Next Generation Internet
IPv6 and Beyond

Thomas C. Schmidt
t.schmidt@haw-hamburg.de
HAW Hamburg, Dept. Informatik

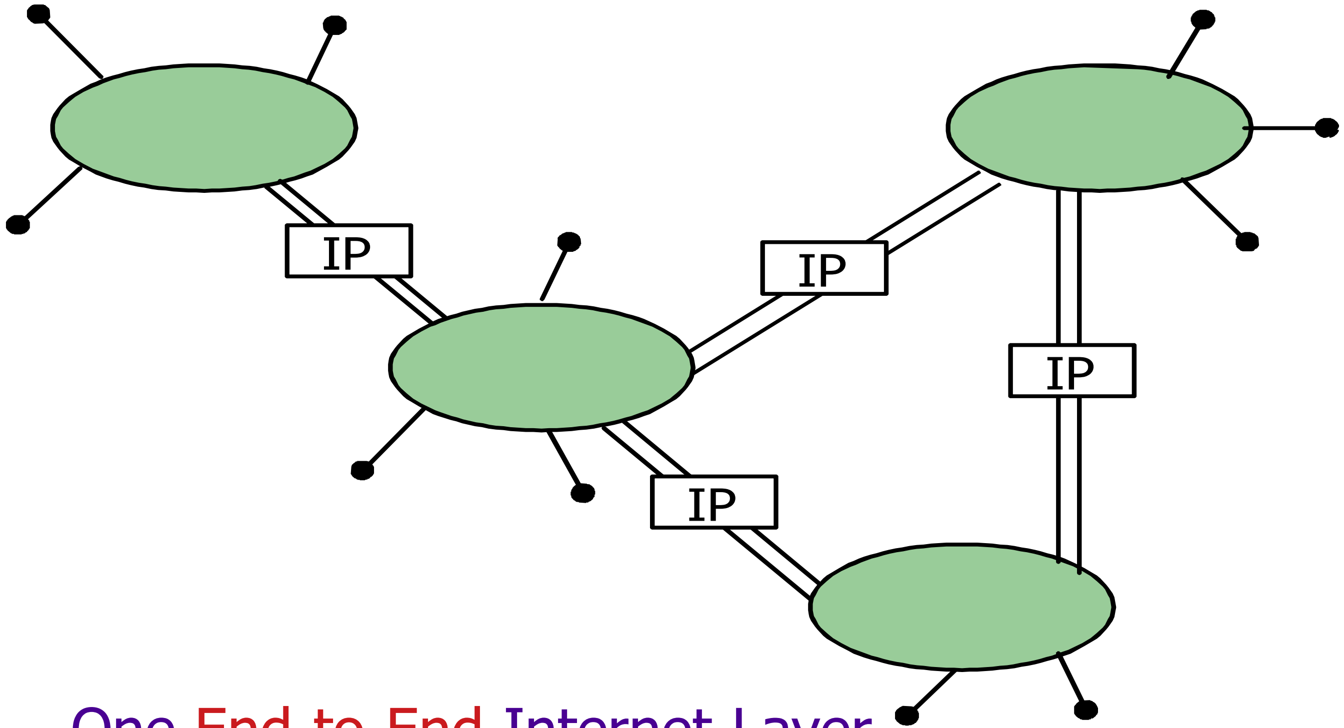
Agenda

- Motivation
- Basic IPv6 Architecture
- IPv6 Migration: Transition and Coexistence
- Future Trends: Beyond IPv6

Agenda

- Motivation
  - The Internet – Paradigm & Reality
  - The Limits of IPv4
  - Internet Service Problems
  - IPv6 Highlights
- Basic IPv6 Architecture
- IPv6 Migration: Transition and Coexistence
- Future Trends: Beyond IPv6

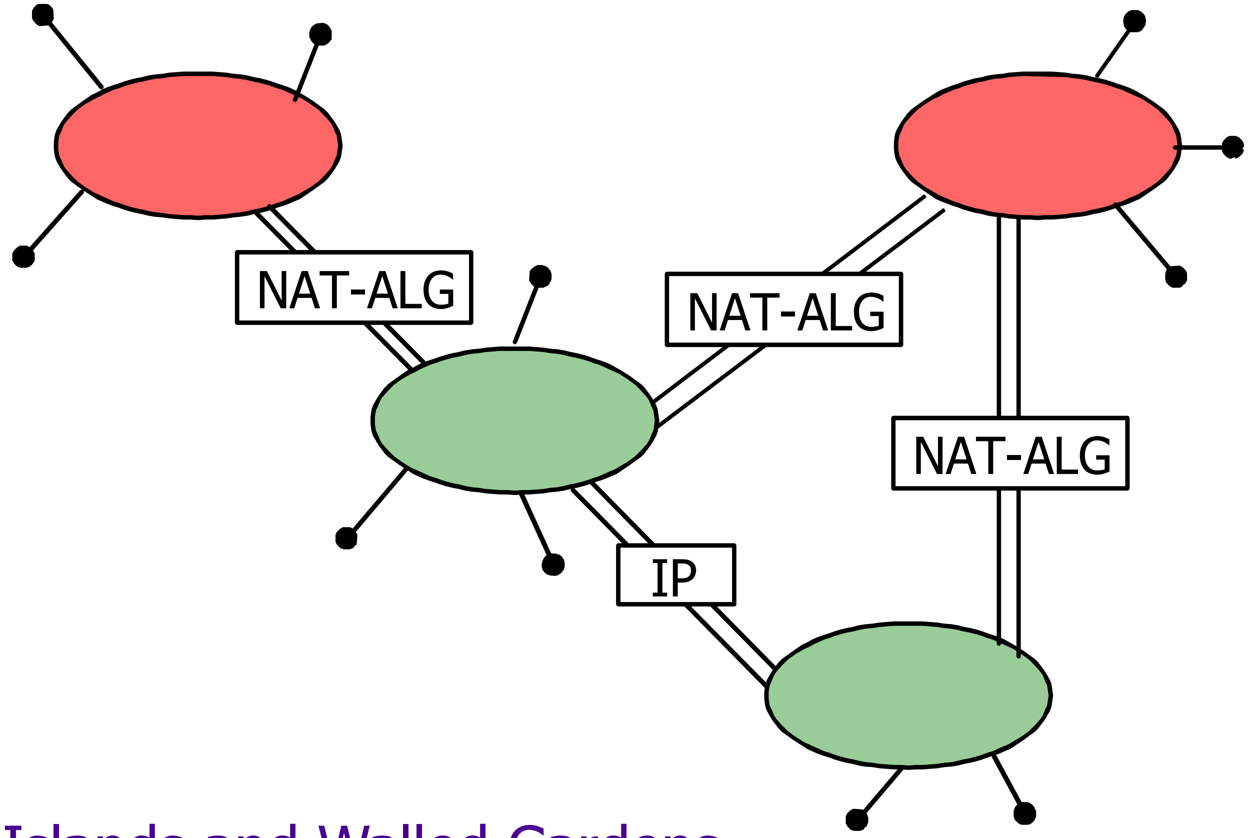
The Internet: Original Paradigm



One End-to-End Internet-Layer

- Global addressing
- Simple, application independent, transparent
- Stateless, application independent gateways

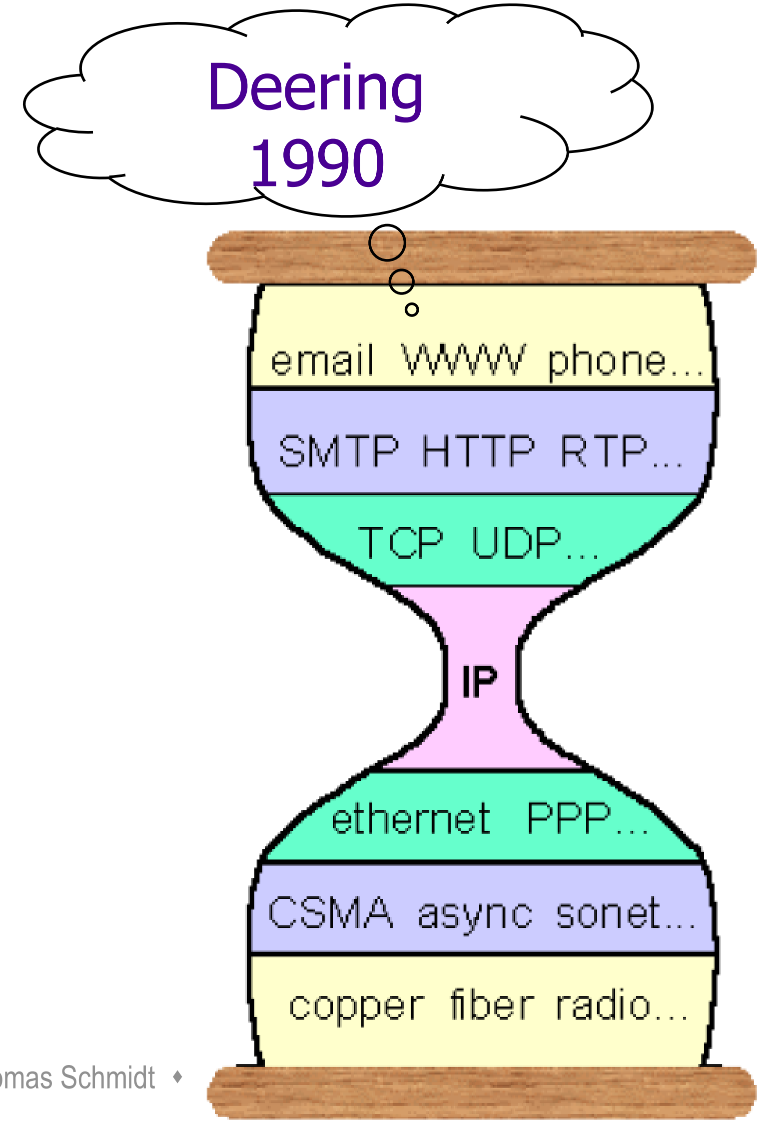
The Internet Today



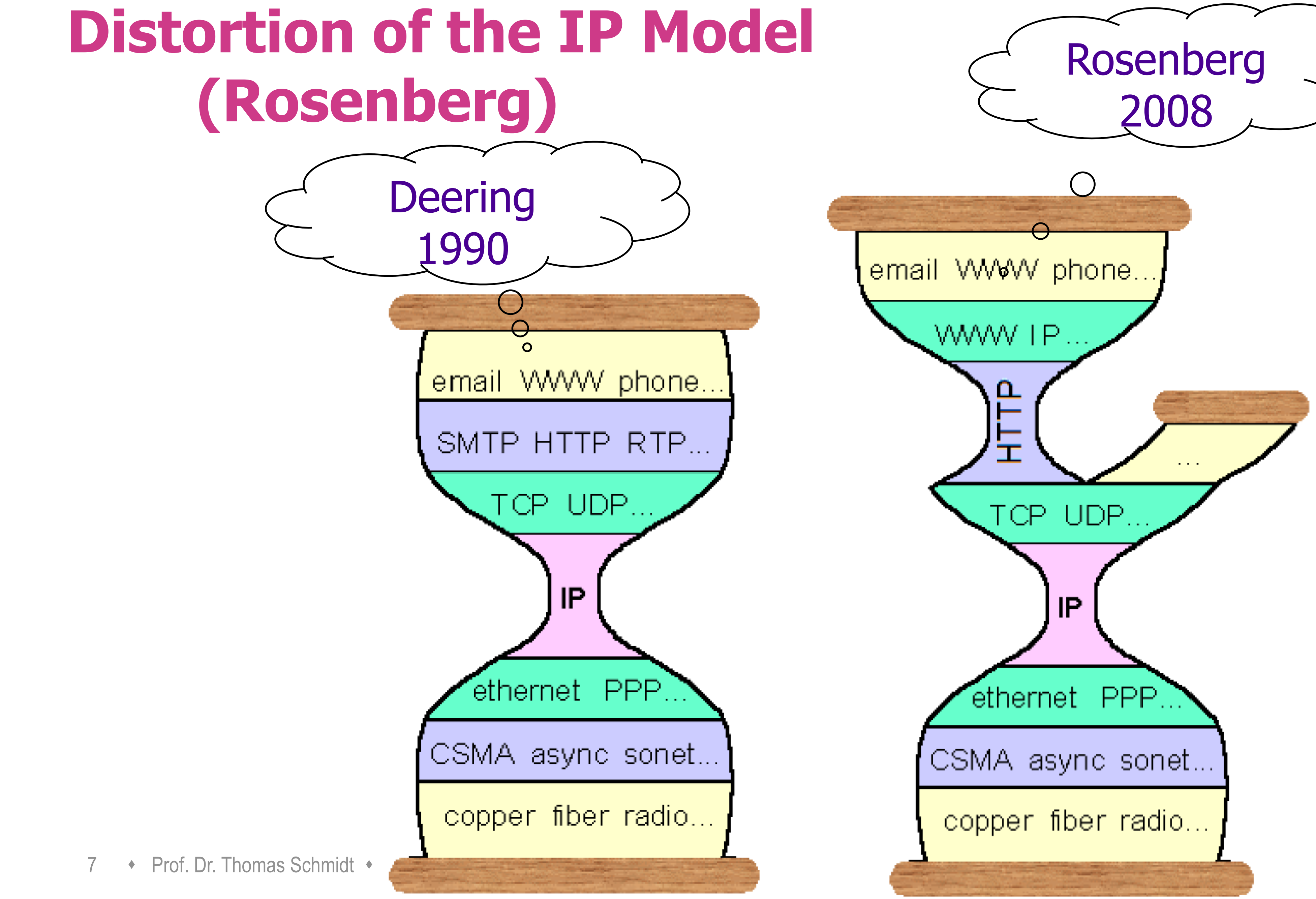
Private Islands and Walled Gardens

- No global addressing, NAT Application-Layer Gateways
- Statefull gateways for selected applications

The IP Model (Deering)



Distortion of the IP Model (Rosenberg)



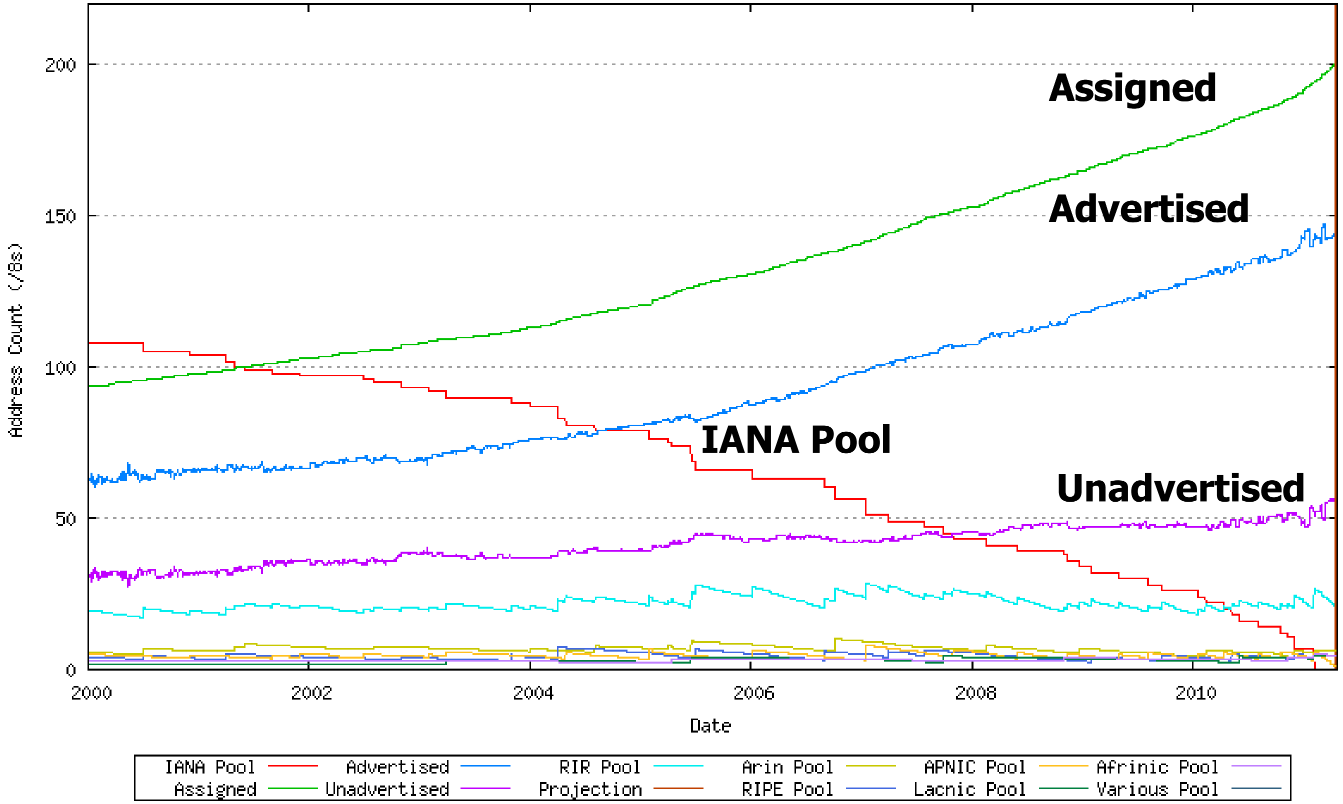
HAW HAMBURG

The Limits of IPv4

- Basic design about 50 years old
  - Packet format ... outdated
  - Hardware development of networks overran IP algorithms
- Address space exhausted
  - ‚Regular' Internet growth ran out of addresses
  - New kinds of Internet devices (mobile telephones, intelligent devices,...) need new quantities of addresses
  - Caused by address bottle-neck: NAT-ALGs
- Support of new services tedious to implement

HAW HAMBURG

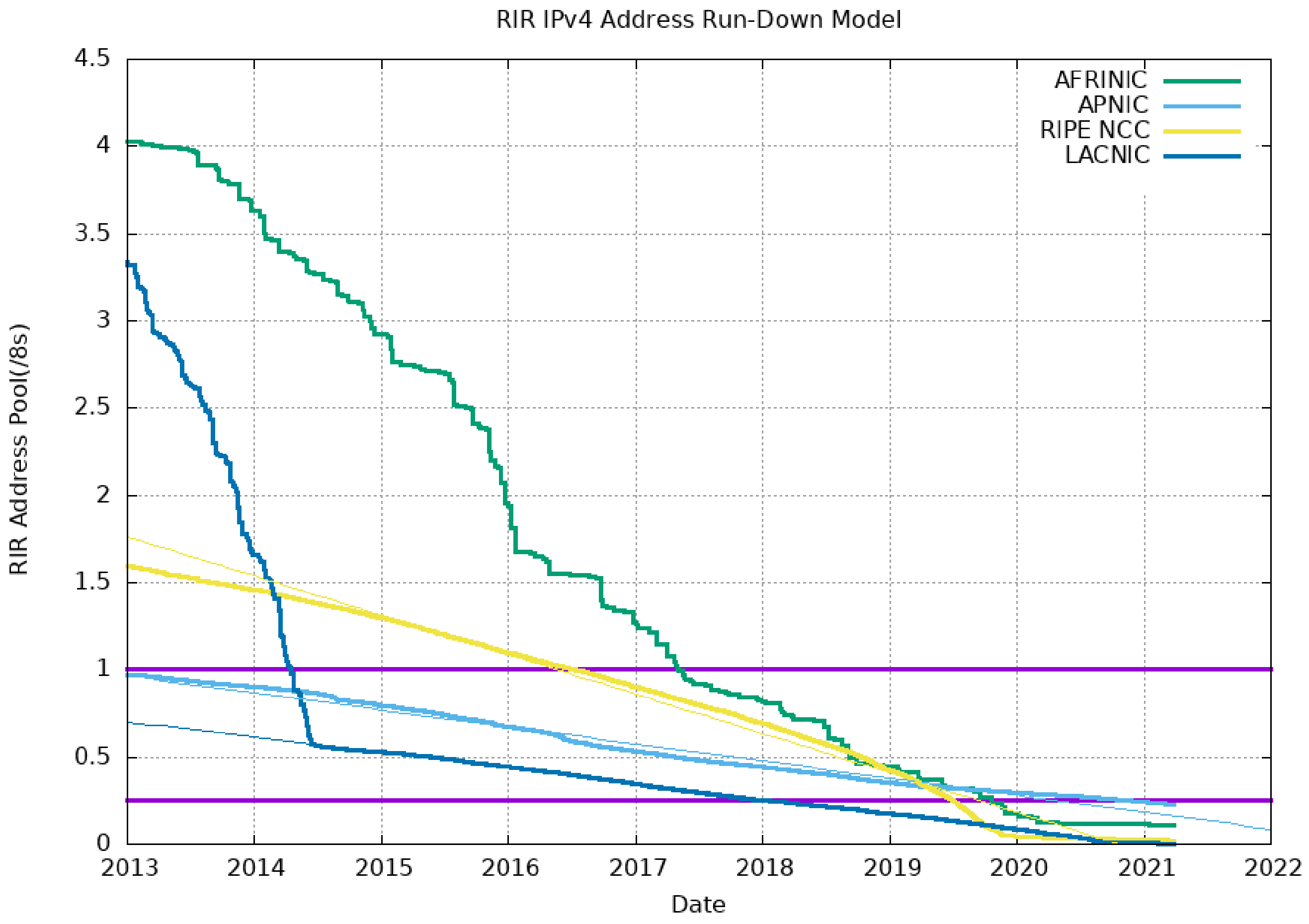
Predicted IPv4 Address Exhaustion



- IANA Unallocated Address Pool Exhaustion: 03-Feb-2011
- Projected RIR Unallocated Address Pool Exhaustion: Apr-11 – Jul-14

Source: Geoff Huston, http://www.potaroo.net/tools/ipv4/ as of Apr. 2011

IPv4 Address Exhaustion



- Arin Exhaustion: 24. Sept. 2015 – remaining NULL
- RIPE NCC: April 2021 – remaining 0.02 * /8
  Source: Geoff Huston, http://www.potaroo.net/tools/ipv4/

HAW HAMBURG

IP Service Problems

- Address configuration: Static, not stateless
- Backbone Routing: Table explosion due to unstructured addresses
- Security: IP over IP tunnelling
- Multicasting: Routing too complex
- Anycasting: Application specific solutions
- QoS: No flow support
- Mobility: Identifier/locator problem - inefficient triangular tunnelling

Why IPng: IPv6 ?

- Tackle the Internet scaling problem: Addressing & Routing
- Return to openness for new services & future development
- Evolve the architecture of the Internet
- Meet new requirements of a 'business-critical' network
- Avoid tedious patchwork to keep the Internet going

HAW HAMBURG

IPv6 Innovations

- Addressing and routing
  - Elimination of address bottle-neck: 128 Bit addresses
  - Address hierarchy can (was intended to) simplify backbone routing
  - Several addresses per interface
- Simple administration
  - Autoconfiguration of interfaces without DHCPv6
  - Floating net masks, renumbering via prefix change
- Security: IPSec
  - Security header extension for authentication, integrity and encryption

HAW HAMBURG

IPv6 Innovations (2)

- o Protocol architecture
  - Slim, fixed header for fast processing
  - Optional extension headers
  - Format framework for header classes
  - No header checksum
  - No fragmentation in routers
- o Improved multicast, anycast, QoS and mobile services
- o Support of Jumbograms (> 64 KB)
- o Transition and coexistence concept IPv4 ↔ IPv6

IPv6 History

- IETF WG IPng began to work in the early 90er
- Winter 1992: 7 proposals for development of IP
  - CNAT, IP Encaps, Nimrod, Simple CLNP, PIP, SIP, TP/IX
- Autumn 1993: several mergers lead to
  - 'Simple Internet Protocol Plus' (SIPP) and 'Common Architecture for the Internet' CATNIP
- July 1994: IPng Area Director recommend roadmap (RFC 1752) on basis of SIPP (Steve Deering)
- Dec. 1995: S. Deering, R. Hinden, „Internet Protocol, Version 6 (IPv6) Specification" (RFC 1883, now RFC 2460)
- July 1999: End user addresses available (RIPE-NCC, APNIC, ARIN)
- May 2007: ARIN advises Internet Community on Migration to IPv6

IPv6 Standardisation

- Key components in standard track:

  Specification (RFC2460) Neighbour Discovery (RFC2461)

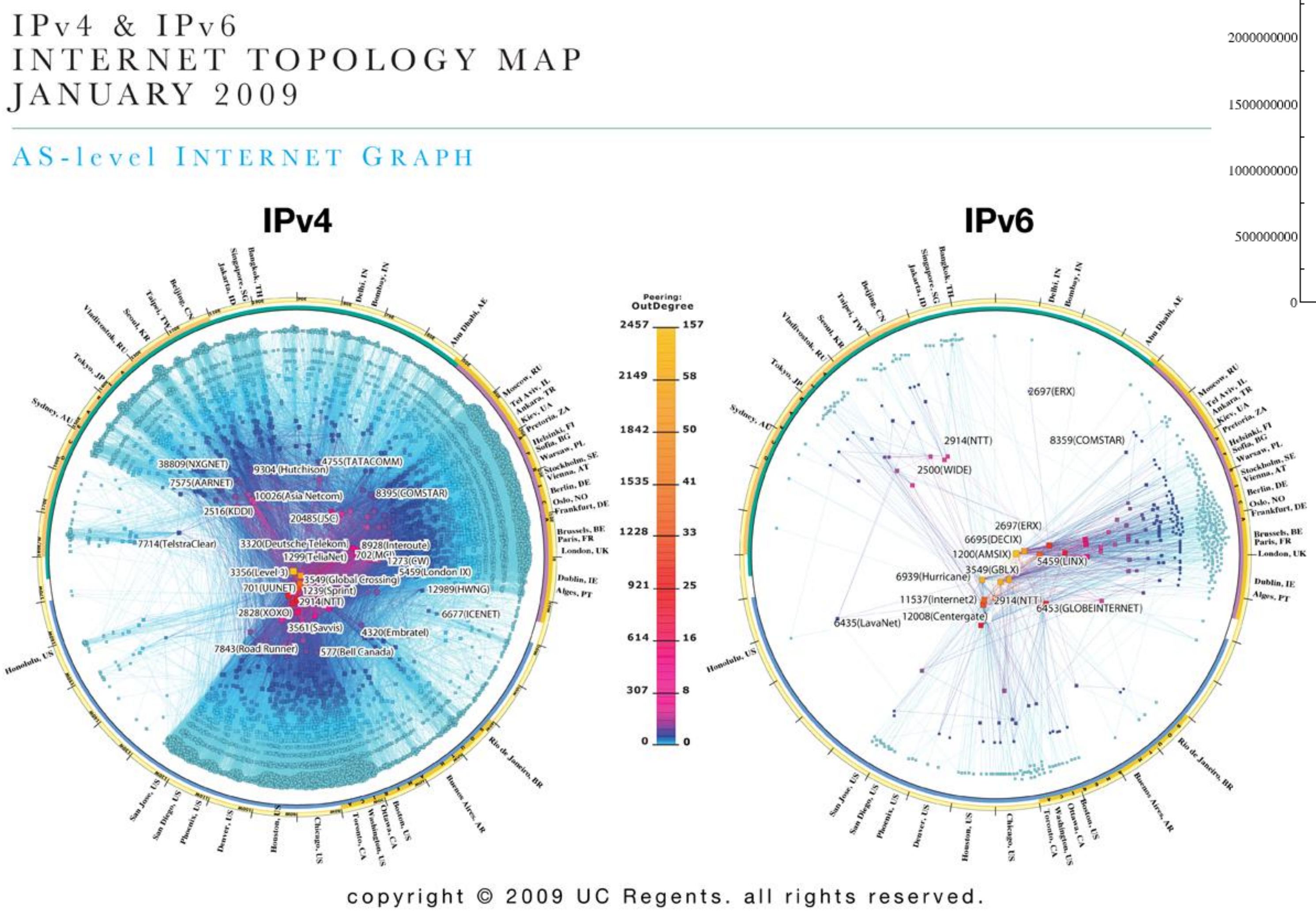
ICMPv6 (RFC2463)	IPv6 Addresses (RFC1884 ++)
RIP (RFC2080)	BGP (RFC2545)
IGMPv6 (RFC2710)	OSPF (RFC2740)
Router Alert (RFC2711)	Jumbograms (RFC2675)
Auto configuration (RFC2462)	….

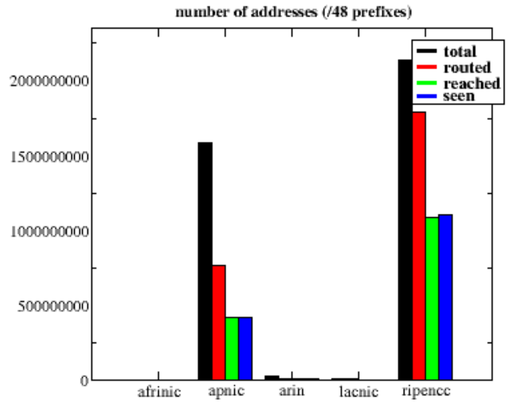
- IPv6 over:

PPP (RFC2023)	Ethernet (RFC2464)
FDDI (RFC2467)	Token Ring (RFC2470)
NBMA(RFC2491)	ATM (RFC2492)
Frame Relay (RFC2590)	ARCnet (RFC2549)

- Since then 1000++ further standards: Flow labelling, MIPv6, 3GPP, Routing advertisement, ….
- Implemented basically in every Internetworking system platform

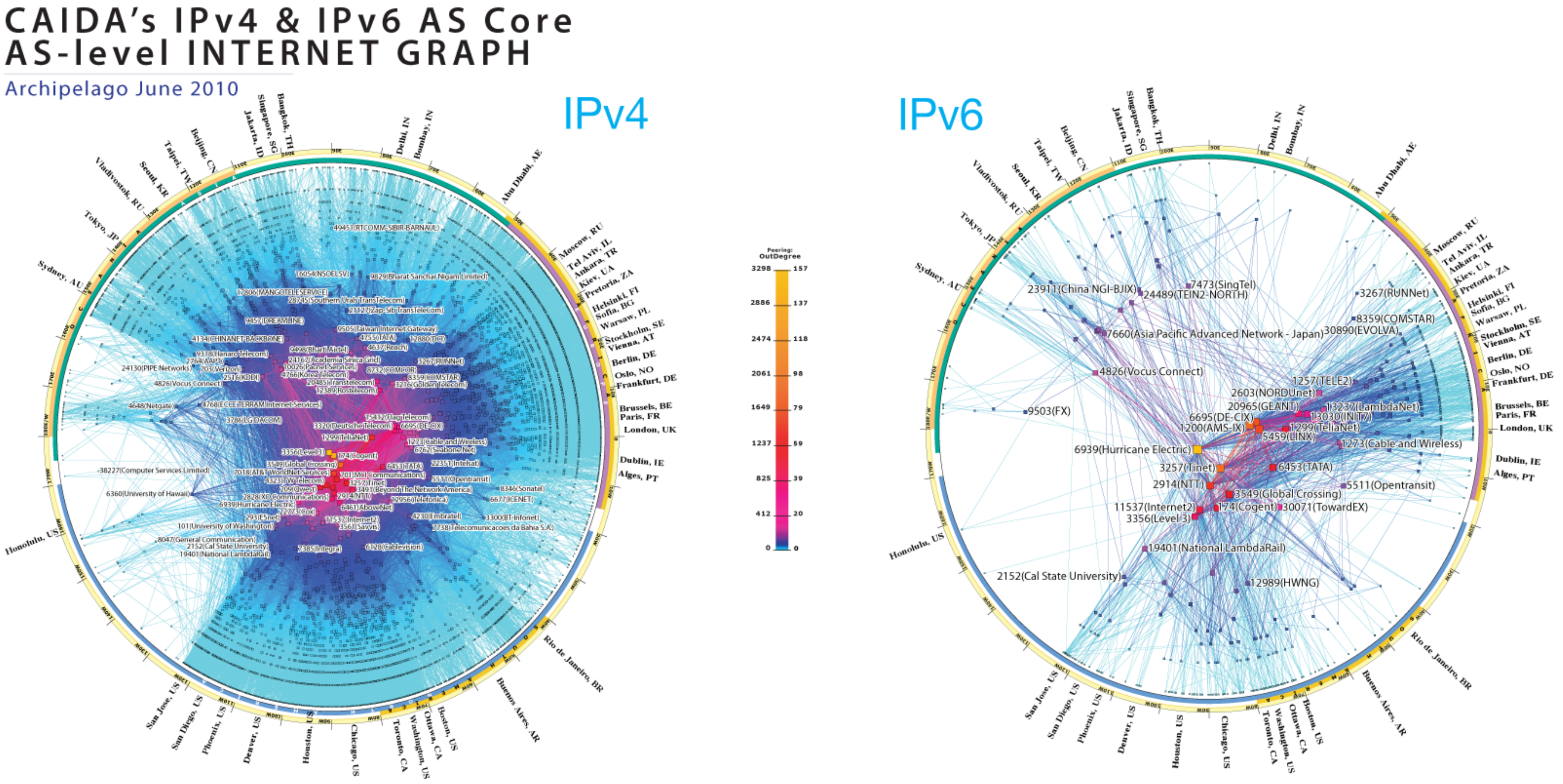
HAW HAMBURG





- Source: CAIDA (http://www.caida.org/research/topology/as_core_network/ipv6.xml)

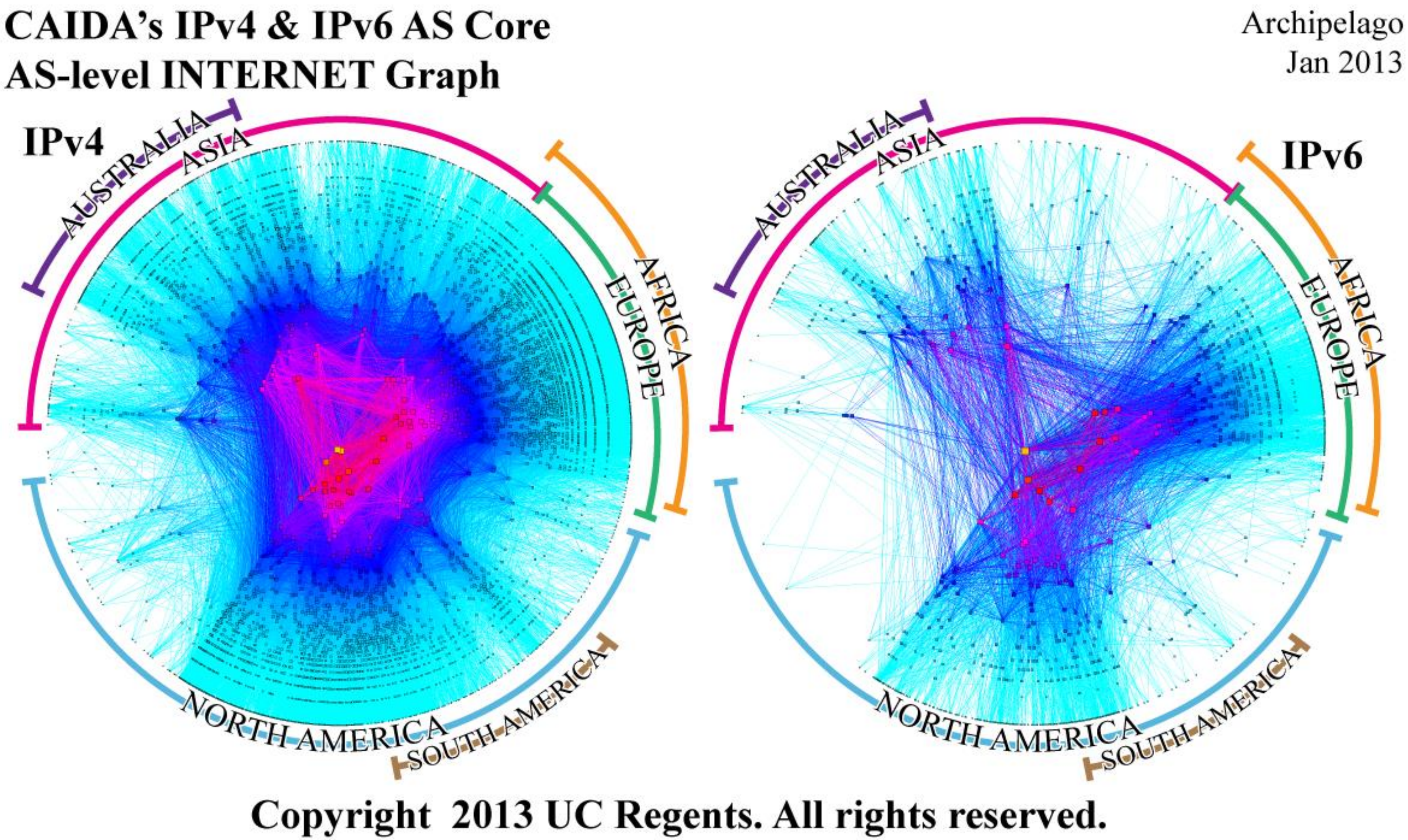
IPv6 Deployment Progress June ‘10



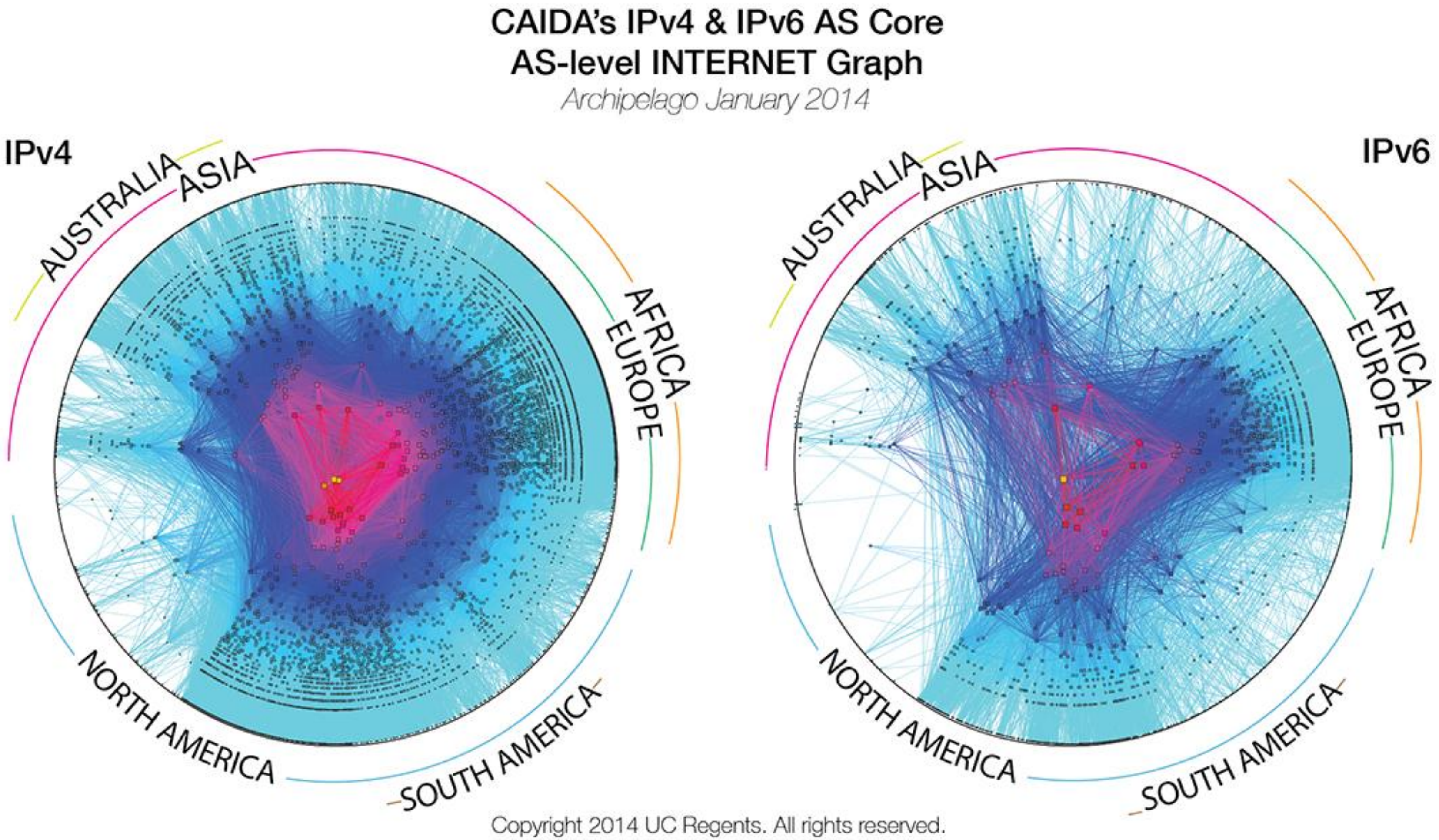
- Source: CAIDA (http://www.caida.org/research/topology/as_core_network/ipv6.xml)

HAW HAMBURG

IPv6 Deployment Progress Jan ‘13

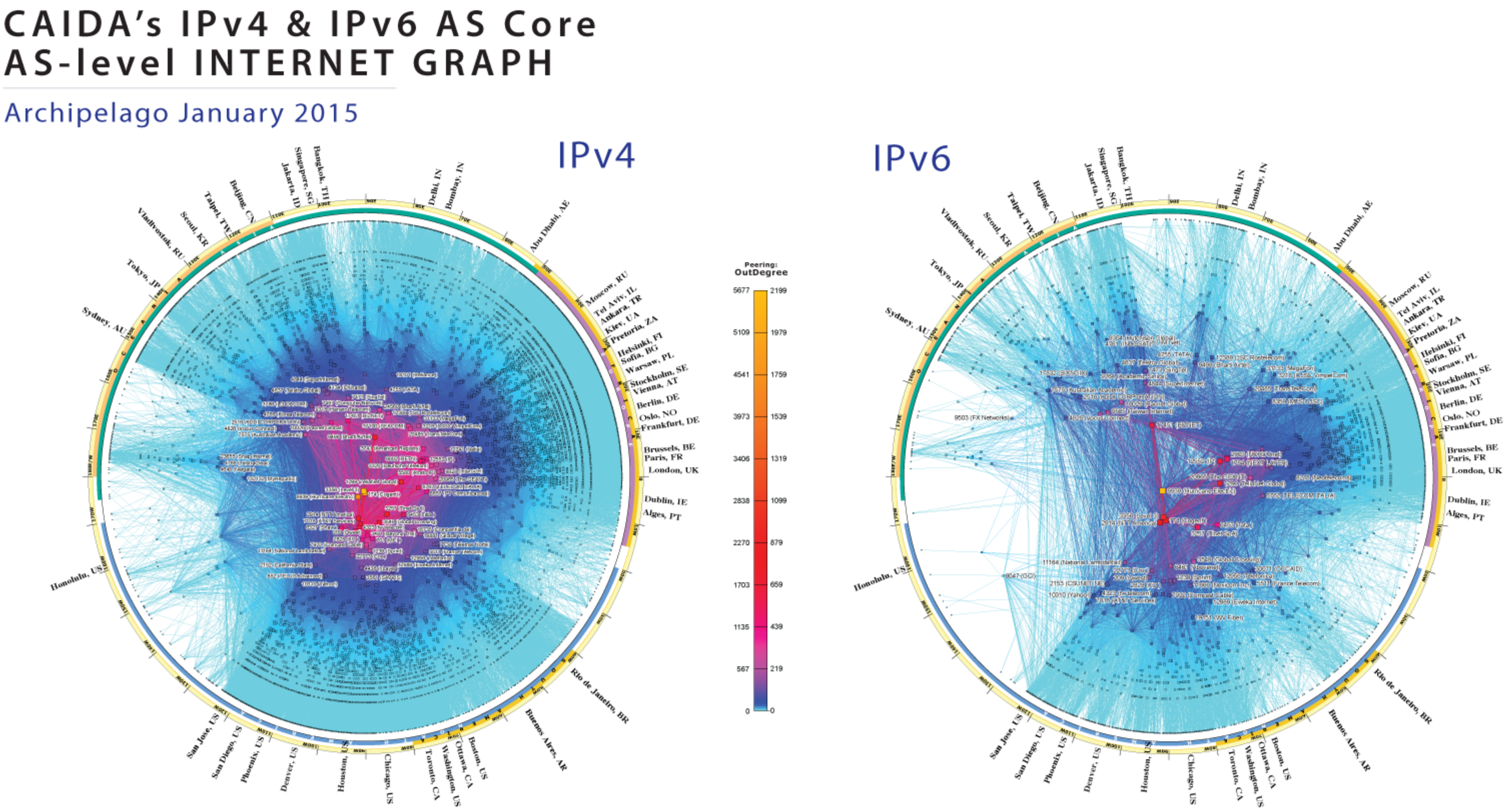


IPv6 Deployment Progress Jan '14



HAMBURG

IPv6 Deployment Progress Jan ‘15



IPv6 Deployment 2020

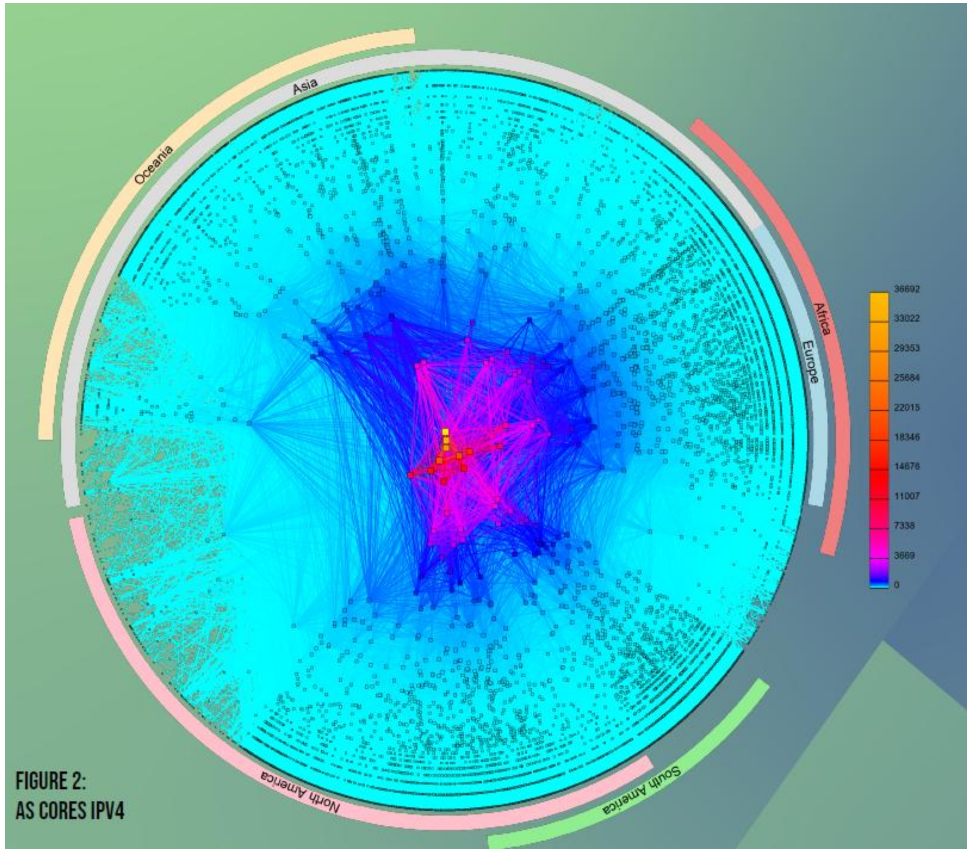


FIGURE 2:
AS CORES IPV4

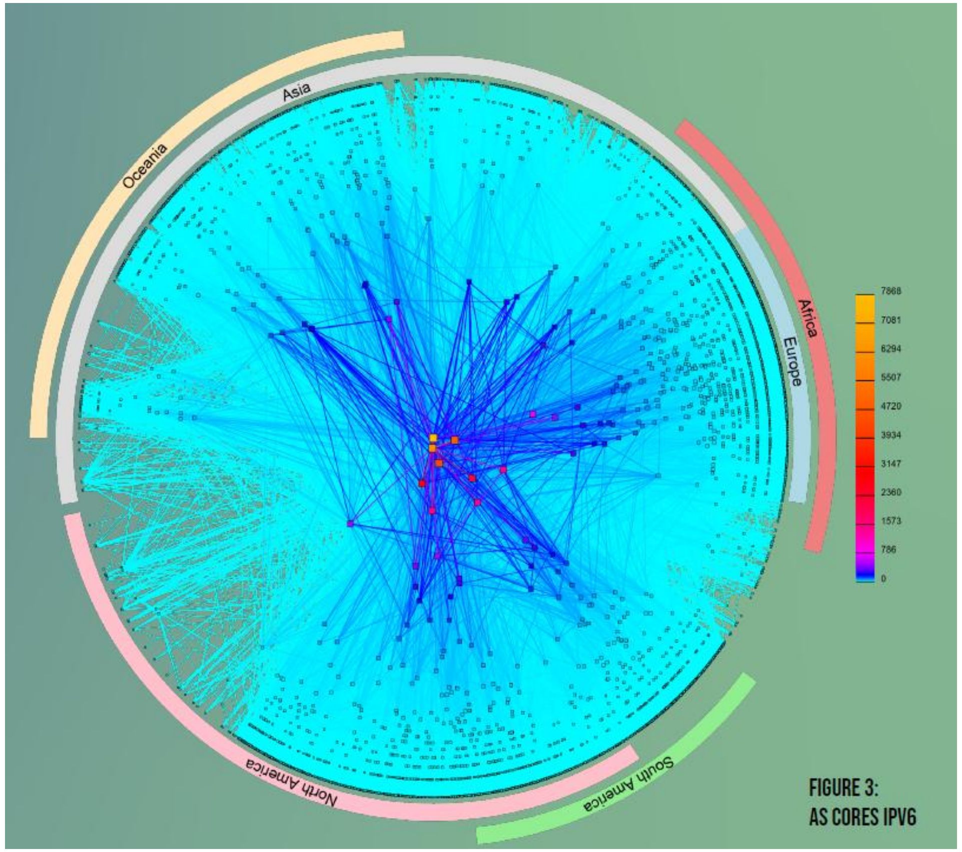


FIGURE 3:
AS CORES IPV6

Agenda

- Motivation
- Basic IPv6 Architecture
  - Addressing
  - Packet Format
  - ICMP, Neighbour Discovery, Autoconfiguration
  - Routing, Anycasting, QoS
- IPv6 Migration: Transition and Coexistence
- Future Trends: Beyond IPv6

Addressing

- IPv6 addresses are 128-bit long and variably built
- Address architecture: RFC 1884, now 4291 (Feb ´06, Hinden & Deering)
- Automatic address configuration
- Global address hierarchy from top level allocation to the interface-ID designated
- Aggregation-based allocation to simplify the global routing (target objective)
- Format prefix (FP) (3 Bit initially) used for identification of address type

HAW HAMBURG

Notation of IPv6 Addresses

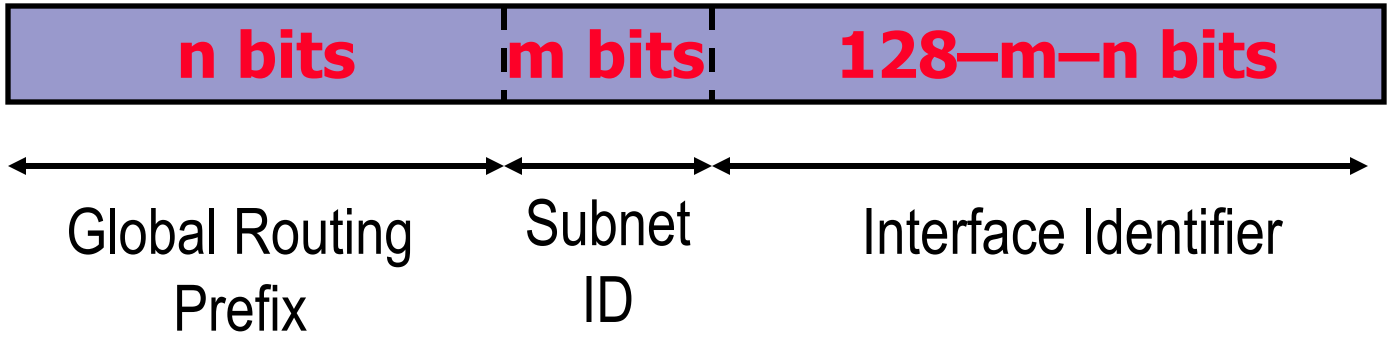
- Standard form: 8 x 16 bit hexadecimal
  Example: 1080:0:FF:0:8:800:200C:417A
- Short form: sequences of nulls replaced by ::
  Example: FF01:0:0:0:0:0:0:43 → FF01::43
- IPv4 compatible addresses:
  Example: 0:0:0:0:0:FFFF:13.1.68.3 → ::FFFF:13.1.68.3
- CIDR notation for prefixes:
  Example: 1080:645:FF::/48

Address Types

Type	Binary Prefix	
o Unicast (one-to-one)		
- global	all not specified elsewhere	
- site-local (deprecated)	1111 1110 11	(FEC0::/10)
- unique local (ULA)	1111 110	(FC00::/7)
- link-local	1111 1110 10	(FE80::/10)
- IPv4-mapped	000…0:FFFF	(::FFFF:xxx.xxx.xxx.xxx)
- loopback	0000…1	(::1/128)
- unspecified	0000…0	(::/128)
o Multicast (one-to-many)	1111 1111	(FF00::/8)
o Anycast (one-to-nearest)	of Unicast Prefixes	

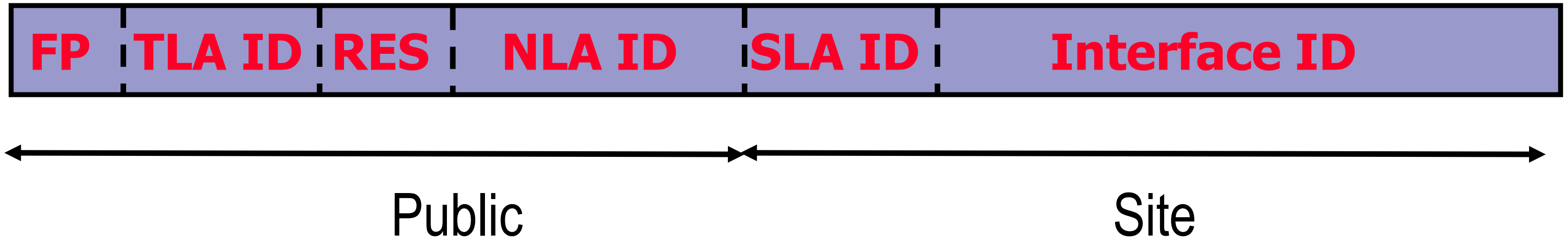
o No broadcast addresses (only multicast)!

Global Unicast Addresses - RFC 4291



- All fields have variable length and are not ‘self-explanatory’ (as of CIDR)
- All global unicast addresses, which do not begin with 000 (binary), carry a 64 bit interface ID, this means m + n = 64

Historic – RFC2374: Aggregatable Global Unicast Format



Previous approach: Standardized prefix hierarchy as Top/Next/Side Level Aggregator

Current approach:

- IAB/IESG Recommendations on IPv6 Address Allocations to Sites
- Left to RIR policies cf. http://www.ripe.net/ripe/docs/ipv6policy.html

  „Wherever possible, address space should be distributed in a hierarchical manner, according to the topology of network infrastructure.”

Local Unicast Addresses

- o Link-local addresses for use during auto-configuration and in nets without routers:

1111111010	0	Interface ID

- o Unique local addresses (RFC 4193), independent of TLA/NLA:
  - Randomly unique Global ID for avoiding conflicts
  - Not intended for global routing (but e.g., for dedicated site interconnects)

1111110	L	Global ID	SLA	Interface ID

Multicast Addresses

11111111	flags	scope	Group ID
8	4	4	112 bits

- Flag field: lower bit indicates permanent (=0) respectively transient (=1) group, remainder reserved for addressing schemes
- Scope field:
  1 - node local
  2 - link-local
  5 - site-local
  8 - organisation local
  B - community-local (deprecated)
  E - global (other values reserved)

Example: Our First IPv6 Net in 2001

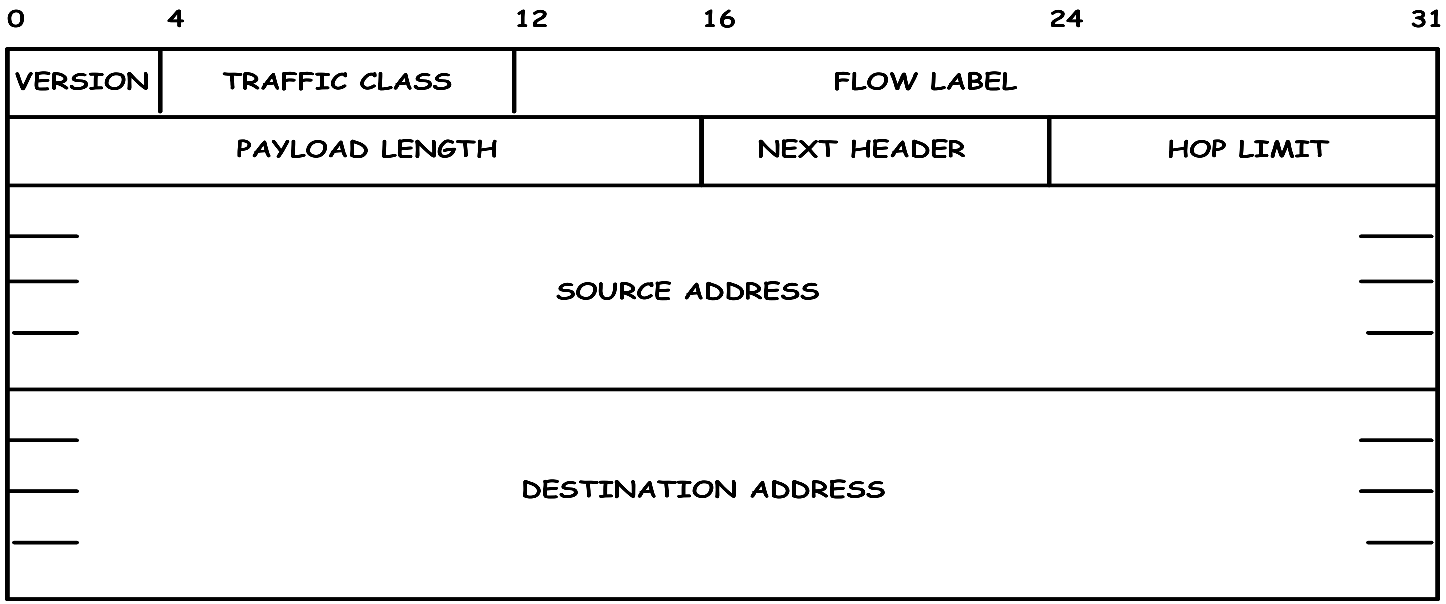
- 2001:: /16 - Pre-set prefix
- 2001:**06**00:: /24 - Regional registry Europa (RIPE)
- 2001:06**38**:: /32 - DFN prefix
- 2001:0638:**0801**:: /48 - Our network address
- 2001:0638:0801:**0001**:: /64 - Our first subnet
- 2001:0638:0801:0001:**0000:0000:0000:0001** /128

  - Our first IPv6 computer address in 2001 ☺

Addressing of Sub-TLAs (Ripe) according to RFC 2450

HAW HAMBURG

IPv6 Packet Format: Basic Header



VERSION	4 Bit	Internet Protocol Version Number = 6
TRAFFIC CLASS	8 Bit	Type of Services (QoS DiffServ field)
FLOW LABEL	20 Bit	Flow Identification at Routers (QoS)
PAYLOAD LENGHT	16 Bit	Octetts of Payload without IPv6-Header
NEXT HEADER	8 Bit	Type of Encapsulated Protocol
HOP LIMIT	8 Bit	TTL-Counter, Decremented per Router
SOURCE ADDRESS	128 Bit	Adress of Sender (128 Bits)
DESTINATION ADRESS	128 Bit	Adress of Receiver (128 Bits)

Compare: IPv4 Header

IP-Protocolkopf

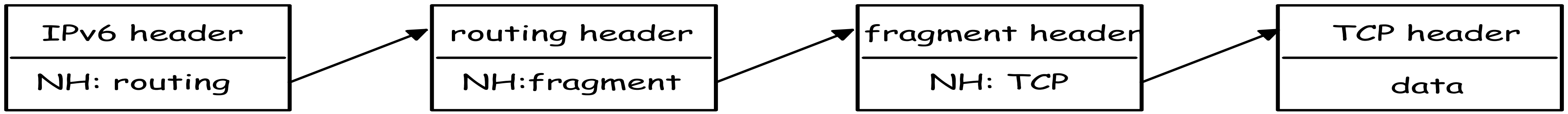
<table>
<tr><td>1</td><td>4</td><td>8</td><td>16</td><td>19</td><td>24</td><td>32</td></tr>
<tr><td>Version</td><td>Länge</td><td>Servicetypen</td><td colspan="4">Paketlänge</td></tr>
<tr><td colspan="3">Identifikation</td><td>D F</td><td>M F</td><td colspan="2">Fragmenabstand</td></tr>
<tr><td colspan="2">Lebenszeit</td><td>Transport</td><td colspan="4">Kopfprüfsumme</td></tr>
<tr><td colspan="7">Senderadresse</td></tr>
<tr><td colspan="7">Empfängeradresse</td></tr>
<tr><td colspan="5">Optionen</td><td colspan="2">Füllzeichen</td></tr>
</table>

Header Changes of IPv4

- Addressing grows from 32 to 128 Bit
- Fragmenting deleted from basis header
- IP Options deleted from basis header → fixed length
- Header Checksum drop out
- Header length field drop out
- Flow Label newly included
- Time to Live → Hop Limit
- Protocol → Next Header
- Service types → Traffic Class
- Length field describes data without header
- Alignment increases from 32 to 64 Bit

IPv6 Packet Format: Option Headers

- Extended option mechanisms: Each header references a possible successive header or data, e.g.:



- Option headers have no length limit (IPv4: 40 Octets), Padding to 8 Octets
- Option headers will be processed only at destination host, not by routers. Exception: Hop-by-Hop Option Header

Basic Option Headers

- Routing
  Advanced routing information (source routing)
- Fragmentation
  Fragmentation / defragmentation information
- Authentication
  Security information: IPsec authentication and integrity
- Encapsulation
  Secure ‚Tunnelling': IPsec for confidential data
- Hop-by-Hop Option
  Dedicated options to be processed by every router
- Destination Option
  Information for the destination host (header extension)

Order of Headers

The processing order of the headers will be arranged by the sender according to the following recommendation (RFC 2460):

1. IPv6
2. Hop-by-Hop Option
3. Destination Option (for Routers with 2.)
4. Routing / Encapsulation
5. Fragmentation
6. Authentication
7. Destination Option (for Endpoints)
8. Upper Layer

Internet Control Message Protocol (ICMPv6)

- RFC 2463 (Conta, Deering), now RFC 4443 + 4884
- Extension header protocol class (following base IP header)
- Defines two (expandable) message classes:

Informational Messages

- Echo Request (128)
- Echo Reply (129)

Error Messages

- Destination Unreachable (1)
- Packet Too Big (2)
- Time Exceeded (3)
- Parameter Problem (4)

IPv6 Neighbour Discovery

- o RFC 2461, now RFC 4861
- o Protocol over ICMPv6
  - Combination of IPv4 Protocols (ARP, ICMP,…)
- o Autonomous interaction between hosts and routers
  - Defines 5 ICMPv6 packet types:
    - Router Solicitation / Router Advertisement
    - Neighbour Solicitation / Neighbour Advertisement
    - Redirect

HAW HAMBURG

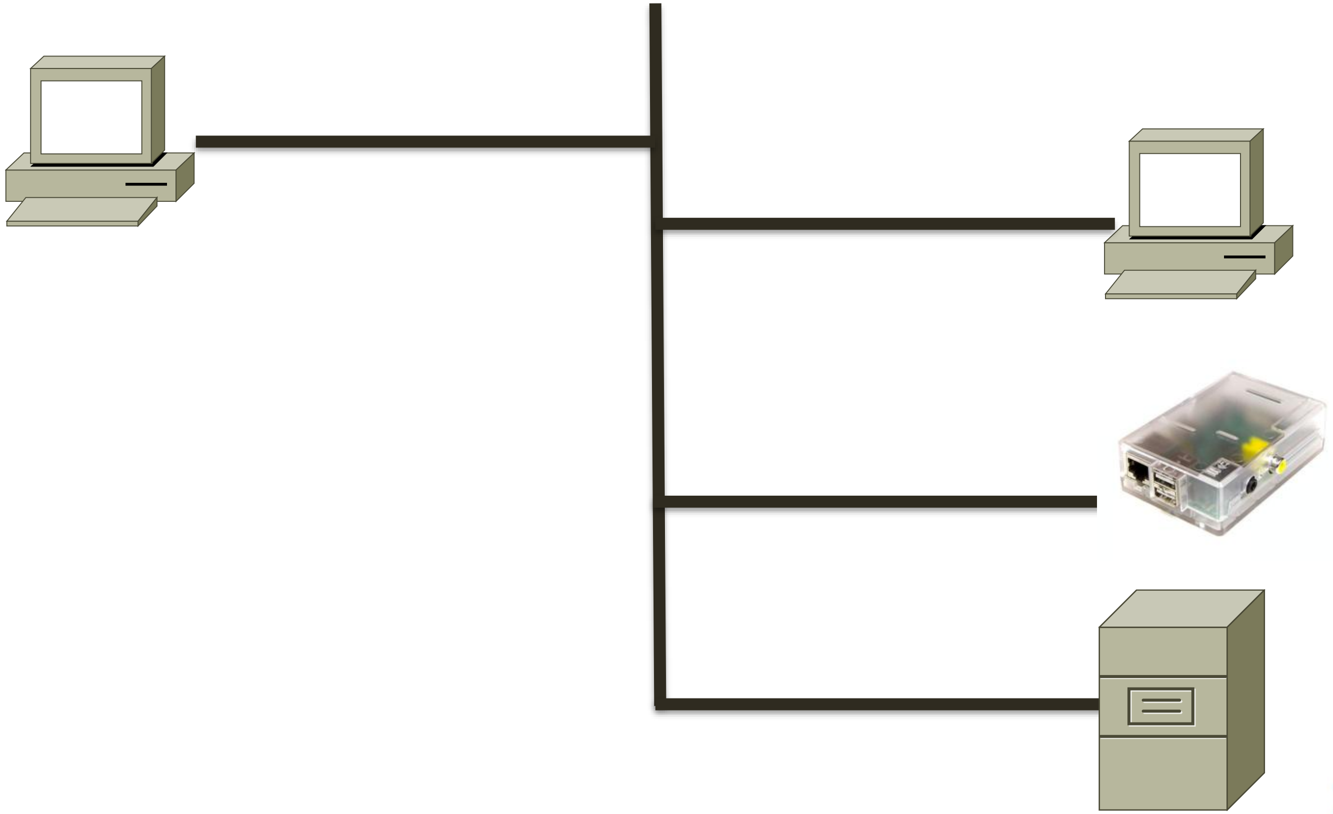
IPv6 Neighbour Discovery (2)

Defines communication mechanisms for nodes on the same link:

- Router discovery
- Prefix discovery
- Parameter discovery, i.e.: link MTU, hop limit,…
- Address auto-configuration
- Address resolution (same function as ARP)
- Next-hop determination
- Neighbour unreachable detection (useful for default routers)
- Duplicate address detection
- Redirect
- Network load balancing

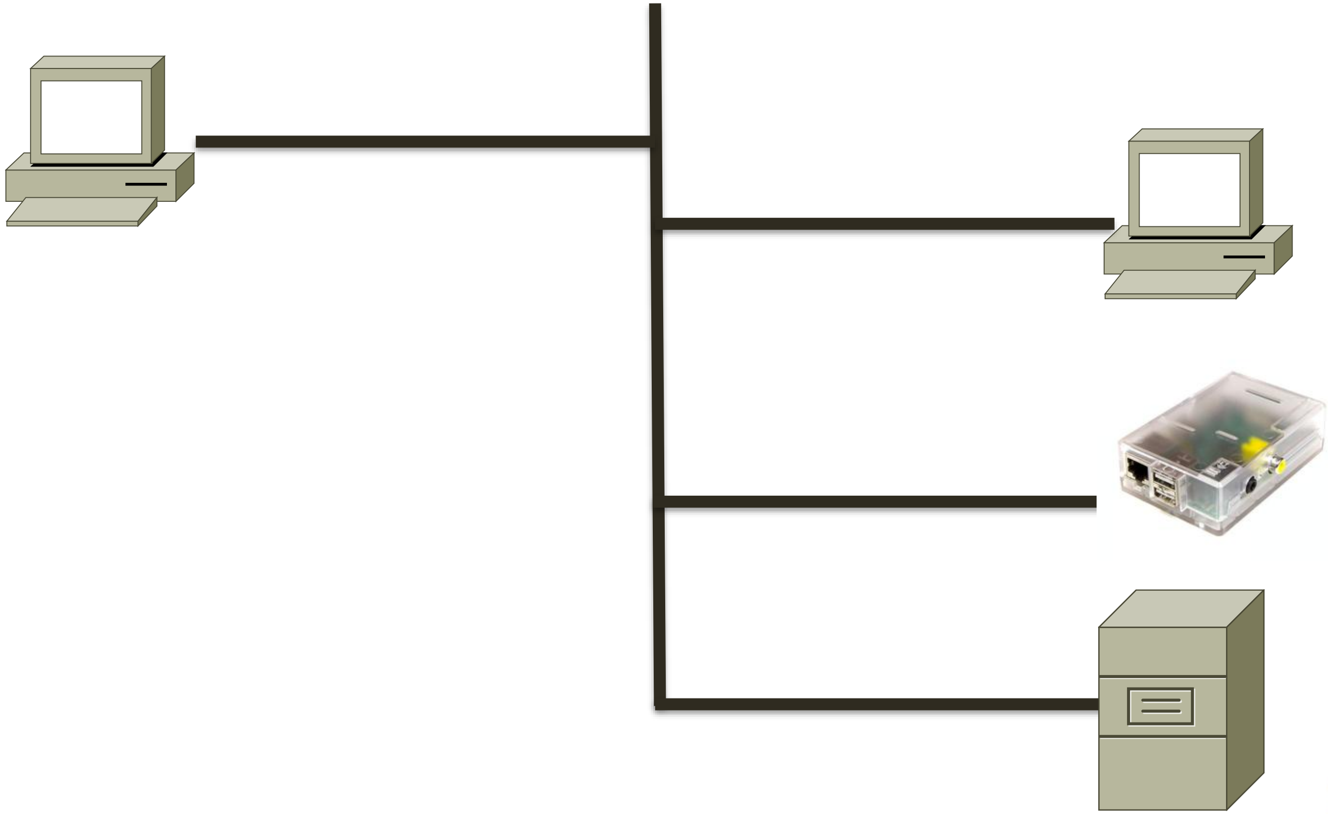
HAW HAMBURG

Neighbour Discovery Handshake



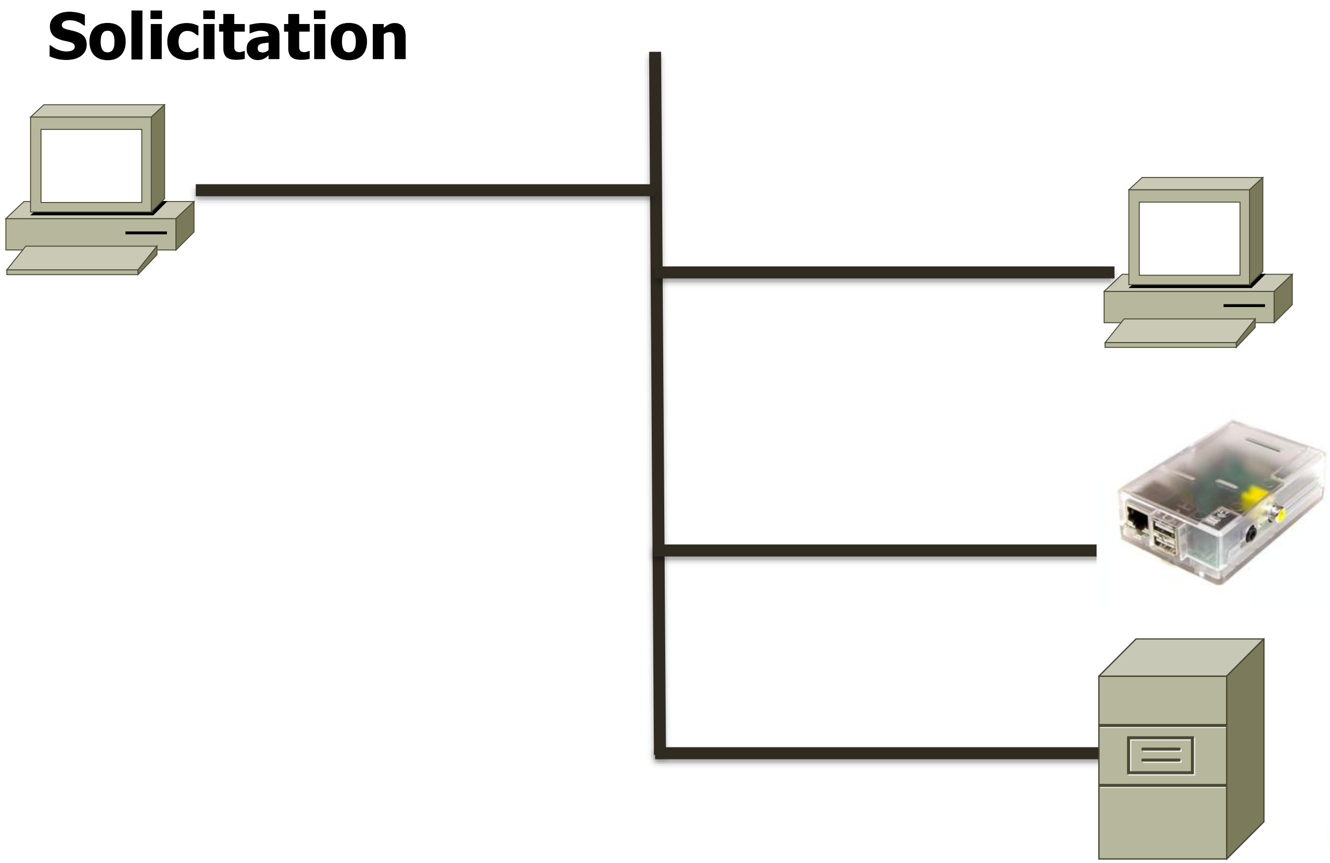
Neighbour Discovery Handshake

- Address of my neighbour?
  Search **FE80**::**ba**27:ebFF:FE3b:7def/64



Neighbour Discovery Handshake

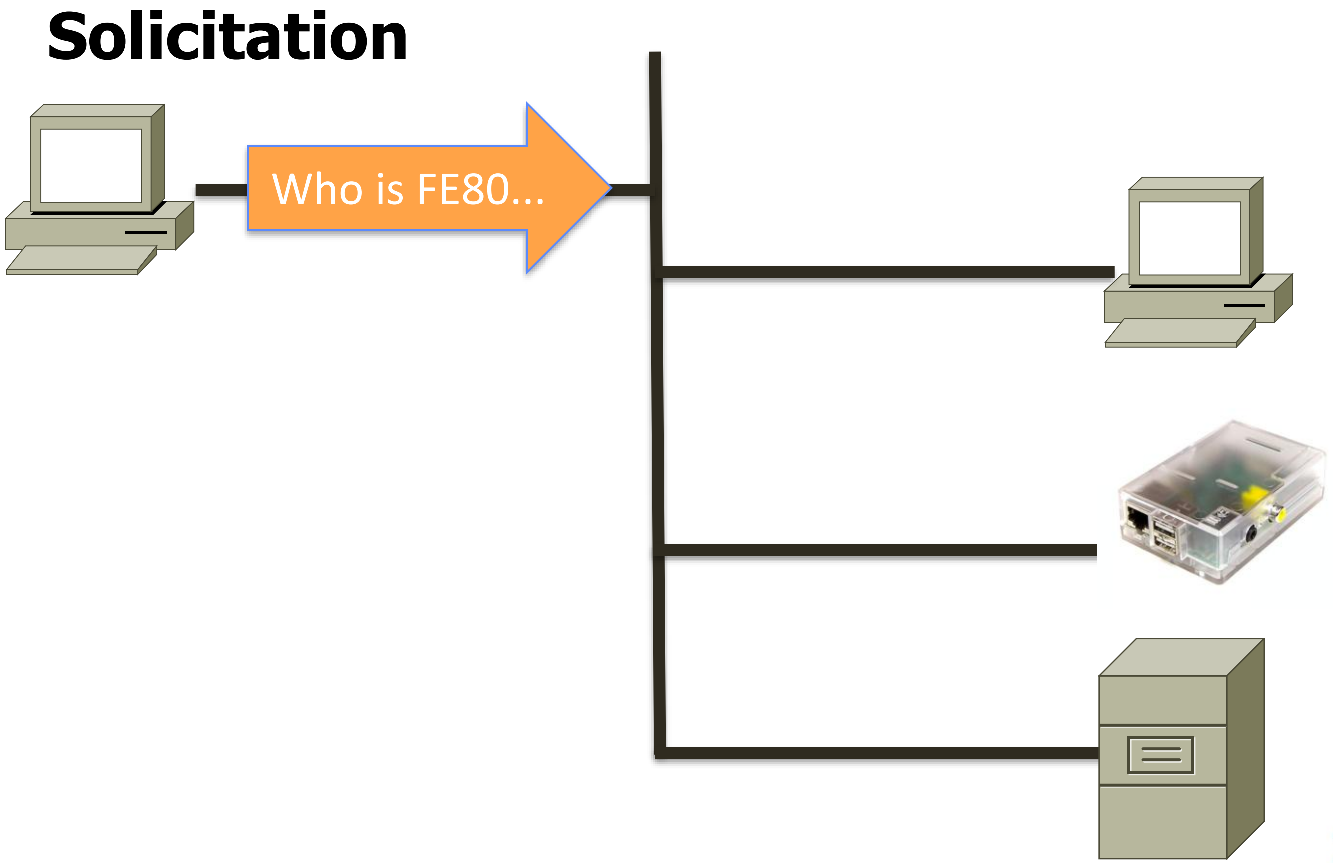
- Address of my neighbour?
  Search **FE80**::**ba**27:ebFF:FE3b:7def/64



HAW HAMBURG

Neighbour Discovery Handshake

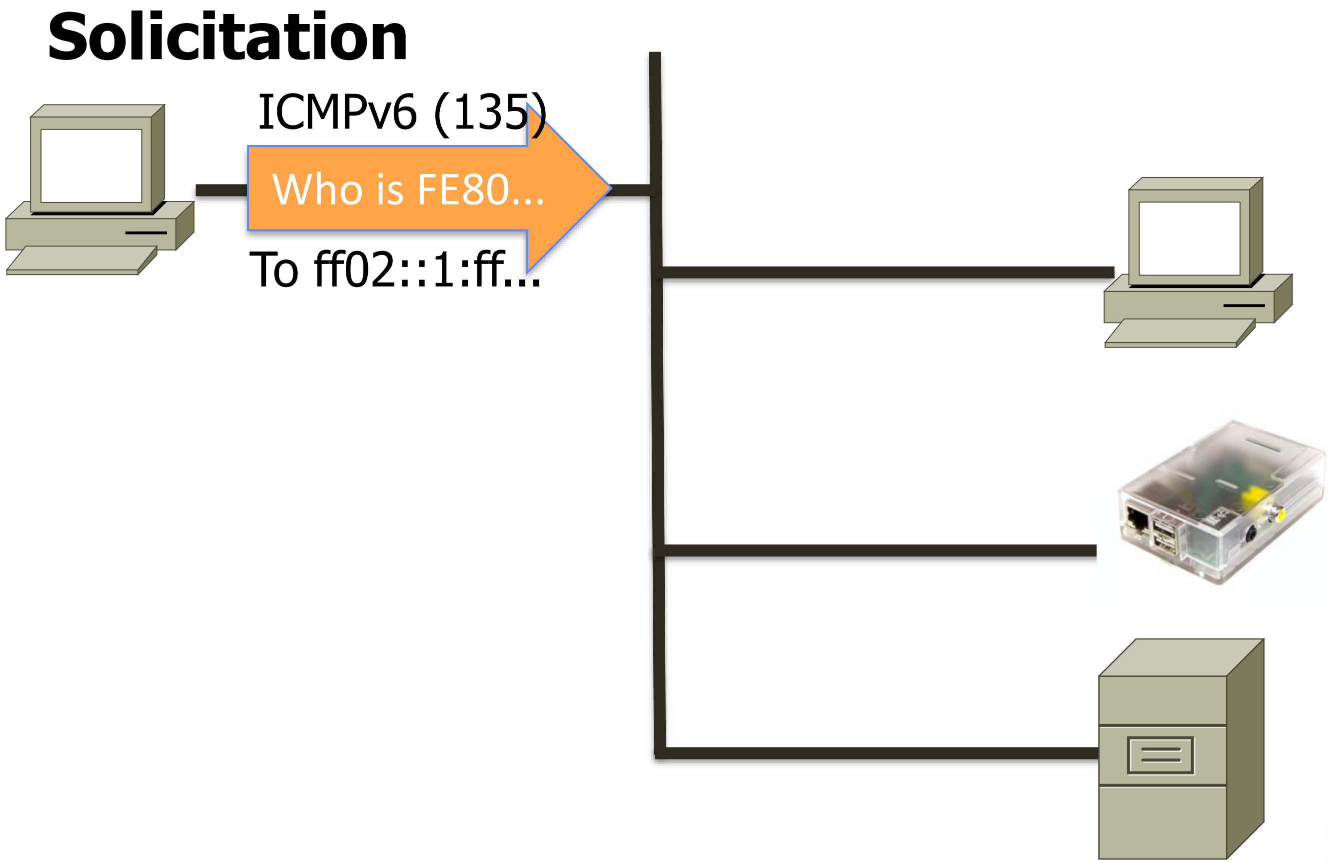
- Address of my neighbour?
  Search **FE80**::**ba**27:ebFF:FE3b:7def/64



HAW HAMBURG

Neighbour Discovery Handshake

- Address of my neighbour?
  Search **FE80**::**ba**27:ebFF:FE3b:7def/64

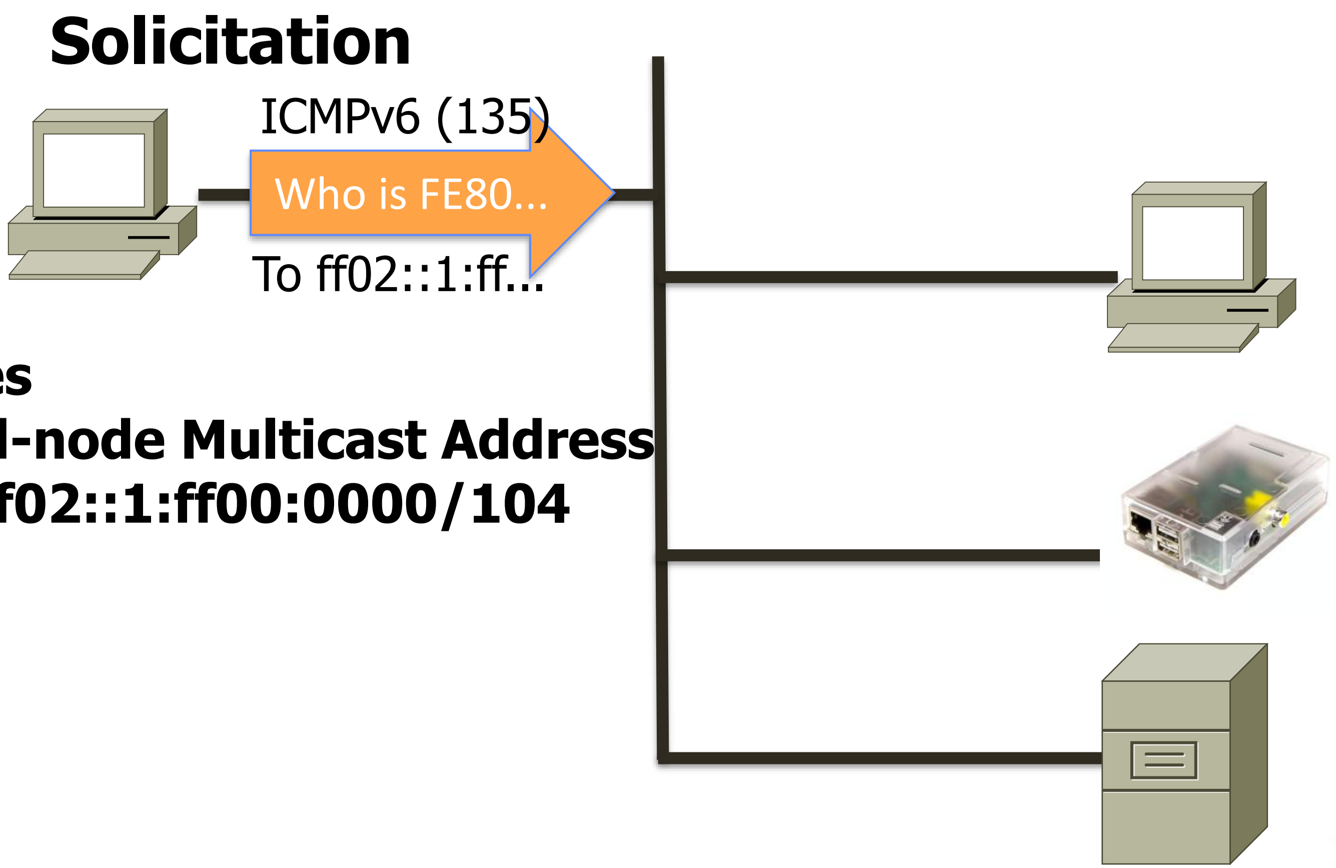


HAW HAMBURG

Neighbour Discovery Handshake

- Address of my neighbour?

  Search **FE80**::**ba**27:ebFF:FE3b:7def/64

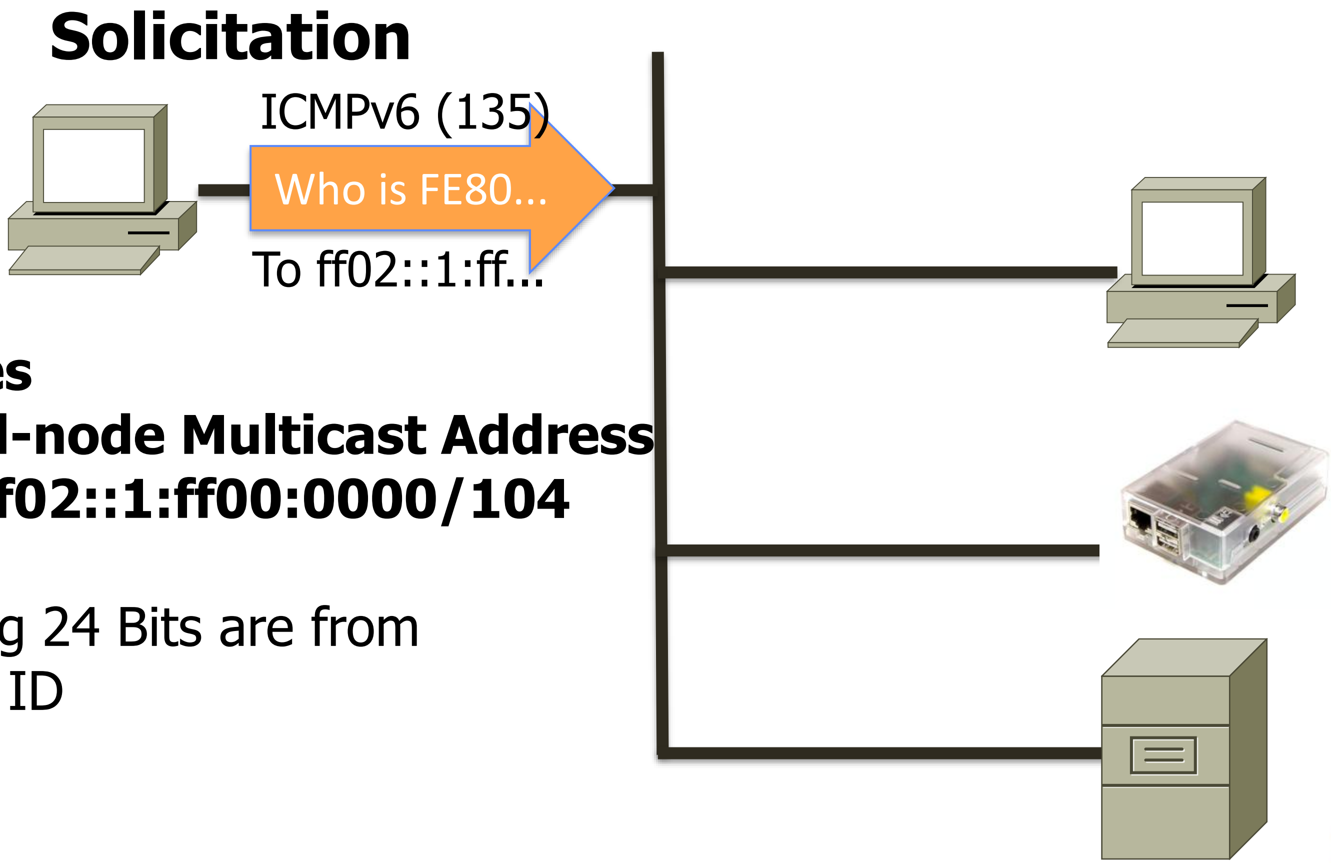


NDP uses
Solicited-node Multicast Address
Prefix: ff02::1:ff00:0000/104

HAW HAMBURG

Neighbour Discovery Handshake

- Address of my neighbour?
  Search **FE80**::**ba**27:ebFF:FE3b:7def/64

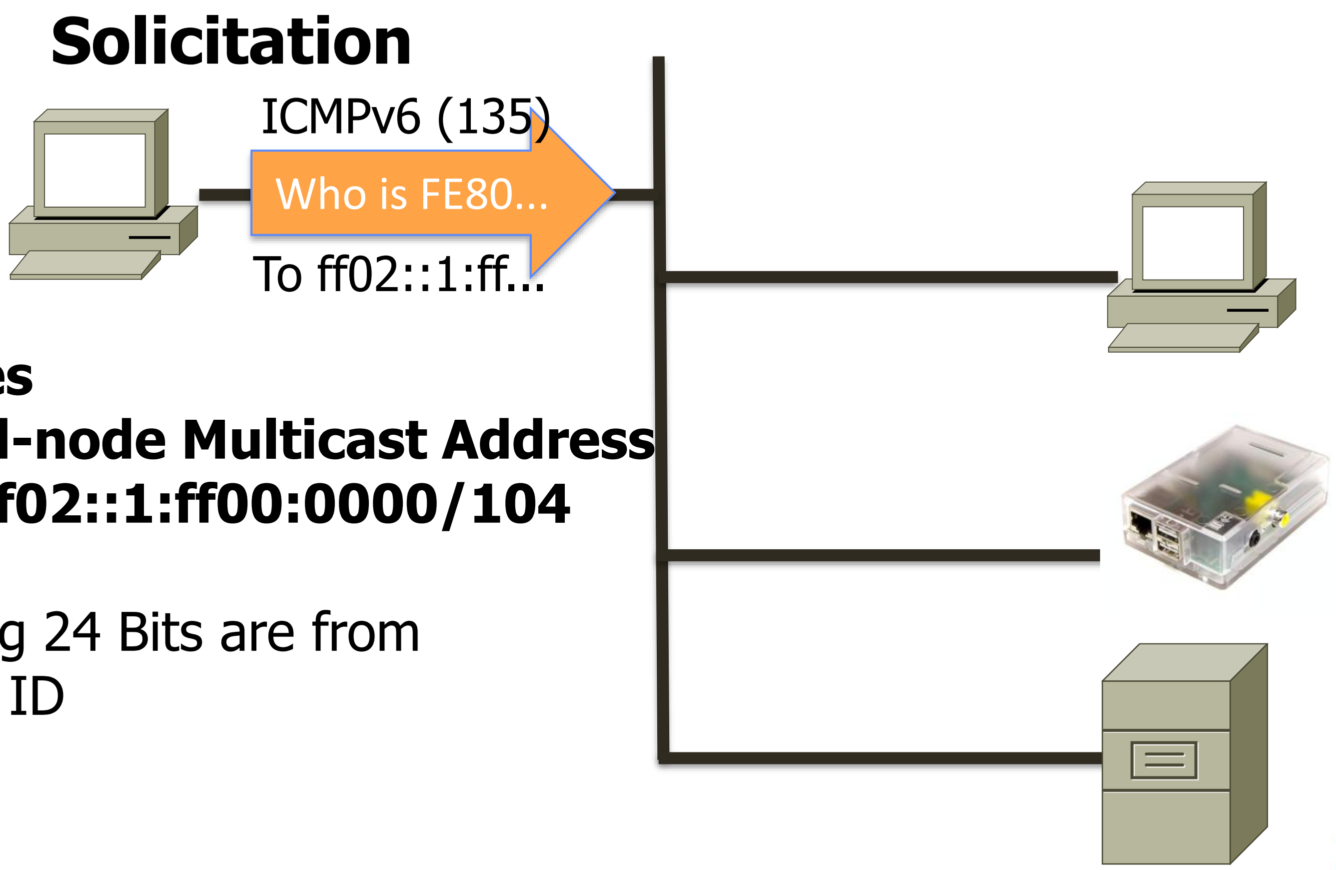


NDP uses
Solicited-node Multicast Address
Prefix: ff02::1:ff00:0000/104

Remaining 24 Bits are from
Interface ID

Neighbour Discovery Handshake

- Address of my neighbour?
  Search **FE80**::**ba**27:ebFF:FE3b:7def/64

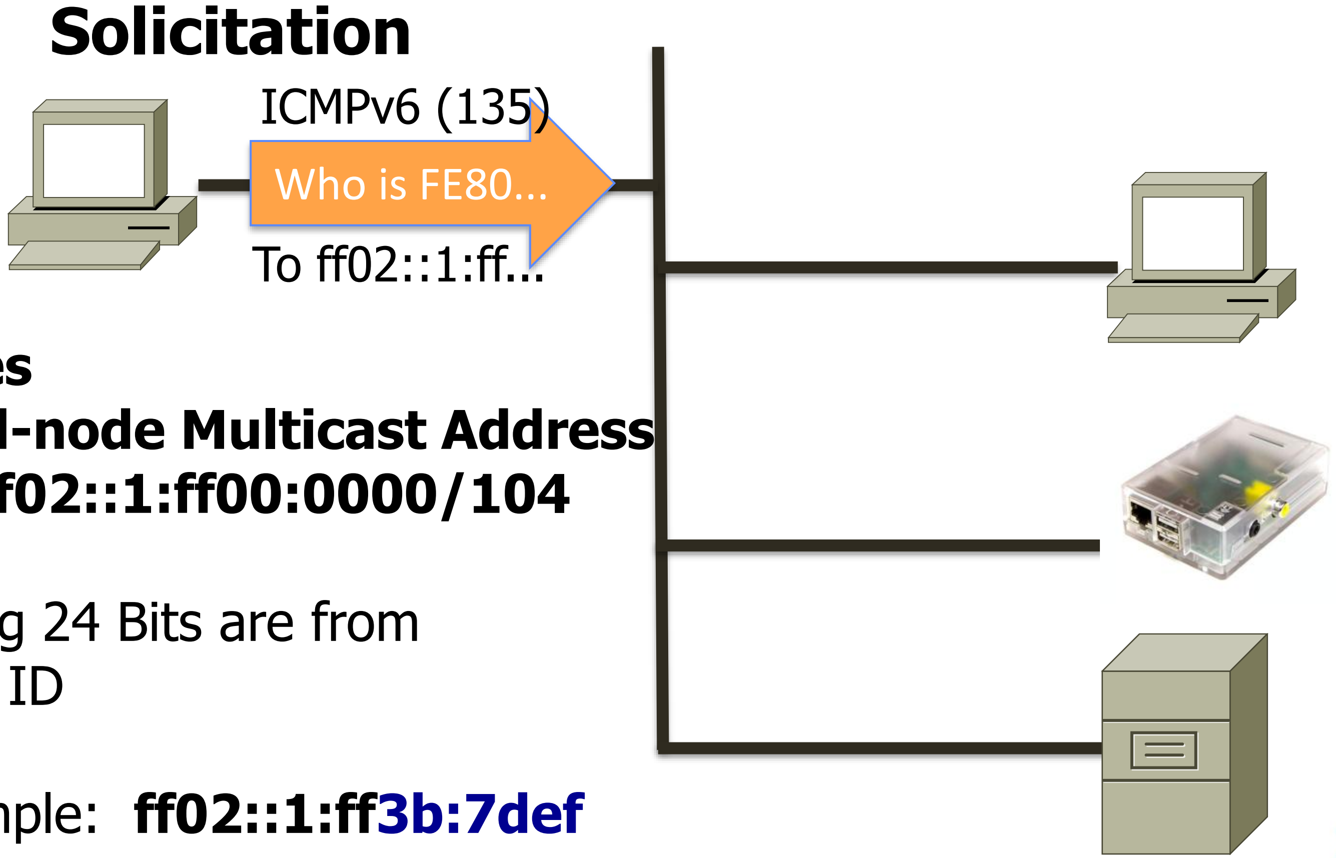


NDP uses
Solicited-node Multicast Address
Prefix: ff02::1:ff00:0000/104

Remaining 24 Bits are from
Interface ID

Neighbour Discovery Handshake

- Address of my neighbour?
  Search **FE80**::**ba**27:ebFF:FE3b:7def/64



NDP uses
Solicited-node Multicast Address
Prefix: ff02::1:ff00:0000/104

Remaining 24 Bits are from
Interface ID

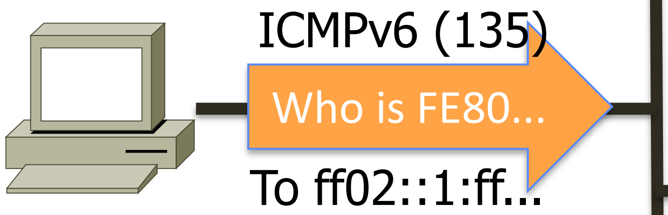
Our Example: **ff02::1:ff3b:7def**

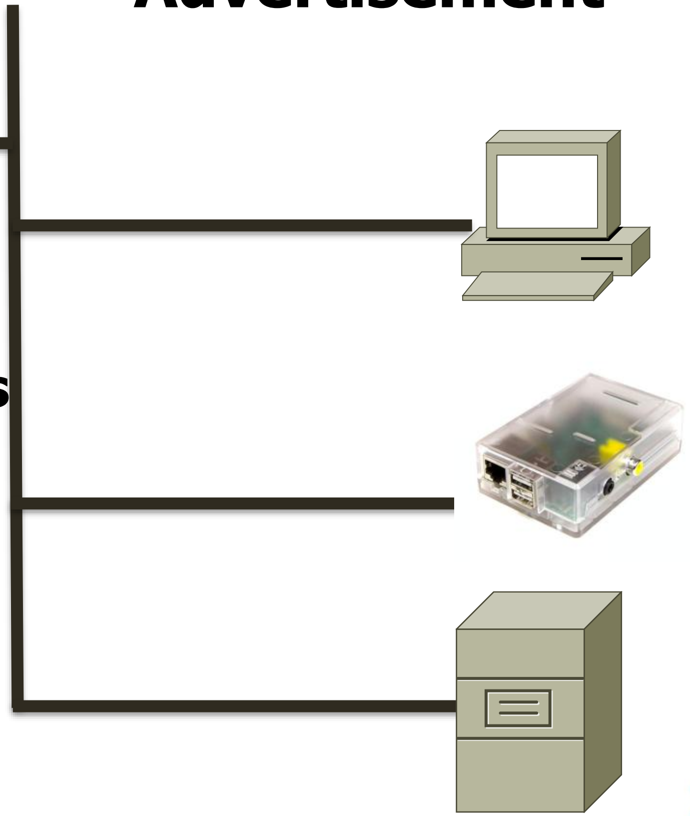
Neighbour Discovery Handshake

- Address of my neighbour?
  Search **FE80**::**ba**27:ebFF:FE3b:7def/64

Solicitation

Advertisement





NDP uses
Solicited-node Multicast Address
Prefix: ff02::1:ff00:0000/104

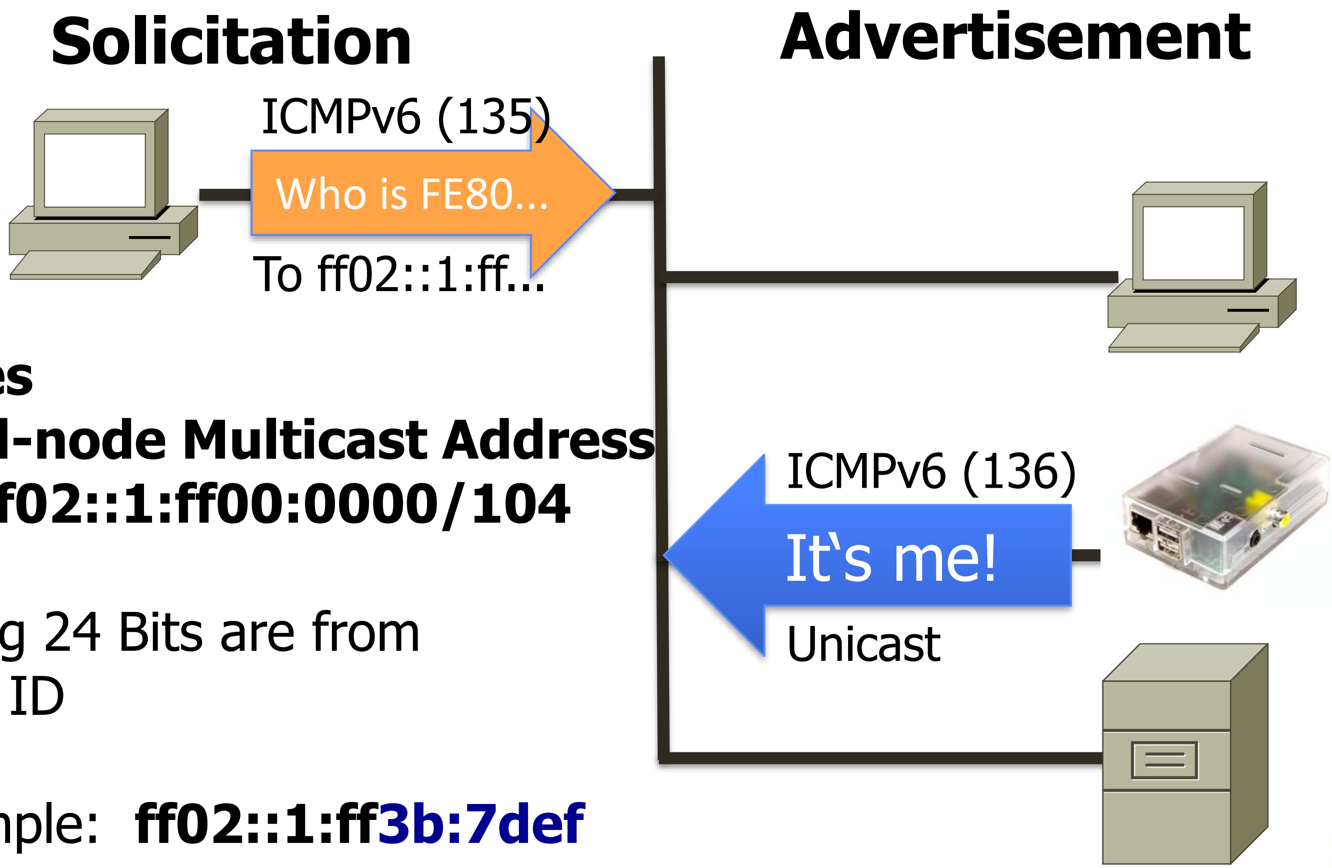
Remaining 24 Bits are from
Interface ID

Our Example: **ff02::1:ff3b:7def**

HAW HAMBURG

Neighbour Discovery Handshake

- Address of my neighbour?
  Search **FE80**::**ba**27:ebFF:FE3b:7def/64



NDP uses
Solicited-node Multicast Address
Prefix: ff02::1:ff00:0000/104

Remaining 24 Bits are from Interface ID

Our Example: **ff02::1:ff3b:7def**

Stateless Address Autoconfiguration - SLAAC

1. Interface assigns a link-local address on activation (default: EUI-64 built from a hardware address).
2. Interface sends *router solicitation,* to skip waiting for router advertisements.
3. Router sends *router advertisement* (prefix, default gateway, …).
4. The interface creates its global address from prefix and link-local address.
5. For verification of uniqueness, an ICMP *neighbour solicitation* will be issued on own address (*Duplicate Address Detection*).

Secure Neighbour Discovery - SEND

- RFC 3971 (Arkko et al.)
- Employs Cryptographically Generated Addresses (CGAs – RFC 3972) to authenticate NDP
  ("prevent ARP spoofing")

Details in MIPv6 Section

- SEND ND messages can self-consistently authenticate its IP sender address (without PKI)
- Router/prefix advertisements require certificates (X.509v3) from a PKI to assure routing responsibility

Auto-Reconfiguration - Renumbering

- New address prefixes can be distributed and retreated:
  - Coexistence periods between old and new prefixes
  - Hosts learn prefix-lifecycle and priorities via router advertisements
  - Old TCP-connections survive within coexistence periods, new TCP-connections survive prefix change
- Prefix-distribution via router renumbering: RFC 4192
- DNS-Structure (AAAA) does not follow this flexibility
- Operational complexity prevented wider deployment

IPv6 DNS

Two approaches:

- o AAAA (“Quad A”) Res. Rec. – RFC 1886, 3596
  - A record type of 128 bit length
  - Reverse lookup domain IP6.ARPA
- o A6 RR – RFC 2874
  - Idea to support renumbering by prefix delegation
  - A6 record contains a domain name pointer:

Prefix len.	Address suffix	Prefix name

- o IETF decided for AAAA – RFC 3363

HAW HAMBURG

MTU Handling

- The minimal Link-MTU for IPv6 is 1280 Bytes (versus 68 Bytes for IPv4).
- Path-MTU discovery:
  - Repeatedly send packets with known path-MTU size until no more ICMP 'Packet-too-big' is received
  - necessary only for packets > 1280 Bytes
- Minimal IP implementation can live without path-MTU discovery and without fragmentation (Packets ≤ 1280 Bytes).
- Upper Layer (TCP): MSS ≤ MTU – 60.
- A Hop-by-Hop Option supports the execution of Jumbograms with sizes up to 2^{32} Octets Payload.

Fragmentation

- IPv6 avoids the expensive fragmentation process on routers. The sender has to fragment, if necessary
- Sender should execute path-MTU discovery
- Routers answer with ICMP „Packet too big"
- IPv6 Fragmentation Header can support upper layers without dynamically executing MTU-Discovery

Next Header	Reserved	Fragment Offset	0 0 M
Original Packet Identifier			

HAW HAMBURG

Generic Packet Tunnelling of IPv6

- RFC 2473 (Conta, Deering)
- Mainly used for explicit routing path control
- Defines (stateful) end points:
  - Tunnel Entry-Point
  - Tunnel Exit-Point (at Unicast, Anycast or Multicast address)
- State variables contain MTU, Traffic Class, Flow Label
- Fragmentation may be necessary at tunnel entry point

New Header	Ext. Hds.	Original Packet (incl. Header)

IPv6 Routing

- Uses CIDR „longest prefix match"/ largest mask to go for.
- Hierarchical addressing is a key issue for a scalable routing
  - VLSM route aggregation
  - Relies on provider-bound prefixes
- Dynamical routing protocols from IPv4 updated:

  Unicast: OSPF, RIP-II, BGP4+,...

  Multicast: MLD, MOSPF, PIM, BIDIR-PIM ...
- Can send packets through predefined regions by
  - routing headers (source routing) – type 0 deprecated (RFC 5095)
  - Anycast

HAW HAMBURG

BGP Interdomain Routing

- BGP announces path vectors for IPv6 prefixes
  - IPv6 prefixes are treated as additional networks from the same AS
- BGP control-plane sessions bound to IP versions
  - IPv6 normally via TCP/IPv6, IPv4 via TCP/IPv4
  - This keeps next-hop in IP family
- IPv6 builds a routing topology parallel to IPv4
  - Paths and routers may differ
  - BGP routers hold two FIBs: IPv4 and IPv6

HAW HAMBURG

Example: BGP RIB Entry

```
TIME: 2008-7-1 02:36:49
TYPE: MSG_TABLE_DUMP/AFI_IP6
VIEW: 0  SEQUENCE: 2702
PREFIX: 2001:0638::/32
ORIGINATED: Mon Jun 30 10:29:18 2008
FROM: 2001:0418:0000:1000:0000:0000:0000:f000 AS2914
AS_PATH: 2914 3549 680
MULTI_EXIT_DISC: 1
COMMUNITIES: 2914:420 2914:2000 2914:3000
```

Inter-AS Link Metric
Priorising redundant
Interprovider Peerings

Routing Policy Group
Groups type of propagation

The Secrets of BGP

see separate BGP lecture

Anycast

- Service to ‘one out of many’ , RFCs 4291, 4786
- Addressing from unicast
  - Prefix must cover all group members (least common)
  - Reg. Global Unicast, but some reserved:
    - Subnet (RFC 2526): subnet router, MIPv6 home agent
    - RFC 3068: 6to4 relays anycast address
    - RFC 4610: anycast rendezvous point in PIM
- Anycast Routing
  - Within IGP: Host routes (dynamic/static)
  - Globally: Announcement of covering prefix, internally replicated instances, some global ASNs for Anycast

Anycast (2)

- There are transport issues with Anycast:
  - Subsequent packets may not always reach the same peer
  - Source address may never be anycast, so a unicast address must be used in a response
  - → TCP and IPSec cause conflicts, UDP deployable
- Today's deployment:
  - Use for routers only
  - Local / regional configuration
  - DNS root servers
  - 6-to-4 transition: Tunnel relay prefix 192.88.99.0/24
  - Multicast: PIM rendezvous point anycast addressing + AMT

IPv6 & QoS

Priority: Traffic Class Field (8 Bit) break down in two classes:

- Flow controlled traffic (0 - 7) – IPv4 compatible

0	Not specified	4	Bulk (i.g. ftp, http)
1	‚Feeder' (i.g. netnews)	5	(Reserved)
2	Unnoticed (i.g. email)	6	Interactive (i.g. telnet, X11)
3	(Reserved)	7	Internet control (i.g. rip)

- Traffic without flow control (Realtime, Constant Bitrate, ...)
  Priority from 8 to 15 (ascending)
- Remaining priorisation according to DiffServ code points

HAW HAMBURG

Flow Labels

20-bit Flow Labels can be used by senders or ingress routers to mark associated packets.

- initial RFC 3697, now RFC 6437
- Goal: accelerated, uniform handling of packet streams through routers
- Flow label assignment: Random per Flow
- Header information consistent per flow (router caching)
- Initially defined router states, now stateless processing

Agenda

- Motivation
- Basic IPv6 Architecture
- IPv6 Migration: Transition and Coexistence
  - Programming: IPv6 API
  - Dual Stack
  - Tunnelling
- Future Trends: Beyond IPv6

IPv4 → IPv6 Porting

- Source and binary code compatibility for existent application: ‘all goes on’, provided IPv4 is around
- Indirection for address data structure, new for IPv6: `addrinfo` is linked list of interface address structs
- Name-to-address translation:
  New functions to support IPv6 and IPv4
- Address conversion functions:
  New functions to support IPv6 and IPv4
- DNS resolver:
  Returns IPv6 or IPv4 address, or both

IPv4 → IPv6 Porting: API Changes

	IPv4	IPv6	
Data structures	AF_INET	AF_INET6	
	in_addr sockaddr_in	in6_addr sockaddr_in6	
Address conversion functions	inet_aton() inet_addr()	inet_pton()	IPv4 and IPv6 functions
	inet_ntoa()	inet_ntop()	
Name-to-address functions	gethostbyname() gethostbyaddr()	getnameinfo() * getaddrinfo() *	

* **POSIX protocol independant functions**

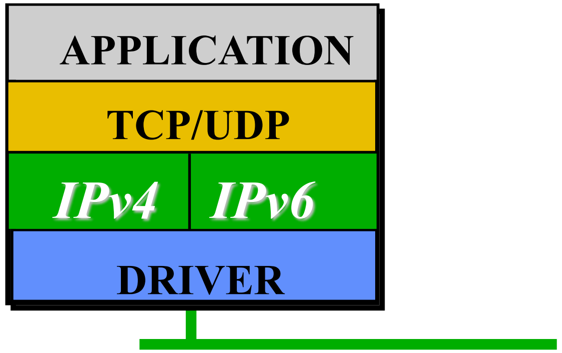
IPv4 → IPv6 Migration

Many migration techniques have been designed and implemented according to the following approach:

- Dual-Stack techniques, which allow the coexistence of IPv4 and IPv6 at the same device and subnet
- Tunnel, which connects IPv6 regions over IPv4 regions
- Protocol translator, which let IPv6 devices with IPv4 devices speak

During migration, the combined use of all these methods is likely.

Dual Stack

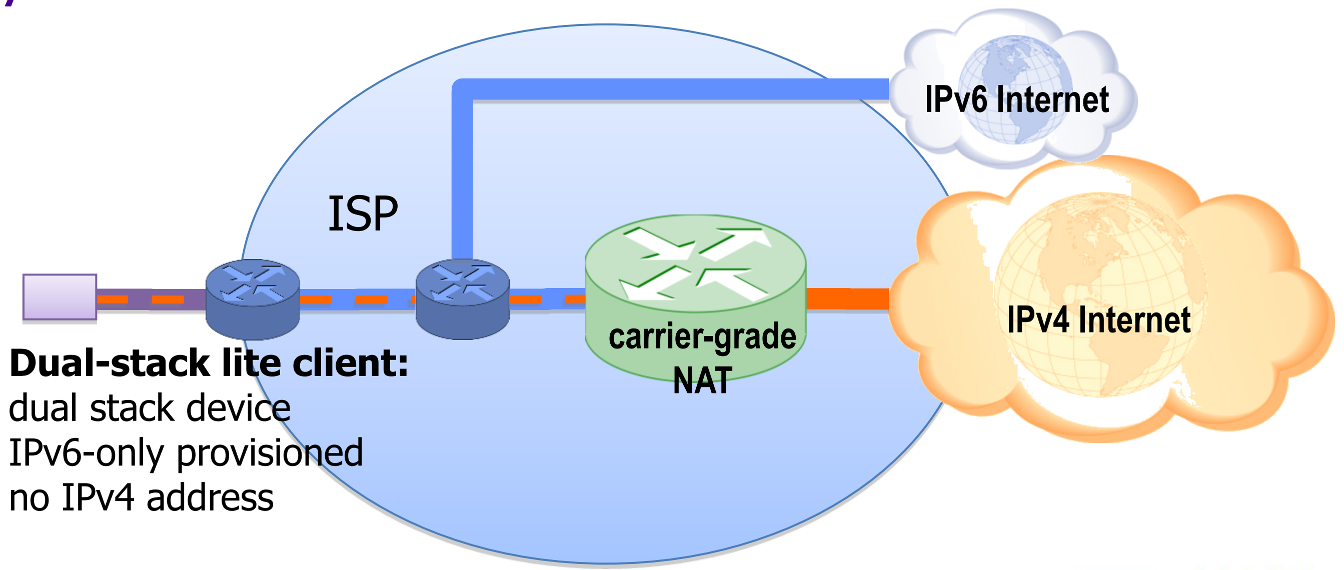


- On activation of IPv6, IPv4 can continuously be used (multi protocol approach)
- Devices can keep their addresses (IPv4 in IPv6)
- Application / libraries choose the IP version:
  - following DNS answers with a IPv6 preference
  - in response to received packets
- The Dual stack operation can continue without limits, allowing the step by step porting of applications
  - But requires an increasing use of IPv4-NATTING (recursively)
- Problems result from inconsistencies of Network / DNS configurations

Dual Stack Lite

- RFC 6333 -

- IPv6-only at customer – but dual stack at system
- IPv4 over IPv6 tunnel to Carrier-grade IPv4-IPv4 NAT
- IPv4 access via globally unique IPv4 address shared among many customers

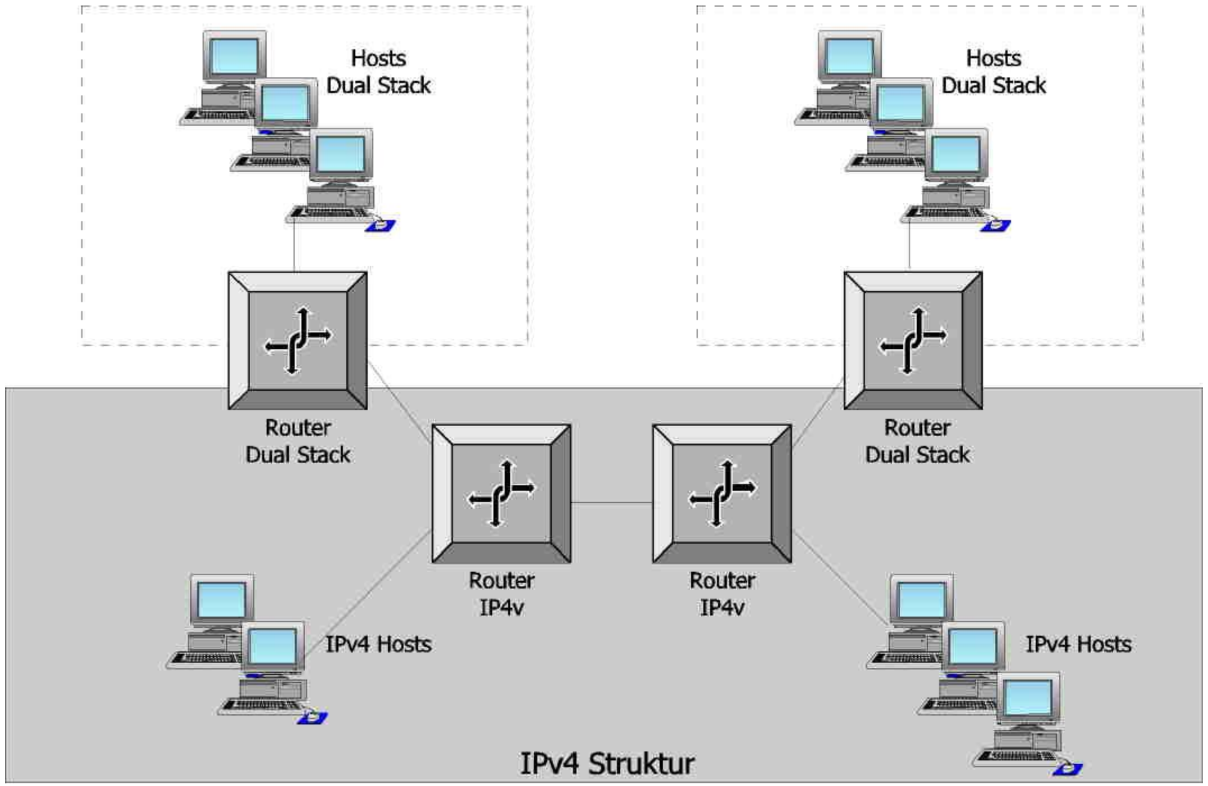


HAW HAMBURG

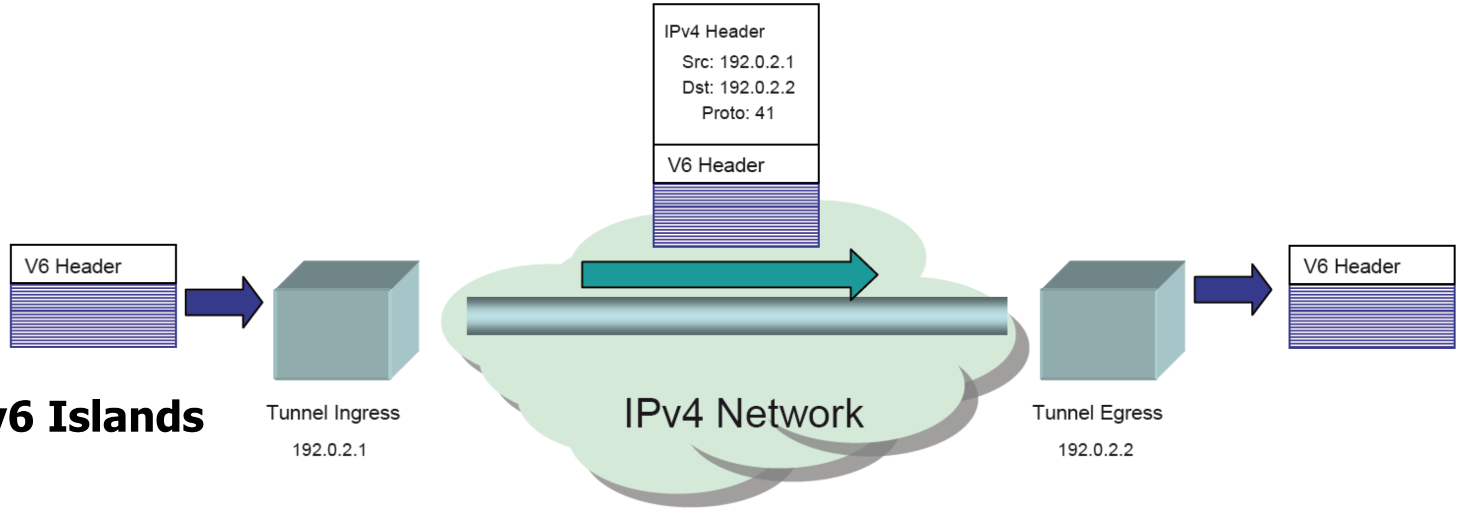
Transition by Tunnels

- o Embedding of IPv6 packets in IPv4 packets
- o Diverse methods to build a tunnel:
  - Manually
  - Tunnel Broker (web based application to build a tunnel)
  - 6-over-4 (intradomain)
  - 6-to-4 (interdomain, IPv4 address as IPv6 site prefix)
- o Perspective:
  - IPv6 use IPv4 as a virtual link layer
  - IPv6 VPN over IPv4

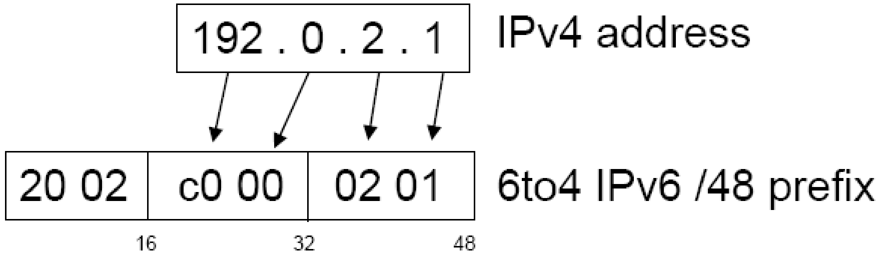
6-over-4 Tunnels



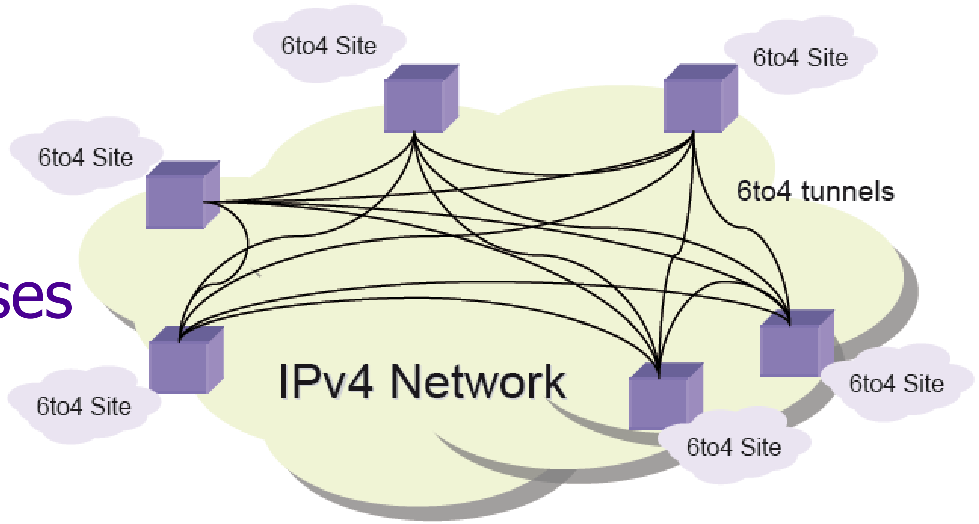
- Isolated IPv6 islands in an IPv4 world
- IPv6 ↔ IPv4 inter-connects: Embedding
- IPv6 ↔ IPv6 inter-connects: Following tunnel configurations



6-to-4

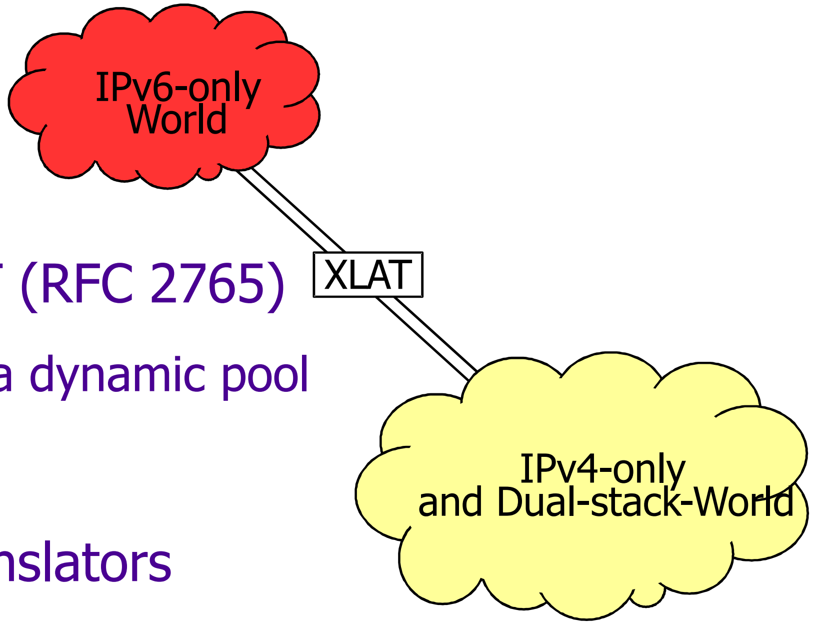


- Defines automated point-to-multipoint tunnels, RFC 3056 ff.
- Assigns an IPv6 network to each IPv4 address (taken as prefix)
- Allows IPv6 islands to automatically interconnect, using IPv4 as a non-broadcast multi-access network
- Automatic tunnel access via well-known Anycast addresses



HAW HAMBURG

Protocol Translation



- Early stateless protocol translator: SIIT (RFC 2765)
  - Used temporary IPv4 addresse from a dynamic pool
- Current familiy of IP/ICMP protocol translators
  - XLATs: RFC 7915, addressing: RFC 6052
  - Stateful and stateless translation schemes
  - Algorithmic IPv4 embedding behind well-known prefix: 64:ff9b::/96
- Scenarios:
  - New IPv6-only domains at Internet edge
  - Old IPv4-only network islands

Agenda

- Motivation
- Basic IPv6 Architecture
- IPv6 Migration: Transition and Coexistence
- Future Trends: Beyond IPv6
  - Lessons Learned
  - IRTF/IETF Work
  - Some Visionary Proposals

Lessons Learned

- IPng deployment
  - Internet is inert "victim" of IPv4 success
  - Internet is "victim" of its uniqueness constraint
- Development of innovations within the Internet
  - Clear trend: broadening the range of applications
  - Unclear: "who steers the IP layer" – Pipe owners pushing packets versus user-driven, intelligent layer 3 services
- Internet design – Quo Vadis?
- Routing – *the* scaling issue
- DNS – inflexible, updates too slow, *but scalable*

Issue: Economic Models for Internet Operation

- Several players:
  - Infrastructure providers (pipes, routers, hosting)
  - Over the Top Provider (international applications & services)
  - Customer-oriented / regional ISPs
  - Edge domain operators (companies, consumer-ISPs)
- Problem: Grouping according to regions, not services
  - Little room for service innovation / dedication
- Approach: Virtualization of infrastructure
  - Allows for slicing of cables (λs), routers and servers
  - Offers playground for specialized service provisioning

HAW HAMBURG

Issue: Naming & Addressing

- IP addresses carry a dual meaning of Identifier (who) and locator (where)
  - Initially ID=name, Loc=address [Cohen, D., "On Names, Addresses and Routings", IEN 23, 1978]
  - TCP used addresses as identifiers (DNS was not ready)
- Mobility re-raises the problem
  - MIPv6 creates an ID/Locator split on network layer
- Host Identity Protocol (HIP, RFC 5201) introduces abstraction layer between network and transport

Issue: Route Scalability

- Routing table size and dynamics in the core increases due to growing numbers and heterogeneity at edge domains
- Two counter measures:
  - Aggregation – PA addressing as proposed by IPv6
  - Separation – decoupling of edges from core by a Map-Encapsulate mechanism
- Ongoing debate
  - Map/Encap opens freedom of design at edge domains

IRTF / IETF Work

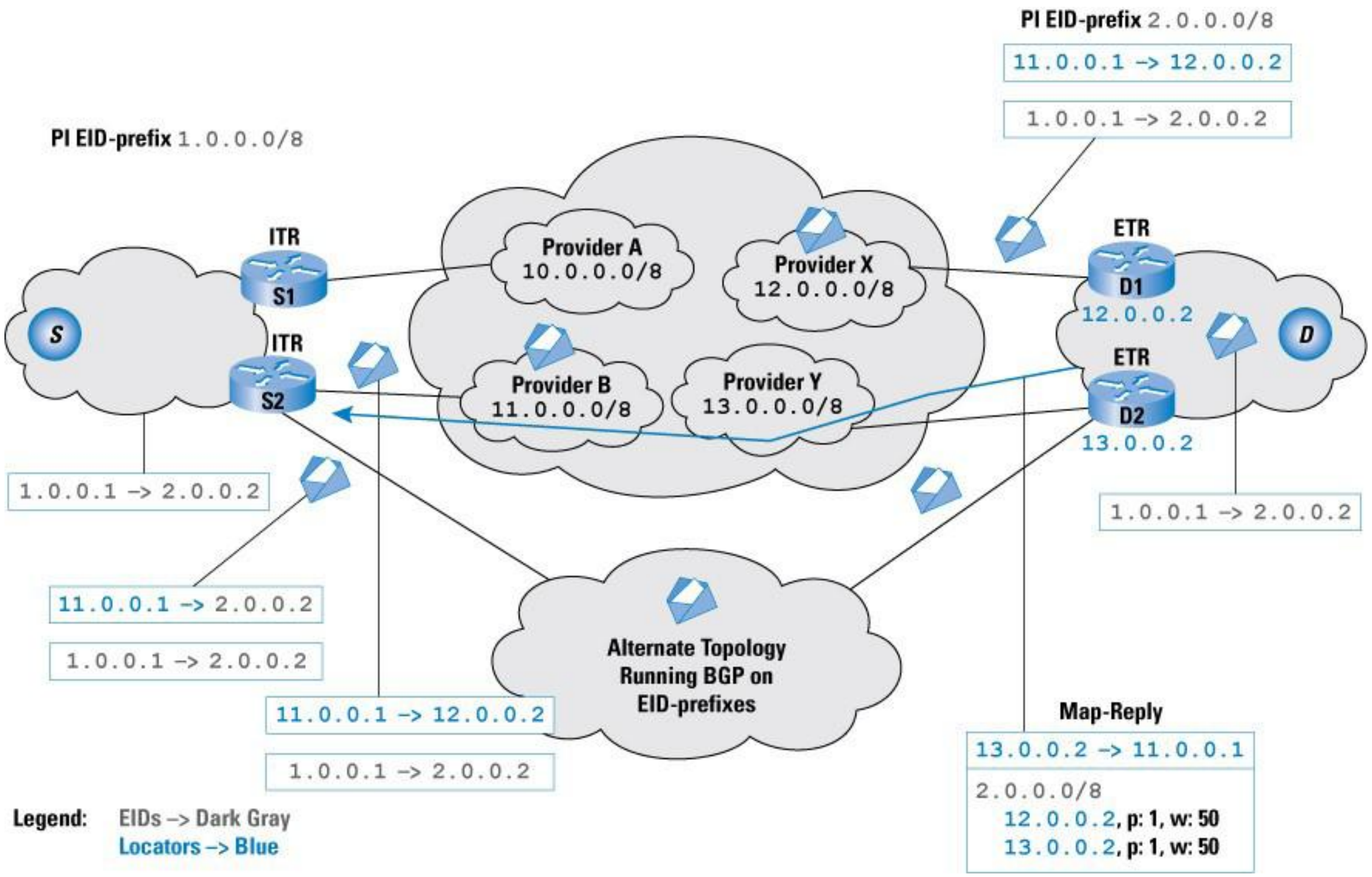
- IAB Workshop on Routing and Addressing
  s. draft-iab-raws-report-00
- From 2007 on, the Routing Research Group called for and discussed proposals
  s. http://trac.tools.ietf.org/group/irtf/trac/wiki/RoutingResearchGroup
- RRG made a decision March '10 and published recommendations in RFC 6115 (Feb. `11)

Locator ID Separation Protocol (LISP)

RFC 6830

- Sites / nodes have addresses (as in DNS): provider independent IDs
- Routers maintain a database of "Routing Locators (RLOCs)": provider-bound IP addresses of routers in a destination site
- Packets are tunnelled from egress to ingress router using RLOCs
- Problem: Router databases … to be learned from ICMP advertisements in LISP
- Result: Removes load from core routers

LISP Scenario



Name-based Socket

`draft-ubillos-name-based-sockets`

- o Straight-forward approach
  - DNS Name is Identifier
  - IP Address is Locator
  - Initial name exchange
    (for backward compatibility)
- o Requires change of Socket API
  - New address family (AF_NAME)
- o Issue: Bootstrapping – how to communicate without DNS?

HAW HAMBURG

Identifier Locator Network Protocol (ILNP)

RFCs 6740-6748

- Idea (Clark/O'Dell ~1995) to split IPv6 addresses
  - Routing Locator – in IPv6: first 64 bits
  - Unique Identifier – in IPv6: second 64 bits
- Locator and Identifier independently stored in a (modified) DNS
  - Additional records for resolution
  - Dynamic update function for locators at DNS
- New, high-level API

HAW HAMBURG

ILNP: Re-interpretation of the IPv6 Header

```
+---------------+----------------+---------------+---------------+
| Version| Traffic Class |        Flow Label                      |
+---------------+----------------+---------------+---------------+
| Payload Length                 | Next Header   | Hop Limit     |
+---------------+----------------+---------------+---------------+
|                         Source Locator                         |
+                                                                +
|                                                                |
+---------------+----------------+---------------+---------------+
|                         Source Identifier                      |
+                                                                +
|                                                                |
+---------------+----------------+---------------+---------------+
|                      Destination Locator                       |
+                                                                +
|                                                                |
+---------------+----------------+---------------+---------------+
|                       Destinaion Identifier                    |
+                                                                +
|                                                                |
+---------------+----------------+---------------+---------------+
```


HAW HAMBURG

Aggregation in Increasing Scopes
`draft-zhang-zhang-evolution`

- Idea: Increase aggregation on global scale on the price of possible detours
  - Reduce global prefix resolution, e.g., route to 145.01/15 to reach an Aggregation Point
  - Provide detailed knowledge (full prefix resolution) at Aggregation Point and tunnel packets over the last hops

HAW HAMBURG

Recommendations of the RRG

- Highly controversial debate lasting over more than 3-Years
- Recommended for further development to the IETF
  1. Aggregation in Increasing Scopes
  2. ILNP
  3. Renumbering

HAW HAMBURG

Clean Slate Internet Initiatives in early 2000s

- Outside IETF/IRTF
- Focus scaling: overall logarithmic … as of DHTs (i^3)
- Focus virtualisation:
  Programmable virtual infrastructures for dedicated purposes
- Focus pluralisation/federation:
  Break with the paradigm of a universal network layer
- Focus programmability:
  - Software-defined Networking
  - Software-defined Radios
- Major US initiative: GENI
- EU: Future Internet Research and Experimentation – FIRE initiative
- Germany: German LAB (G-LAB)

Bibliography

- Pete Loshin: IPv6 – *Theory, Protocol and Practice.* Elsevier, 2004.
- 6Net Consortium: *An IPv6 Deployment Guide*, Sept. 2005.
- Benedikt Stockebrand: *IPv6 in Practice,* Springer, 2010.
- Qing, Li / Jimnei, Tatuya / Shima, Keiichi: IPv6 Core / Advanced Protocols Implementation, Morgan Kaufmann, 2007.
- Zach Shelby, Carsten Bormann: *6LoWPAN: The Wireless Embedded Internet,* 2nd Ed., Wiley & Sons, 2011.
- Dave Thaler (IAB): *Evolution of the IP Model*, draft-iab-ip-model-evolution, 2008.
- Tony Li: *Recommendation for a Routing Architecture*, RFC 6115, 2011 .
- Dan Jen et al.: *Towards A New Internet Routing Architecture: Arguments for Separating Edges from Transit Core*, ACM Hotnets, 2008.
- Nick McKeown: *Software-defined Networking*, Infocom Keynote Talk, Rio de Janeiro 2009
- Drafts, RFCs: tools.ietf.org, http://www.rfc-editor.org

HAW HAMBURG