Labcourse “Routerlab”

Internet Protocol Version 6 (IPv6)
IPv4 Shortcomings

• IPv4 addresses have 32 bits only
  – not enough for 1 IP address per person
  – dynamic IPs, NAT, …

• Manual configuration
  – time consuming (in larger networks)
  – error-prone (wrong addresses, duplicates, …)

• IPv4 header format
  – variable length header (option field)
  – inefficient to parse if IP options present
IP Next Generation = IPv6

• New layer 3 protocol
• Key changes
  – 128 bit address length (vs. 32 bit)
  – Autoconfiguration
  – Restructured / optimized layer 3 headers
  – IPSEC security layer
  – Mobile IP(v6)
• But: all basic principles stay the same
IPv6 Benefits: Address Format

• 32 bits in IPv4 ↔ 128 bits in IPv6

• IPv4
  – 4 x 8 bits, decimal notation, separated by "." 

• IPv6
  – 8 x 16 bits, hexadecimal, separated by "::"
  – Drop leading zeroes (:0123:0001' = ':123:1')
  – Only one series of zeroes can be reduced to '::'
  – Examples:
    • 2001:200:0:8002:203:47ff:fea4:3085
    • 2001:608::2
    • fe80::210:60ff:fe80:3a16
IPv6 Address delegation: Hierarchy

- Bigger networks, *fixed size* assignments
  - Providers receive /19../32 network blocks
  - Every customer receives a /48 network block
  - Every LAN uses a /64 network
  - Inside LAN: 64 bit host part = "interface ID"

- Right now: Only allocations from p=001
  - 2xxx:: and 3xxx:::)

<table>
<thead>
<tr>
<th>p</th>
<th>LIR-Alloc</th>
<th>NLA</th>
<th>SLA</th>
<th>Interface-ID 64 Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>32</td>
<td>48</td>
<td>64</td>
</tr>
</tbody>
</table>
IPv6 Routing

• Forwarding / routing table lookup: similar to IPv4
• Same basic rule: "most specific wins"
  – 2001:608:b:1::/64
  – 2001:608:b::/48
• Default route is 0::0/0
• Routing protocols (BGP, OSPF) and routing table buildup follow same principles as IPv4
IPv6 Benefits: Autoconfiguration

• *Every* link uses fe80::/64 for link-local stuff
  – Hosts in isolated networks automagically communicate

• Router can announce global addresses
  – Router Advertisement (RA) ICMP packets
  – e.g., 2001:608:4:0::/64)

• Clients will use *all* available /64 prefixes
  – Compute the host part from their MAC address
  – EUI-64: Algorithm for computing 64-bit host part from 48-bit (Ethernet) MAC address
EUI-64 Autoconfiguration Example

- MAC address: 00:10:60:80:3A:16
- Link-local prefix fe80::64
- Router advertises prefix 2001:608:4:0::/64
- MAC converted to host part of IPv6 address
  - 00:10:60:80:3A:16 → ::210:60ff:fe80:3a16
  - Append this to all (!) prefixes
- Resulting interface configuration

  eth0 Link encap: Ethernet HWaddr 00:10:60:80:3A:16
  inet6 addr: 2001:608:4:0:210:60ff:fe80:3a16/64 Scope:Global
  inet6 addr: fe80::210:60ff:fe80:3a16/64 Scope:Link
IPv6 Addresses frequently seen

- "local" addresses
  - fe80::/64 link-local addresses
- "global" addresses
  - 2001:: early IPv6 production networks
  - 2002:IPv4::/48 6to4 migration method
  - ff0x:: global multicast address ranges
# Ipv4 vs. IPv6 header

## IPv6 header

<table>
<thead>
<tr>
<th>Version</th>
<th>Priority</th>
<th>Flow Label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload Length</td>
<td>Next Header</td>
<td>Hop Limit</td>
</tr>
</tbody>
</table>

| Source Address |

| Destination Address |

## IPv4 header

<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>Type of Service</th>
<th>Total Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td>Flags</td>
<td>Fragment Offset</td>
<td></td>
</tr>
<tr>
<td>Time to Live</td>
<td>Protocol</td>
<td>Header Checksum</td>
<td></td>
</tr>
</tbody>
</table>

| Source Address |

| Destination Address |

<table>
<thead>
<tr>
<th>Options</th>
<th>Padding</th>
</tr>
</thead>
</table>
Migration towards IPv6

Problems
- v4 host wanting to talk to v6 host
- v6 networks that are only connected by v4 infrastructure

- Migration techniques:
  - Dual-stacked hosts/router (v4+v6 IP stack on same machine)
  - Dual-stacked proxies / application-level gateways
  - Tunneling
    - Manually configured tunnels
    - Automatic tunneling (6to4, ISATAP, Teredo)
    - Tunnels configured by tunnel broker
Dual Stack

- Application
- TCP
- UDP
- IPv4
- IPv6

Frame Protocol ID
- 0x0800
- 0x86dd

Data Link (Ethernet)
6to4 IPv6 Addresses

- Converts IPv4 to hex and integrates into 6to4 IPv6 address
- Gives a /48 prefix to attached IPv6 networks
6to4 Tunneling

- Tunnel automatically created by dual-stacked router
6to4 Relays

Routing protocol advertizes 192.88.99.0/24

IPv4-side: 192.88.99.1
IPv6-side: 2001:db8:5000:3b/64

IPv6 data -> IPv6 data -> IPv4 -> IPv6 data
6to4 Relays (Reverse Direction)

- 192.88.99.1: Anycast address

IPv4/IPv6 Dual-Stack Internet

IPv4-side: 192.88.99.1
IPv6-side: 2001:db8:5000:3b/64

Routing protocol advertizes 2002::/16

• 192.88.99.1: Anycast address