Copyright Notice

This set of transparencies, hereinafter referred to as slides, is protected by copyright laws and provisions of International Treaties. The title and copyright regarding the slides (including, but not limited to, each and every image, photography, animation, video, audio, music and text) are property of the authors specified on page 1.

The slides may be reproduced and used freely by research institutes, schools and Universities for non-profit, institutional purposes. In such cases, no authorization is requested.

Any total or partial use or reproduction (including, but not limited to, reproduction on magnetic media, computer networks, and printed reproduction) is forbidden, unless explicitly authorized by the authors by means of written license.

Information included in these slides is deemed as accurate at the date of publication. Such information is supplied for merely educational purposes and may not be used in designing systems, products, networks, etc. In any case, these slides are subject to changes without any previous notice. The authors do not assume any responsibility for the contents of these slides (including, but not limited to, accuracy, completeness, enforceability, updated-ness of information hereinafter provided).

In any case, accordance with information hereinafter included must not be declared.

In any case, this copyright notice must never be removed and must be reported even in partial uses.
Outline

→ A new version of IP: why?
→ Addresses
→ Modified protocols
→ Socket programming interface
→ Neighbor discovery
→ Transition to IPv6(?)
A NEW VERSION OF IP: WHY?
Why a new IP?

Only one true answer

A larger address space
Other answers

→ More efficient on LANs
→ Multicast and anycast
→ Security
→ Policy routing
→ Plug and play
→ Traffic differentiation
→ Mobility
→ Quality of service support

Ported to IPv4
A long way to IPv6 adoption

➤ Long period for defining and migrating to IPv6

➤ Problems needed an interim solution in IPv4

➤ When IPv6 reached "production" stage, many IPv4 solutions were acceptable
Why are IPv4 addresses scarce?

32 bit long

About 4 billion addresses!!!

however ...
Only part of the addresses are assigned to stations

→ Class A, B and C

→ Addresses beginning by bx111 are used for multicast and else

Hence, “just” 3.5 billion addresses can be used!!!
They are used hierarchically

→ The prefix used in a physical network cannot be used in a different one

→ Lots of unused addresses
Interim (IPv4) solutions to the saturation of address space

- Introduction of network with "taylored" size
- Netmask
- Private addresses
  - Intranet, RFC 1918
  - Not enough to solve the problem
  - It should be used in conjunction with NAT or ALG
- Network Address Translator (NAT)
  - Extremely popular
- Proposal for RSIP (Realm Specific IP)
- ALG (Application Level Gateway)
Was all of this effective?
Agencies assigning addresses

IANA distributes (better: distributed) /8 IPv4 network prefixes to regional registries

- ARIN
- LACNIC
- AfriNIC
- RIPE NCC
- APNIC
Situation (2010)

IPv4 Address Pool Status

- IETF_Reserved: 35.0782
- IANA_Pool: 20
- IANA_Registry: 2
- VARIOUS: 47.9218
- AFFINIC: 2
- APNIC: 40
- ARIN: 70
- RIPECC: 33
- LACNIC: 6
Situation (2011)

IPv4 Address Pool Status

- IETF_Reserved: 35,0762
- AFRINIC: 12,9961
- APNIC: 55
- ARIN: 83,9257
- RIPENCC: 49
- LACNIC: 20
Routing scalability issues

► Routing table size
► Internet size
► Each subnetwork must be advertised

► Problems
► Router resource limitations
► Too much information to manage
► Routing protocol limitations
► High probability of route changes
► Mainly affecting backbone routers
Routing scalability issues

http://bgp.potaroo.net/
Isn’t there a solution with IPv4?

- Aggregate multiple routes in one
  - Shorter prefix including others
  - 1.2.1.0/24, 1.2.2.0/24 ...
  - 1.2.0.0/16

- CIDR (Classless Inter-Domain Routing)

- Limited by non-rational assignment of IP prefixes
Interim (IPv4) solutions to routing scalability

- CIDR
- Classless Inter-Domain Routing
- Limiting the assignment of IP addresses
- Regional Internet Registry: assign address blocks only to big players
- E.g., minimum /20 (4096 addresses) network

- Scalability of routing protocols
- With no solution, at present
- Problem not completely solved
- It is the major problem that IPv6 wanted to solve that it is still open
Birth of IPv6

- IETF Boston Meeting (1992), “Call for proposals”
- Appointment of dedicated Working Groups
- Several proposals
  - TUBA: adopting OSI CNLP as new IP
  - CATNIP: integration of different network (IP, CLNP, IPX) and transport (TP4, SPX, TCP, UDP) protocols
  - SIPP: incremental over IPv4
    - Fix some drawbacks
    - Simple: increasing the address field and eliminating unused ones
- Winning proposal: SIPP with 128 bit addresses
So, how many addresses should IPv6 have?

→ A scientific approach
→ Addressing efficiency

\[ H = \frac{\log_{10} \text{(number of addresses)}}{\text{number of bits}} \]
Addressing Efficiency

→ In existing networks
  → H varies between 0.22 and 0.26

→ Assuming one million billion networked stations
  → 68 bits in the minimum efficiency case
Melius abundare quam deficere

128 bits (16 bytes)

655,570,793,348,866,943,898,599 IPv6 addresses per sqm of Hearth surface
Notation

8 hexadecimal numbers separated by "."

Groups of 2 bytes

FEDC:BA98:0876:45FA:0562:CDAF:3DAF:BB01
1080:0000:0000:0007:0200:A00C:3423:A089
Shortcuts

Leading Os in each digit group can be omitted

$\rightarrow$ 1O8O:O:O:7:2OO:AOOC:3423

Groups of Os can be substituted by “::”

$\rightarrow$ 1O8O::7:2OO:AOOC:3423
Addressing Space Organization

- Multicast
- 1111 1111
- FFxx:...
→ Link local/site local
  → 1111 111O  1
→ Link local
  → 1111 111O  1O
→ FE8O::/64
→ Site local (deprecated)
  → 1111 111O  1O
  → FEC0::/1°
→ FE[C-F]…
Private Addresses

→ Equivalent to IPv4 private addresses

→ 1111 1101

→ FD/10
# Local Unicast Addresses

## Link Local

- **Prefix:** 1111-1110-10
- **Subnet ID:** 0
- **Scope:** Interface ID
- **Address Format:** FE80

## Site Local

- **Prefix:** FE[C-F]x
- **Subnet ID:** any
- **Scope:** Interface ID

## Private

- **Prefix:** 1111-1101
- **Subnet ID:** randomly generated
- **Scope:** Interface ID
- **Address Format:** FD
Remaining addresses

Global Unicast
Global Unicast Addressing Space Organization

- Addresses for IPv4 interoperability
  - 0...0 (80 bit) → 0::/80

- To be used during transition phase

- IPv4-mapped addresses
  - 16 bit a 1 → 0:0:0:0:FFFF::/96
IPv4-compatible

Another 16 bits to 0 → 0::/96
E.g. 0:0:0:0:0:0:A00:1

Compact notation

::A00:1

Special notation

::10.0.0.1
Aggregatable Global Unicast

→ Begin with bxOO1
→ Topology-based assignment
→ Service provider hierarchy
→ Effective aggregation
Different assignment criterion for other addresses
Same routing principles as IPv4
Address Structure

- Prefix: n bit
- Interface Identifier: 128 - n bit
Same Address Assignment Principles as IPv4
(different terminology)

→ Sub network: set of hosts with same prefix

→ Link: physical network

Subnetwork $\equiv$ link
→ On-link hosts have same prefix

→ Communicate directly

→ Off-link stations have different prefix

→ Communicate through a router
Prefix

Address/netmask pair is substituted by a “Prefix”

⇒ FEDC:0123:8700::/36
⇒ 1111111011011100
00000001001000111000

No address classes
Address Assignment

IPv6 - 44
Plug and Play

Scenarios
→ Dentist Office
→ Thousand computers on the dock

Solution: autoconfiguration
→ Stateless: no server needed
→ Statefull: DHCP server
MODIFIED PROTOCOLS
What changes in the protocol architecture?

→ IP
→ ICMP
→ ARP
→ IGMP
→ Integrated in ICMP
→ Integrated in ICMP
Upgraded But Not Changed

- DNS (type AAAA record)
- RIP and OSPF
- BGP and IDRP
- TCP and UDP
- Socket interface

What about layer independence?
SOCKET PROGRAMMING INTERFACE
What is it?

- Programming interface for TCP/IP services
- Used in application implementation
- UDP messages
- Bytes on TCP connections
Underlying Principles

→ Originated in Unix Environment
→ I/O as file access
→ Socket descriptor equivalent to a file descriptor for network use
Socket

- Point of access to network services
- Associated to TCP connection or UDP session
Socket Operations

→ Wait for connection requests on a port
→ Server
→ `listen()`

→ Accept requests (server)
→ Connect to a port of a remote server
→ Client
→ Requires specifying **address** and port
→ Send and receive data
PACKET HEADER FORMAT
Do You Remember the IPv4 Header?

- **VER**
- **HLEN**
- **ToS**
- **Total Length**
- **Identifier**
- **Protocol**
- **Checksum**
- **TTL**
- **Source Address**
- **Destination Address**
- **Options**
- **PAD**
- **Fragment Offset**
- **ToS**
- **Total Length**

IPv6 - 56
Here is the IPv6 One

IPv6 - 57

© M. Baldi: see page 2
Field Removal

- Not very useful
- Checksum
- Not used in each packet
- Fragmentation
- No longer needed
- Header length
Extension Headers

→ Added when useful
→ Not needlessly processed in each packet
Extension Headers

→ Hop By Hop Option
→ Routing
→ Fragment
→ Authentication
→ Encrypted Security Payload
→ Destination Option
Extension Header Format

- Next Header
- Length
- Extension data

More extension data

More extension data
Header Chaining

- IPv6 Header N.H.=TCP
- TCP Segment

- IPv6 Header N.H.=Routing
- Routing header N.H.=TCP
- TCP Segment

- IPv6 Header N.H.=Routing
- Routing header N.H.=Fragm.
- Fragm. header N.H.=TCP
- TCP Segment
NEIGHBOR DISCOVERY
New Function in ICMP

→ It substitutes ARP
→ Based on multicast
→ Most likely only one station gets involved
Solicited Node Multicast Address

- Subscribed by all hosts
- FF02::1:FF/104 | 24 least significant bits of IP address
- Likely 1 host per group
IPv6 Multicast Transmission

→ Based on MAC multicast
→ IPv6 multicast address mapped to MAC address
→ 33-33 | 4 least significant bytes of IPv6 address
Address Resolution

→ ICMP Neighbor Solicitation
→ To Solicited Node
 Multicast Address
 of target IPv6 address
→ ICMP Neighbor Advertisement
→ To requester address
Resolution Example

- To find the MAC address of host 2001:ABCD:EF98
- ICMP Neigh Sol to Sol Node Mult Add: FF02::1:FFCD:EF98
Host Cache

→ Mapping between IPv6 and MAC address

→ Equivalent to ARP cache
TRANSITION TO IPv6 (?)
IPv4 to IPv6 Transition

→ Incremental
→ Seamless
→ Smooth
How can we enable this?

→ Dual-stack approach
→ IPv6 as a new protocol
→ Generate/receive v6 or v4 packets as needed
→ Address mapping
→ Tunneling
→ Translation mechanisms
IPv6 Islands Grow

IPv6-only Hosts

IPv6

Dual-stack Translating Devices

IPv6

IPv4

IPv6

IPv6

IPv6 Islands Grow
Native IPv6 Connectivity

IPv6-only Hosts

Dual-stack Translating Devices

IPv4

IPv6

IPv6
All the Way to the Doomsday

IPv4 in IPv6 Tunnel
Are we ready?

➔ All protocol specified

➔ For a while: since 1996!!
IPv6

- Implemented on routers
- Even if less stable than IPv4
- Possibly not all functionalities
- Some hardware implementations (Layer 3 switch)
IPv6

- Implemented in end systems
- Windows since 2000 and XP
- Unix, FreeBSD, Linux
- Quite a few applications
- Possibly with a few bugs
When will it happen?

→ Large IPv4 install base
→ Only one true motivation:

Address space depletion
The issue has been mitigated

- Provident address assignment
- Extensive use of private addressing
- NAT and proxying
So, don’t we need IPv6?

→ NAT not suitable for all applications

→ Problematic with security mechanisms
User traceability
Not practical with servers
Not many → public addresses
Acceptable limitations so far
Just Plain Address Space Exhaustion

- Especially in the Asia-Pacific region
- IANA ran out of class A prefixes in Feb 2011
- RIPE by end 2011

Possibly legislation
Current IPv6 web deployment

www.vyncke.org