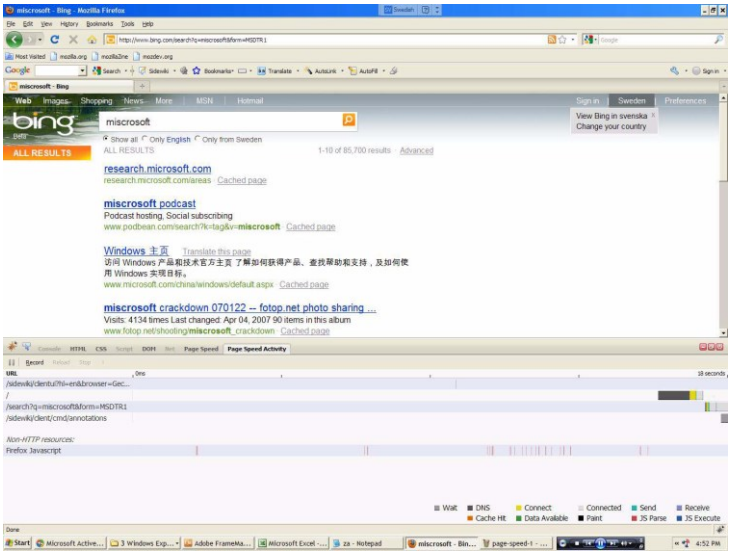
Slide 75


DNS performance - top 100
From www.dnsserverlist.org - top 100 servers query times (2010.03.21)
DNS lookup times - fastest 100 entries
Seconds
0.12
0.1
0.08
0.06
0.04
0.02
0
1 6 11 16 21 26 31 36 41 46 51 56 61 66 71 76 81 86 91 96
Series1
IK1552
SPRING 2019
SLIDE 75

Slide 76

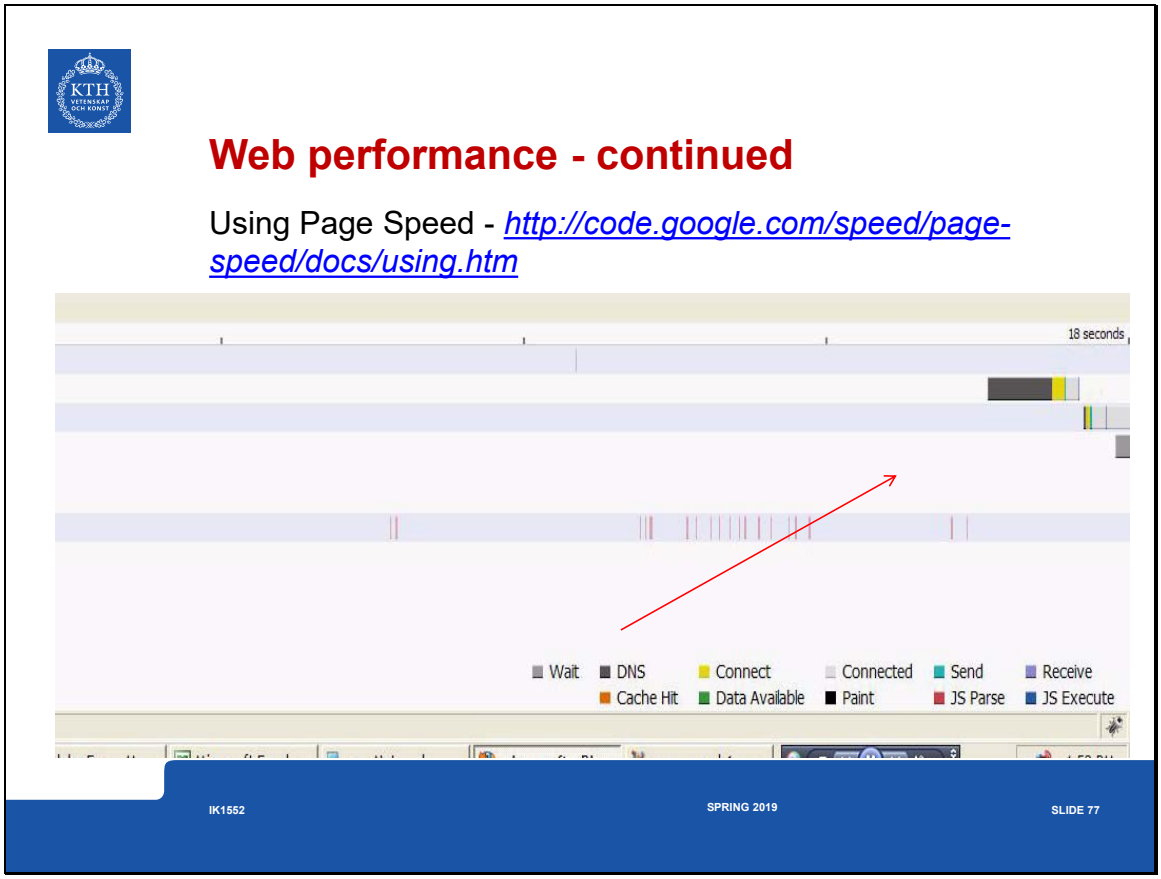
Web performance

Using Page Speed (formerly *http://code.google.com/speed/page-speed/docs/using.htm*)

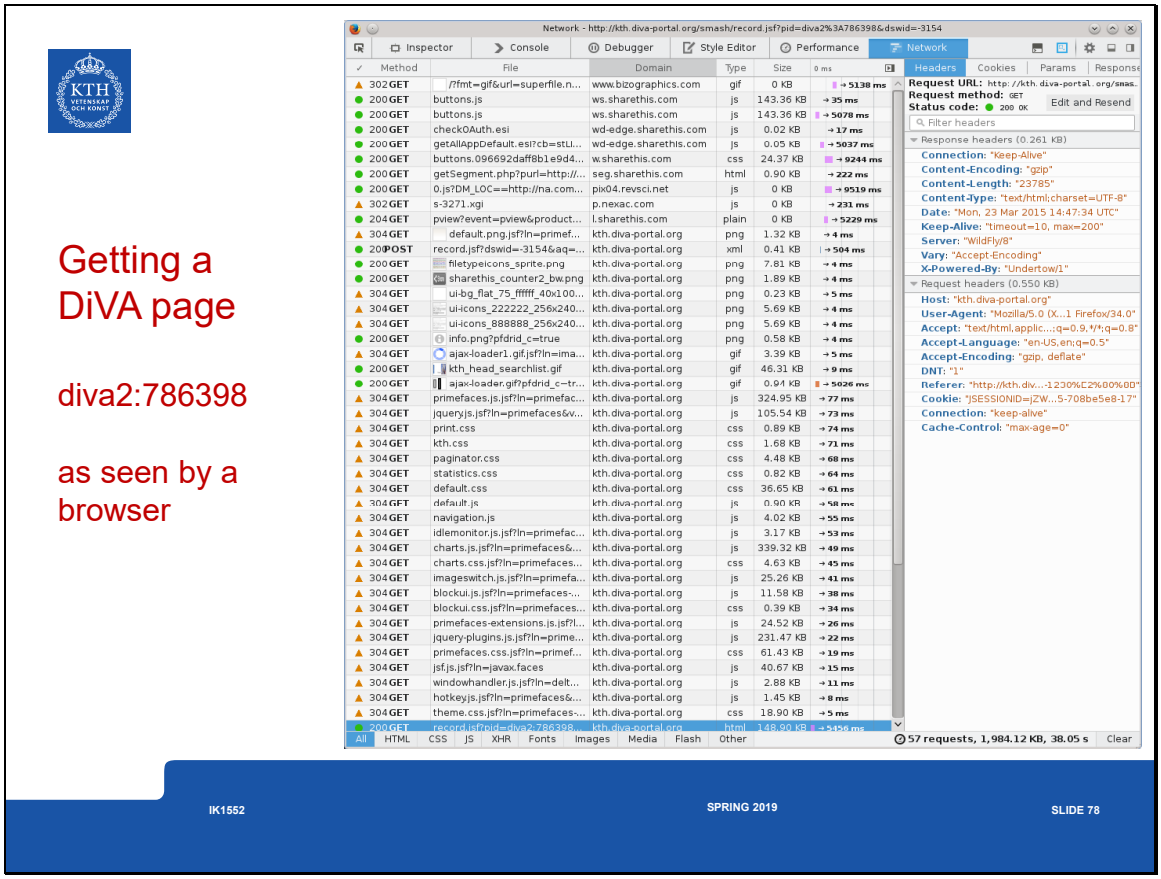


see https://developers.google.com/speed/pagespeed/

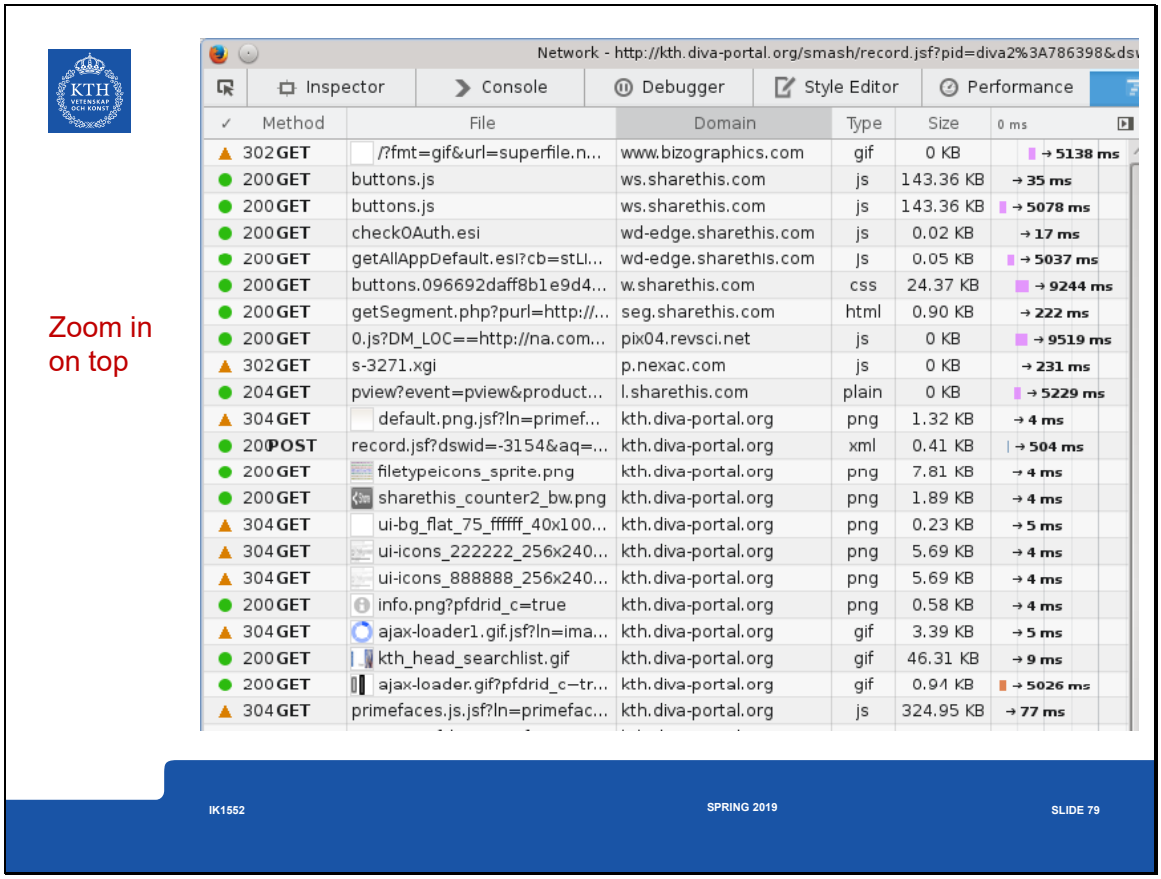
Slide 77


Web performance - continued
Using Page Speed - http://code.google.com/speed/page-speed/docs/using.htm
18 seconds
Wait
DNS
Connect
Connected
Send
Receive
Cache Hit
Data Available
Paint
JS Parse
JS Execute
IK1552
SPRING 2019
SLIDE 77

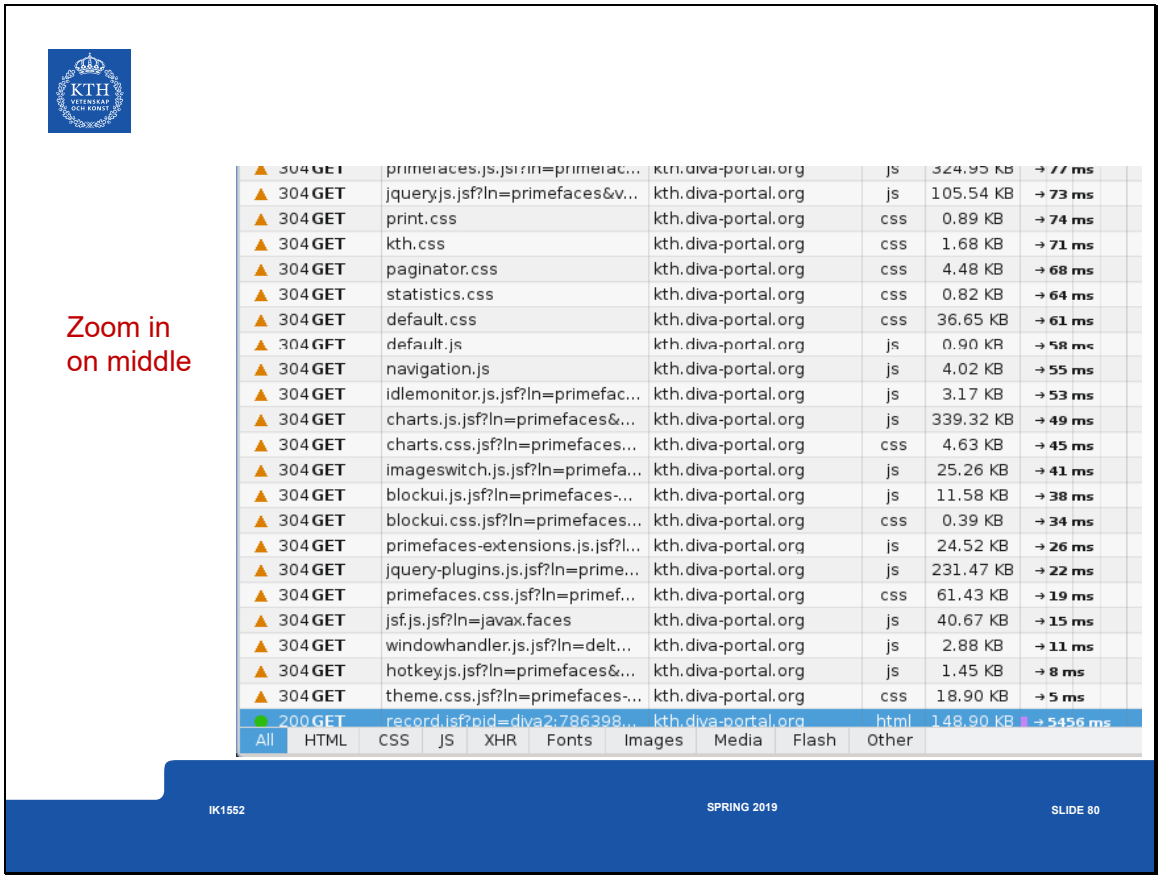
Slide 78


Getting a DiVA page
diva2:786398
as seen by a browser
Network - http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A786398&dswid=-3154
Inspector
Console
Debugger
Style Editor
Performance
Network
Method
File
Domain
Type
Size
Headers
Cookies
Params
Response
Request URL: http://kth.diva-portal.org/smas...
Request method: GET
Status code: 200 OK
Edit and Resend
Filter headers
Response headers (0.261 KB)
Connection: "Keep-Alive"
Content-Encoding: "gzip"
Content-Length: "23785"
Content-Type: "text/html;charset=UTF-8"
Date: "Mon, 23 Mar 2015 14:47:34 UTC"
Keep-Alive: "timeout=10, max=200"
Server: "WildFly/8"
Vary: "Accept-Encoding"
X-Powered-By: "Undertow/1"
Request headers (0.550 KB)
Host: "kth.diva-portal.org"
User-Agent: "Mozilla/5.0 (X...1 Firefox/34.0"
Accept: "text/html,applic...;q=0.9,*/*;q=0.8"
Accept-Language: "en-US,en;q=0.5"
Accept-Encoding: "gzip, deflate"
DNT: "1"
Referer: "http://kth.div...-123D%C2%80%8D"
Cookie: "JSESSIONID=jZW...5-708be5e8-17"
Connection: "keep-alive"
Cache-Control: "max-age=0"
All
HTML
CSS
JS
XHR
Fonts
Images
Media
Flash
Other
57 requests, 1,984.12 KB, 38.05 s
Clear
IK1552
SPRING 2019
SLIDE 78

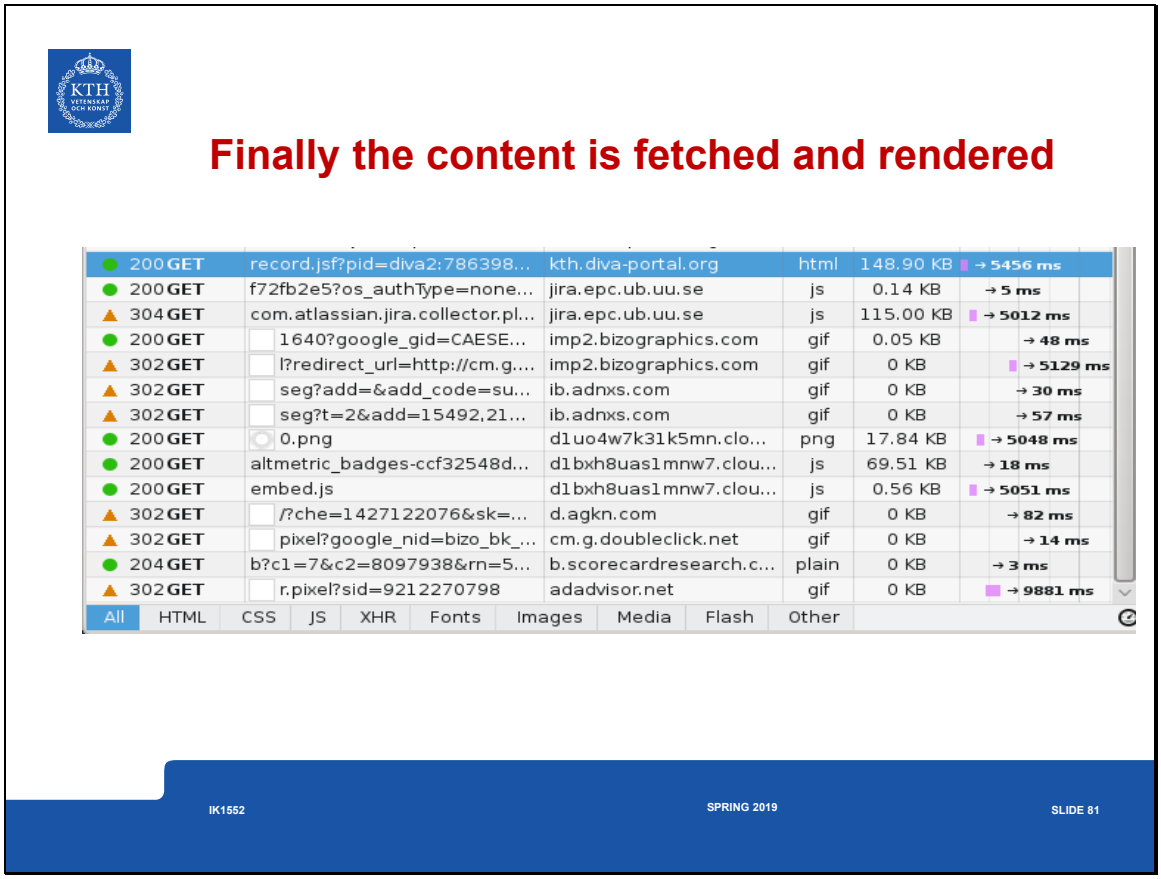
Slide 79


Zoom in on top
Network - http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A786398&dsv
Inspector
Console
Debugger
Style Editor
Performance
Method File Domain Type Size 0 ms
302 GET /?fmt=gif&url=superfile.n... www.bizographics.com gif 0 KB → 5138 ms
200 GET buttons.js ws.sharethis.com js 143.36 KB → 35 ms
200 GET buttons.js ws.sharethis.com js 143.36 KB → 5078 ms
200 GET checkOAuth.esi wd-edge.sharethis.com js 0.02 KB → 17 ms
200 GET getAllAppDefault.esi?cb=stLi... wd-edge.sharethis.com js 0.05 KB → 5037 ms
200 GET buttons.096692daff8b1e9d4... w.sharethis.com css 24.37 KB → 9244 ms
200 GET getSegment.php?purl=http://... seg.sharethis.com html 0.90 KB → 222 ms
200 GET 0.js?DM_LOC==http://na.com... pix04.revsci.net js 0 KB → 9519 ms
302 GET s-3271.xgi p.nexac.com js 0 KB → 231 ms
204 GET pview?event=pview&product... l.sharethis.com plain 0 KB → 5229 ms
304 GET default.png.jsf?ln=primef... kth.diva-portal.org png 1.32 KB → 4 ms
20 POST record.jsf?dswid=-3154&aq=... kth.diva-portal.org xml 0.41 KB → 504 ms
200 GET filetypeicons_sprite.png kth.diva-portal.org png 7.81 KB → 4 ms
200 GET sharethis_counter2_bw.png kth.diva-portal.org png 1.89 KB → 4 ms
304 GET ui-bg_flat_75_ffffff_40x100... kth.diva-portal.org png 0.23 KB → 5 ms
304 GET ui-icons_222222_256x240... kth.diva-portal.org png 5.69 KB → 4 ms
304 GET ui-icons_888888_256x240... kth.diva-portal.org png 5.69 KB → 4 ms
200 GET info.png?pfdrid_c=true kth.diva-portal.org png 0.58 KB → 4 ms
304 GET ajax-loader1.gif.jsf?ln=ima... kth.diva-portal.org gif 3.39 KB → 5 ms
200 GET kth_head_searchlist.gif kth.diva-portal.org gif 46.31 KB → 9 ms
200 GET ajax-loader.gif?pfdrid_c=tr... kth.diva-portal.org gif 0.94 KB → 5026 ms
304 GET primefaces.js.jsf?ln=primefac... kth.diva-portal.org js 324.95 KB → 77 ms
IK1552
SPRING 2019
SLIDE 79

Slide 80


Zoom in
on middle
304 GET primefaces.js.jsf?ln=primefac... kth.diva-portal.org js 324.95 KB → 77 ms
304 GET jquery.js.jsf?ln=primefaces&v... kth.diva-portal.org js 105.54 KB → 73 ms
304 GET print.css kth.diva-portal.org css 0.89 KB → 74 ms
304 GET kth.css kth.diva-portal.org css 1.68 KB → 71 ms
304 GET paginator.css kth.diva-portal.org css 4.48 KB → 68 ms
304 GET statistics.css kth.diva-portal.org css 0.82 KB → 64 ms
304 GET default.css kth.diva-portal.org css 36.65 KB → 61 ms
304 GET default.js kth.diva-portal.org js 0.90 KB → 58 ms
304 GET navigation.js kth.diva-portal.org js 4.02 KB → 55 ms
304 GET idlemonitor.js.jsf?ln=primefac... kth.diva-portal.org js 3.17 KB → 53 ms
304 GET charts.js.jsf?ln=primefaces&... kth.diva-portal.org js 339.32 KB → 49 ms
304 GET charts.css.jsf?ln=primefaces... kth.diva-portal.org css 4.63 KB → 45 ms
304 GET imageswitch.js.jsf?ln=primefa... kth.diva-portal.org js 25.26 KB → 41 ms
304 GET blockui.js.jsf?ln=primefaces-... kth.diva-portal.org js 11.58 KB → 38 ms
304 GET blockui.css.jsf?ln=primefaces... kth.diva-portal.org css 0.39 KB → 34 ms
304 GET primefaces-extensions.js.jsf?l... kth.diva-portal.org js 24.52 KB → 26 ms
304 GET jquery-plugins.js.jsf?ln=prime... kth.diva-portal.org js 231.47 KB → 22 ms
304 GET primefaces.css.jsf?ln=primef... kth.diva-portal.org css 61.43 KB → 19 ms
304 GET jsf.js.jsf?ln=javax.faces kth.diva-portal.org js 40.67 KB → 15 ms
304 GET windowhandler.js.jsf?ln=delt... kth.diva-portal.org js 2.88 KB → 11 ms
304 GET hotkey.js.jsf?ln=primefaces&... kth.diva-portal.org js 1.45 KB → 8 ms
304 GET theme.css.jsf?ln=primefaces-... kth.diva-portal.org css 18.90 KB → 5 ms
200 GET record.jsf?pid=diva2:786398... kth.diva-portal.org html 148.90 KB → 5456 ms
All HTML CSS JS XHR Fonts Images Media Flash Other
IK1552
SPRING 2019
SLIDE 80

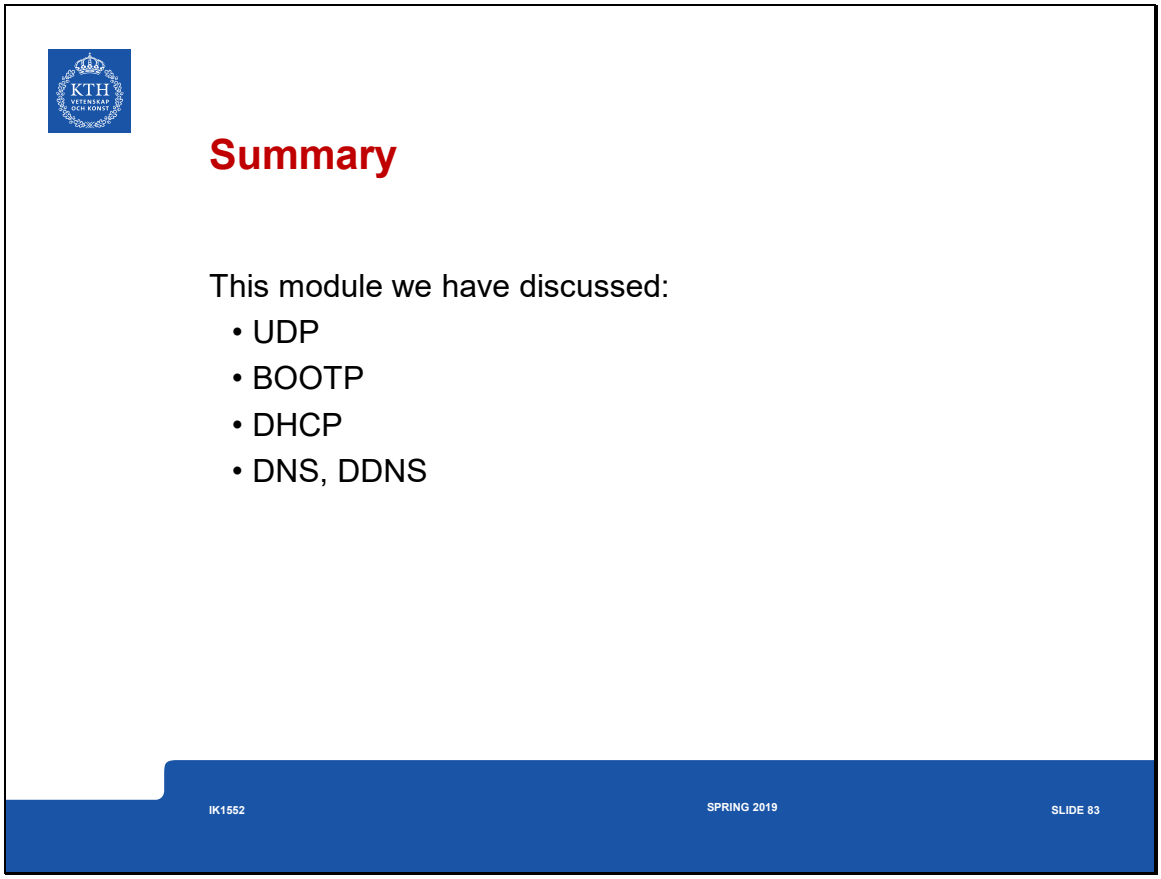
Slide 81


Finally the content is fetched and rendered
200 GET record.jsf?pid=diva2:786398... kth.diva-portal.org html 148.90 KB → 5456 ms
200 GET f72fb2e5?os_authType=none... jira.epc.ub.uu.se js 0.14 KB → 5 ms
304 GET com.atlassian.jira.collector.pl... jira.epc.ub.uu.se js 115.00 KB → 5012 ms
200 GET 1640?google_gid=CAESE... imp2.bizographics.com gif 0.05 KB → 48 ms
302 GET l?redirect_url=http://cm.g.... imp2.bizographics.com gif 0 KB → 5129 ms
302 GET seg?add=&add_code=su... ib.adnxs.com gif 0 KB → 30 ms
302 GET seg?t=2&add=15492,21... ib.adnxs.com gif 0 KB → 57 ms
200 GET 0.png d1uo4w7k31k5mn.clo... png 17.84 KB → 5048 ms
200 GET altmetric_badges-ccf32548d... d1bxh8uas1mnw7.clou... js 69.51 KB → 18 ms
200 GET embed.js d1bxh8uas1mnw7.clou... js 0.56 KB → 5051 ms
302 GET /?che=1427122076&sk=... d.agkn.com gif 0 KB → 82 ms
302 GET pixel?google_nid=bizo_bk_... cm.g.doubleclick.net gif 0 KB → 14 ms
204 GET b?c1=7&c2=8097938&rn=5... b.scorecardresearch.c... plain 0 KB → 3 ms
302 GET r.pixel?sid=9212270798 adadvisor.net gif 0 KB → 9881 ms
All HTML CSS JS XHR Fonts Images Media Flash Other
IK1552
SPRING 2019
SLIDE 81

Slide 82


Public/Commercial DNS services
A number of companies are providing DNS servers to speed up DNS lookups:
• Google Public DNS
http://code.google.com/speed/public-dns/
• OpenDNS
http://www.opendns.com/
• Neustar DNS Advantage
https://www.neustar.biz/services/dns-services/dns-advantage-free-recursive-dns
…
IK1552
SPRING 2019
SLIDE 82

Slide 83


Summary
This module we have discussed:
• UDP
• BOOTP
• DHCP
• DNS, DDNS
IK1552
SPRING 2019
SLIDE 83

Slide 84