IPv6

Basics & Usage

A mini talk

Sebastian Werner

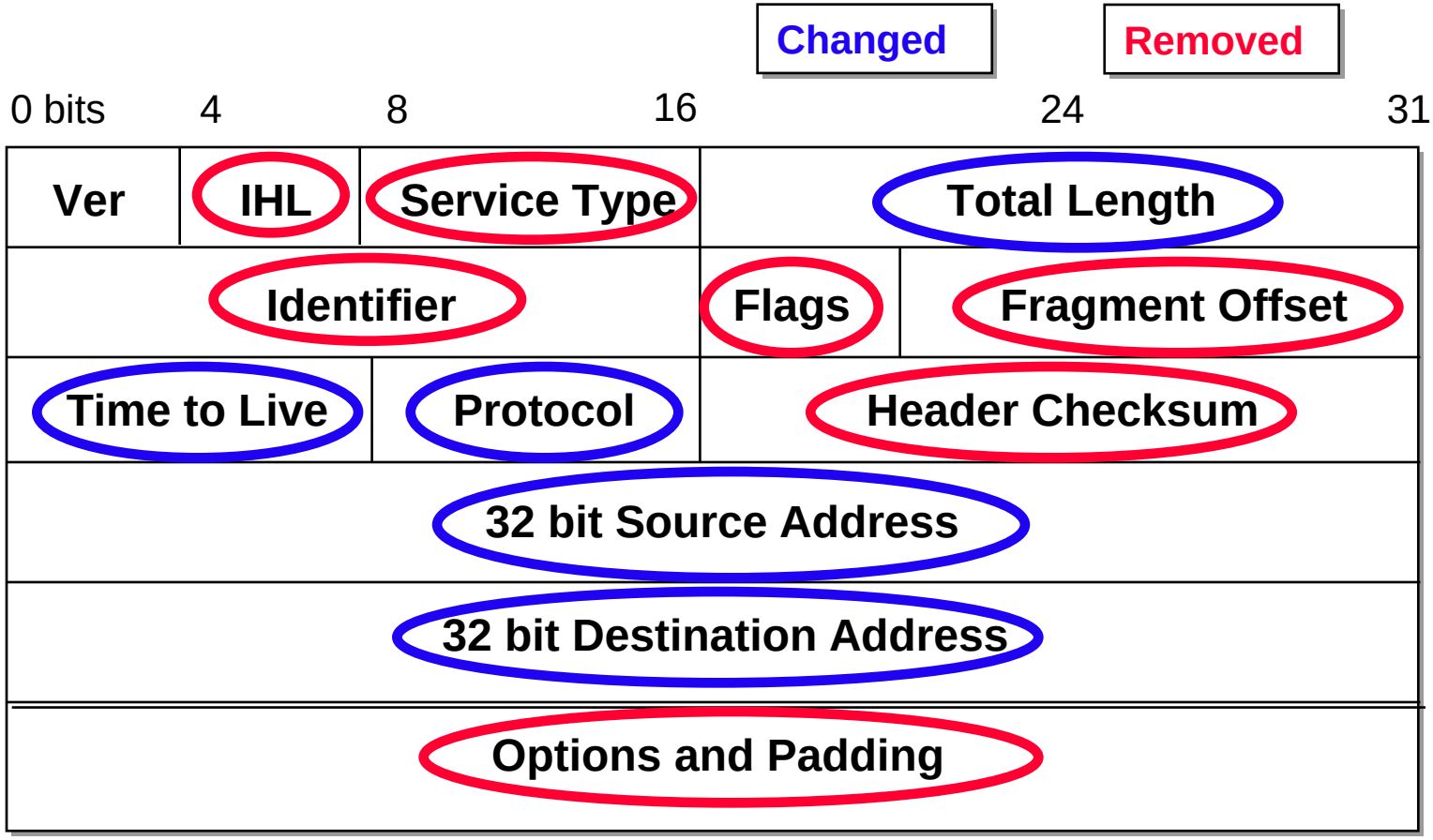
blackwing@ccc.de

Topics

- Short review of IPv6 standard
- Address scheme & autoconfig
- Current v6 Stacks
- DNS
- Firewalling pitfalls
- Get connected!
- What next?

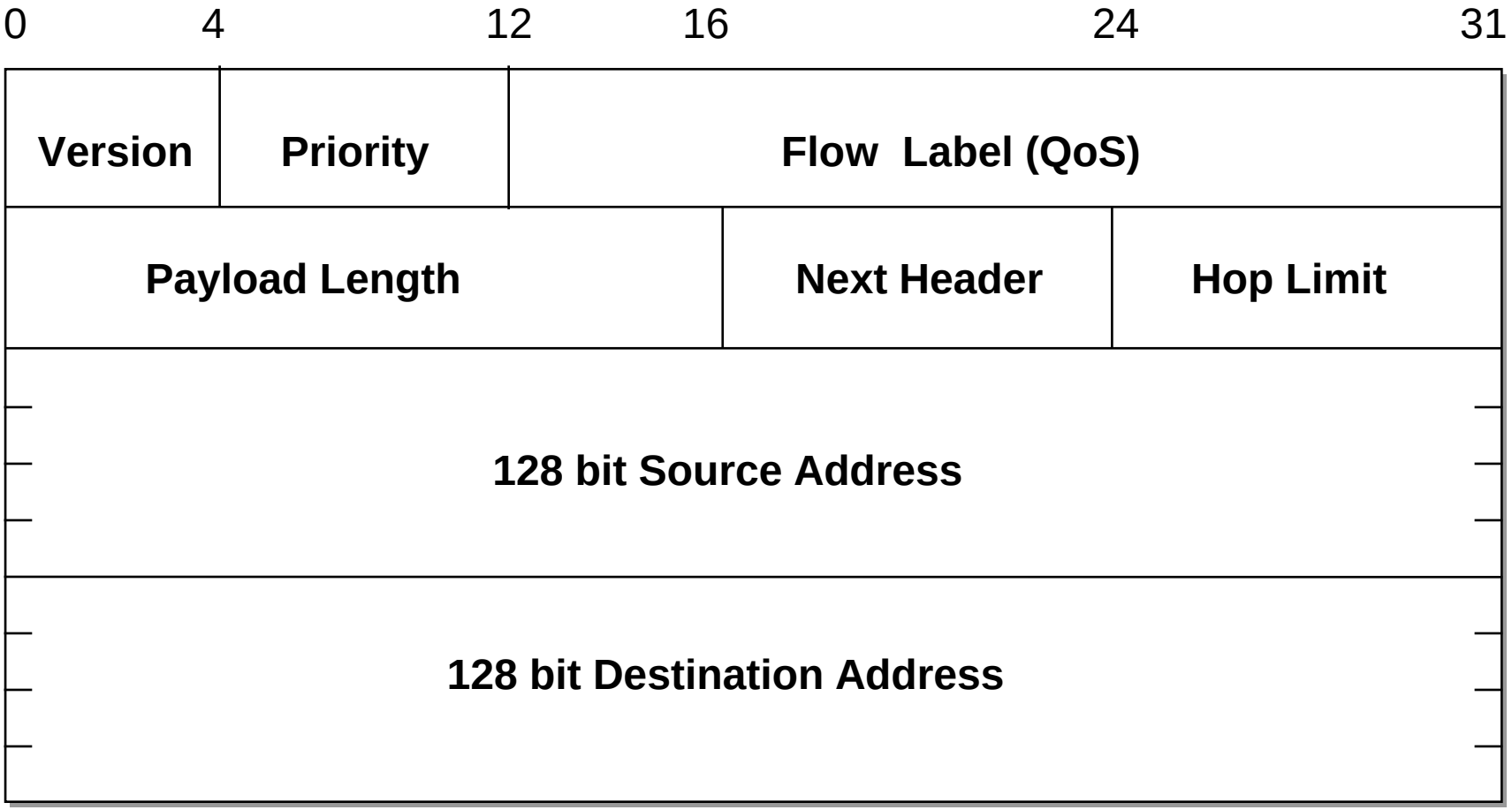
IPv4 Header

20 octets + options : 13 fields

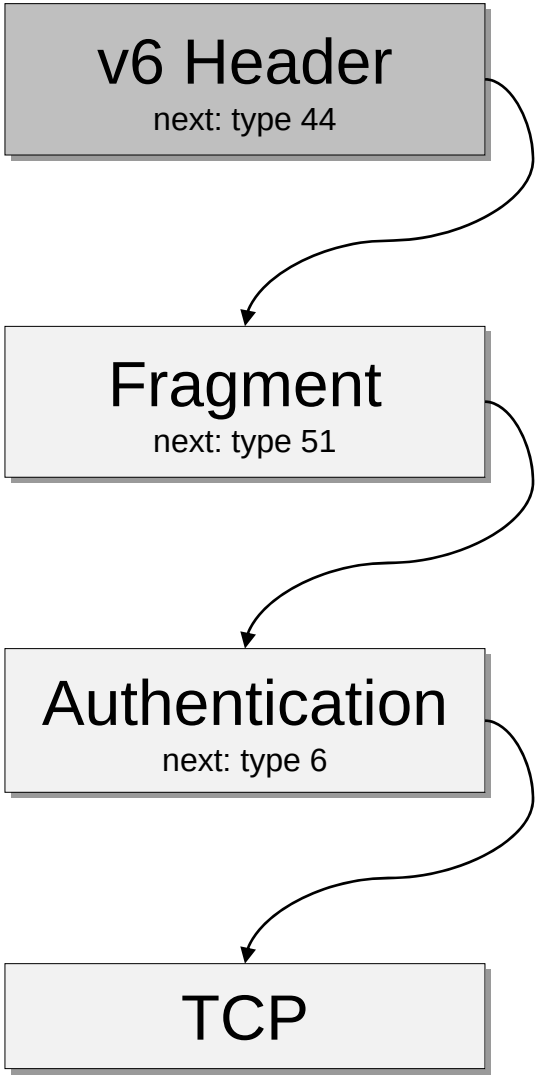


IPv6 Header

40 Octets, 8 fields



Extension Headers



- Basic v6 header very short
- Extra options via extension headers:
  - Hop-Options *0*
  - Routing *43*
  - Fragment *44*
  - Authentication (IPSec Phase 2) *51*
  - ESP (IPSec Phase 1) *50*
  - Destination Options *60*
- Next-Layer also via this field
  - TCP *6*
  - UDP *17*
  - ICMPv6 *58*
- Problem: Hard to parse / firewall.

See /etc/protocol or Wikipedia → Protokoll (IP) for a complete list

Special stuff to keep in mind

- Payload lenght: 16bit – Size in Octets.
  - Means: Maximum of 64kbyte *(-1 byte)* per Packet
  - BUT: Jumbo packets via Hop-Options: Up to: 4GB/packet
- Flow Labels
  - Intended for Quality of Services Use
  - Idea: Once assign Flow-ID, Always go same path (like MPLS)
  - Currently unused
  - Discussion of use for Flow-Routing
- Extension headers
  - Used to do Mobile IP
  - Might f*ck up a router with 10mbit/s ☺
  - Sometimes you might bypass firewalls…
  - Most routers just drop those packets – Because of that
  - Same story as for IPv4
- Routers do not do any fragmentation! (PMTU discovery)

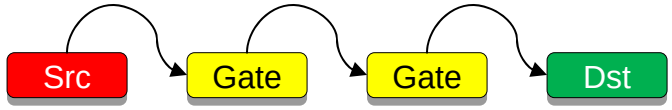
Benefits of IPv6 Addresses

- stable, unique addresses for all devices
  - note: stable does not mean permanent!
  - allow continued growth of the Internet (for centuries to come)
  - restore end-to-end transparency of the Internet
- additional benefits:
  - plug-and-play (no need for configuration servers)
  - verifiable end-to-end packet integrity (no need for NATs)
  - simpler mobility (no need for “foreign agent” function)
  - integrated security / crypto via IPSec (no more need for SSL!)
  - no fragmentation

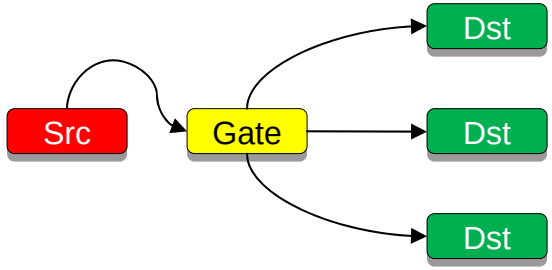
    (end to end fragmentation – should reduce delays)

General address types

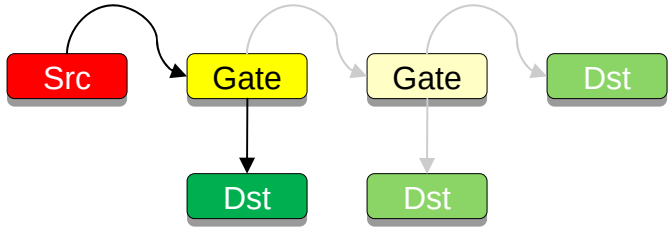
- Unicast adresses
  - `2000::/3` is generally routable
  - `2002::/16` is 6to4 deployment
  - `3ffe::/16` was the 6bone testbed networking space (deprecated)



- Multicast adresses – *One to many*
  - `ff00::/8`
    - `ff01::/12` – node local
    - `ff02::/12` – link local
    - `ff05::/12` – site local
    - `ff0e::/12` – global

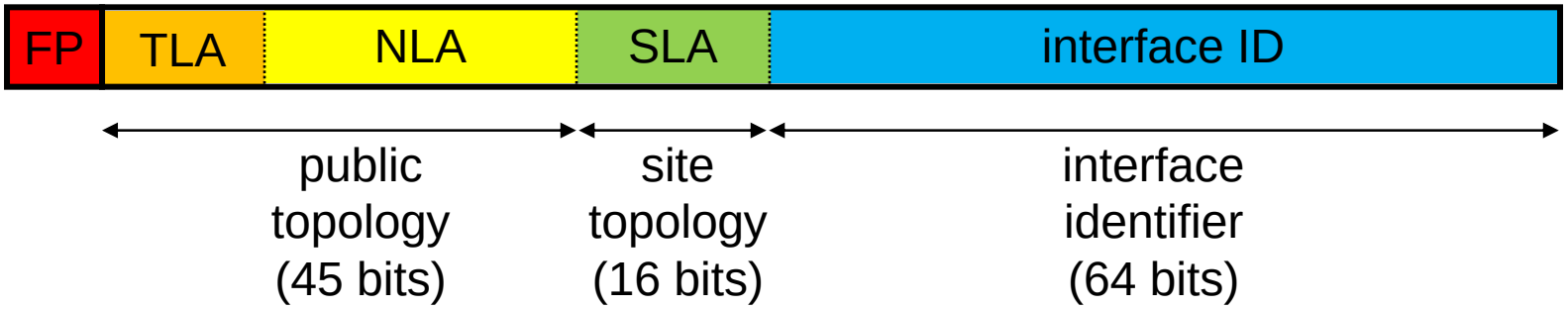


- Anycast adresses – *One to nearest*
  - `2001:0000::/32`



RFC2372

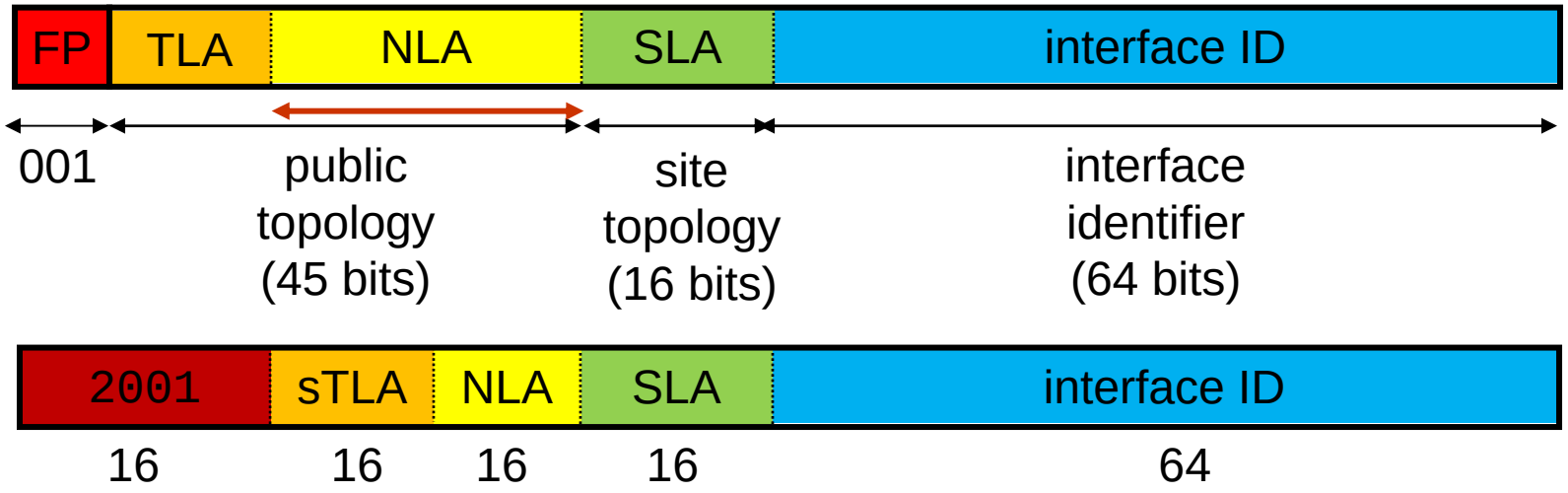
Global Unicast Addresses



FP = Format Prefix (001)
TLA = Top-Level Aggregator
NLA = Next-Level Aggregator(s)
SLA = Site-level Aggregator

- Basic idea: Aggregate by topology to minimize routing table.
- This structure showed to be a moving target
- Aim is good aggregation and flexibility

Global Unicast Address Formats



Example:

RIPE (Regional Registry) got `2001:600::/23`
aggregates European providers

DFN (Top-Level Provider) got `2001:638::/32`
aggregates universities in .de

My former university got `2001:638:a00::/48`
can still assign Site-level!

http://www.iana.org/assignments/ipv6-unicast-address-assignments/

IPv6 Host Address

Formed from a combination of the:

Prefix **Interface ID**

2001	0638	0a00	cafe	02A0	C9FF	FE43	95A7

Prefix Representation 2001:638:a00:cafe:/64

Node EUI64 address 02:A0:C9FF:FE43:95:A7
MAC Address 00-A0-C9 - 43-95-A7

Separation of “who you are” from “where you are connected to”

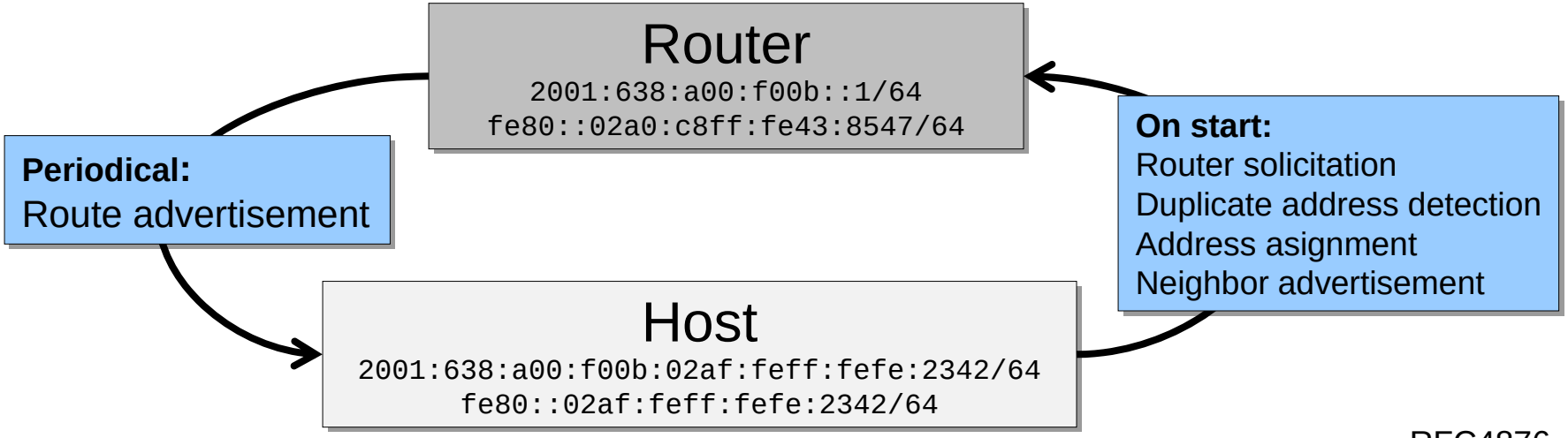
- Prefix: Routing topology
- Interface ID: Node Identifier (MAC address)
- EUI64: Global/Universal Bit 2nd lowest bit of 1st byte. Typical: 00 → 02
  Pad with FF:FE in middle.

Neighbor Discovery Protocol (RFC2461)

- Used to determine physical adresses of neighbors on local link – communication via multicast
- All hosts have
  - Neighbor cache („ARP“ table – link layer to local adress)
  - Destination cache (Which is next-hop for where?)
  - Prefix list (Which prefixes are on-link?)
  - Default router list
- Defines ICMPv6 types
  - **133**: Router solicitation – Request „Hello“ from all routers
  - **134**: Router advertisement
  - **135**: Neighbor solicitation
    (sent to ethernet multicast 33:33:FF:*xx:yy:zz*)
  - **136**: Neighbor advertisement
  - **137**: Redirect – Update route at a client by a router

Autoconfiguration

- Host generates EUI64 part via MAC address
- Assigns a site-local address (fe80::*$EUI64*) per link
- Router send route advertisements that propagate the prefix
- Host assigns *`$PREFIX:$EUI64`* to interface and adds route: *default via `fe80::$ROUTER_EUI64`*
- Neighbor advertisements ensure *IP6 → MAC* matching
- Neighbor detection also ensures collision-free IP6 adresses (no double assignments)



RFC4876

Current v6 stacks config

- All do autoconfig, all suck in IPSEC
- Windows(cmd line)
  - `ipv6 install` (XP only - Vista & Win7 have it already)
  - `netsh interface ipv6 set privacy disabled persistent`
- FreeBSD
  - Built Kernel with `INET6`
  - `sysctl -w net.inet6.ip6.accept_rtadv=1`
  - `rtsol`
- Linux
  - V6 Kernel → `modprobe ipv6`
  - `echo 1 > /proc/sys/net/ipv6/conf/eth0/autoconf`

Linux v6 config & diagnosis

- Please use iproute2 tools
  omit ioctl stuff (`route, ifconfig, arp`)
- Check for v6 routes: `ip -6 route show`
- Add route `ip -6 route add default via $IP`
- List v6 adresses: `ip -6 address show`
- Add static IP: `ip -6 addr add $IP/64 dev eth0`
- List neighbors: `ip -6 neighbor show`
- Debug fuckups: `tcpdump ip6 …`
- Scan hosts: `nmap -6 $DESTINATION`
- Path mtu: `tracepath6 $DESTINATION`
- Forward traceroute: `traceroute6 or mtr`
- Reverse traceroute: *not yet available / implemented…*
  RFC4561 - Definition of a Record Route Object (RRO) Node-Id Sub

IPv6 DNS

- DNS works with *types* in requests: `ANY, A, PTR, TXT ..`
- Usually: no need to change `resolv.conf`

  You can also specificy nameserver in v6 there
- `nsswitch.conf` – check for hosts: dns6
- Change in IPv6: Use AAAA instead of A!

```
$ dig garantiert.net ANY
;; QUESTION SECTION:
;garantiert.net.                IN      ANY
;; ANSWER SECTION:
garantiert.net.     3600    IN      SOA  garantiert.net …
…
garantiert.net.     3600    IN      A    85.131.211.34
garantiert.net.     3600    IN      AAAA 2001:4d50:100:5::2
…
```

DNS server config for v6

- Config is straighforward
- Forward lookups are easy: e.g. bind .zone:

```
$TTL 86400
bla.foobar.de. AAAA
    2001:6f8:93d:3b:45f6:1342:1387:2ffe
```

djbdns

```
6bla.foobar.de:200106f8093d03b45f6134213872ffe:86400
```

- Reverse lookup via *ip6.int* and *ip6.arpa* domains
- Like v4 reverse – PTR records:

```
$ host 2001:4d50:100:5::2
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.5.0.0.0.0.0.1.0.0.5.d.4
.1.0.0.2.ip6.arpa domain name pointer garantiert.net.
```

IPv6 firewalling & routing

- Start thinking about security NOW.
- There are very few policies in act.
- Faking source adress is easy nowadays.
- Most services listen at * and also v6!

- **USE FIREWALL. I MEAN IT!**

- Routing: Enable v6 forwarding:
  ```
  # echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
  ```
- Install `radvd` for route advertisements

Recommended v6 ruleset

- Use `ip6tables` as known by `iptables`.

```
-P INPUT DROP
-P FORWARD DROP
-A INPUT   -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT --protocol icmpv6 --icmpv6-type echo-request -j ACCEPT
-A INPUT -p tcp --destination-port ssh  -m state --state NEW –j
   ACCEPT
-A OUTPUT –m state –state NEW –j ACCEPT
```

- For each internal host:

```
-A FORWARD –in-interface $internal –ext-interface $external –
   source $IP  -m state --state NEW -j ACCEPT
```

- Please check via `nmap` or whatever else.

Getting connected

- In some cases native v6 is available (university, data center)
  - Go ahead, bug them.
- DSL case: Native v6 connectivity via pppv6 session:
  http://www.sixxs.net/faq/connectivity/?faq=native&country=us
- Tunneled IPv6 for dynamic & static hosts
  - Sixxs provides free tunnels to get v6 connectivity

Whats next?

- Check your connectivity!
  - traceroute6, tracepath6
- Enable your services
- Use v6 only services
  - Irc
  - News (alt.binaries !!!)
- 6to4 – How to interconnect
- IPSec – A never-ending story of failures
- Getting your applications ready

Gory details

- S. Hagen: IPv6 Essentials, O'Reilly, 2006
- D. Malone & N.R. Murphy: IPv6 Network Administration, O'Reilly, 2005
- German Linux IPv6 Howto, 2009 & English Version
- IPv6 Beginners Howto

There are quite many books… But most are from vendors (Cisco, Microsoft etc) and quite focused.

Check your library or ask me. Got some in stock ☺

Thanks for your attention!

Questions or Suggestions?!