The Network Layer

IP addressing

Lecture given by Emmanuel Lochin

ISAE-SUPAERO

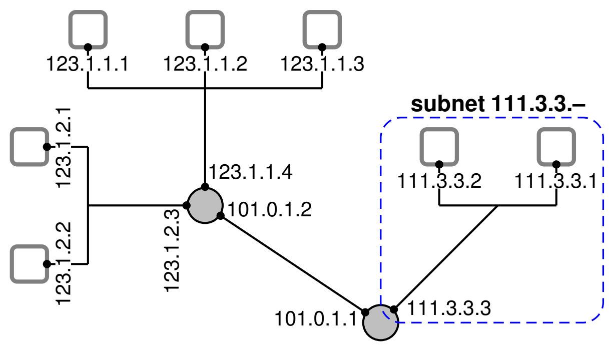
Original slides from A. Carzaniga (Univ. Lugano)
Extended/modified by E. Lochin (ISAE-SUPAERO) with author permission

Textbook Chap. #4 Sections 4.4.2, 4.4.4

Outline

- IPv4 Addressing
  - network addresses
  - classless interdomain routing
  - address allocation and routing
  - longest-prefix matching
- IPv6
  - motivations and design goals
  - datagram format
  - comparison with IPv4
  - extensions

Interconnection of Networks



IPv4 Addressing

- 32-bits **addresses**
- Written in "dotted quad" notation i.e. four 8-bits numbers separated by dots e.g $a.b.c.d$ where $\{a, b, c, d\}$ are 2^8 numbers
- Addresses are allocated in blocks called prefixes $\Rightarrow$ a prefix (denoted L) corresponds to a subnet
- You have then 2^{32-L} hosts addresses for a given prefix and is denoted $a.b.c.d/L$
  - For historical reason you should use $2^{32-L} - 2$ addresses only (first is the network itself, last the broadcast address)
- The prefix is called netmask when written in "dotted quad" notation as shown below for $L = 24$:

$$192.168.1.1 \quad \overbrace{11000000 \quad 10101000 \quad 00000001}^{\text{subnet}} \quad 00000001_{\text{two}}$$

$$255.255.255.0 \quad \overbrace{11111111 \quad 11111111 \quad 11111111}^{\text{netmask}} \quad 00000000_{\text{two}}$$

Netmask

- Network addresses, **mask** notation : address/mask
- A prefix of length p corresponds to a mask

$$M = \overbrace{11\cdots1}^{p \text{ times}}\;\overbrace{00\cdots0}^{32-p \text{ times}}\;{}_{\text{two}}$$

  - e.g., 128.138.207.160/27=128.138.207.160/255.255.255.224
  - 127.0.0.1/8= ?
  - 192.168.0.3/24= ?
  - 135.176.181.11/16= ?

IPv4 Addressing

- An IP address is associated with an interface, not a host
  - a host with more than one interface may have more than one IP address $\Rightarrow$ we then talk about IP alias
- The assignment of addresses over an Internet topology is crucial to limit the complexity of routing and forwarding
  - This assignment is managed by non-profit companies such as AFNIC for France
- The key idea is to assign addresses with the same prefix to interfaces that are on the same subnet

Classless Interdomain Routing

- All interfaces in the same subnet share the same **address prefix**
  - e.g., in the previous example we have 123.1.1.—, 123.1.2.—, 101.0.1.—, and 111.3.3.—
- Network addresses prefix-length notation : address/prefix-length
  - e.g., 123.1.1.0/24, 123.1.1.0/24, 101.0.1.0/24, and 111.3.3.0/24
  - 123.1.1.0/24 means that all the addresses share the same leftmost 24 bits with address 123.1.1.0
- This addressing scheme is not limited to entire bytes. For example, a network address might be 128.138.207.160/27

Examples

- Network address 128.138.207.160/27

$$\overbrace{10000000 \quad 10001010 \quad 11001111 \quad 101}^{\text{subnet}}\,00000_{\text{two}}$$

Question : does these addresses belong to the same subnet ?

128.138.207.185 ?

$$10000000 \quad 10001010 \quad 11001111 \quad 10111001_{\text{two}}$$

128.138.207.98 ?

$$10000000 \quad 10001010 \quad 11001111 \quad 01100010_{\text{two}}$$

128.138.207.194 ?

$$10000000 \quad 10001010 \quad 11001111 \quad 11000010_{\text{two}}$$

Ranges

- What is the range of addresses in 128.138.207.160/27 ?

Ranges

- What is the range of addresses in 128.138.207.160/27 ?

$$\overbrace{10000000\quad 10001010\quad 11001111\quad 101}^{\text{subnet}}\,00000_{\text{two}}$$

Ranges

- What is the range of addresses in 128.138.207.160/27 ?

$$10000000\quad 10001010\quad 11001111\quad \underbrace{101}_{\#\text{ subnets } 2^3}\quad 00000$$

Ranges

- What is the range of addresses in 128.138.207.160/27 ?

$$10000000\quad 10001010\quad 11001111\quad 101\quad \underbrace{00000}_{\#\text{ hosts } 2^5 - 2}$$

Ranges

- What is the range of addresses in 128.138.207.160/27 ?

$$\overbrace{10000000\quad 10001010\quad 11001111\quad 101}^{\text{subnet}}\,00000_{\text{two}}$$

$10000000\quad 10001010\quad 11001111\quad 10100000_{\text{two}}$
$10000000\quad 10001010\quad 11001111\quad 10100001_{\text{two}}$
$10000000\quad 10001010\quad 11001111\quad 10100010_{\text{two}}$
$10000000\quad 10001010\quad 11001111\quad 10100011_{\text{two}}$
⋮
$10000000\quad 10001010\quad 11001111\quad 10111111_{\text{two}}$

128.138.207.160–128.138.207.191

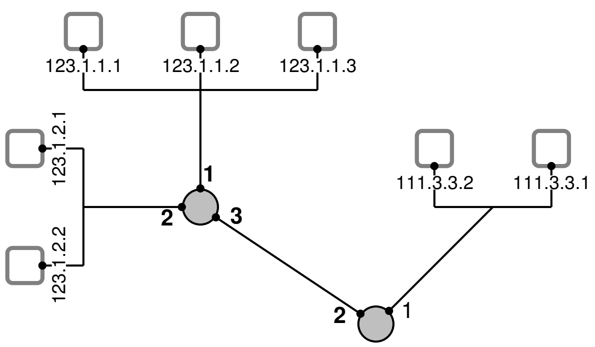
Classless Interdomain Routing

- This **any-length prefix** scheme is also called classless interdomain routing (CIDR)
  - as opposed to the original scheme which divided the address space in “classes”

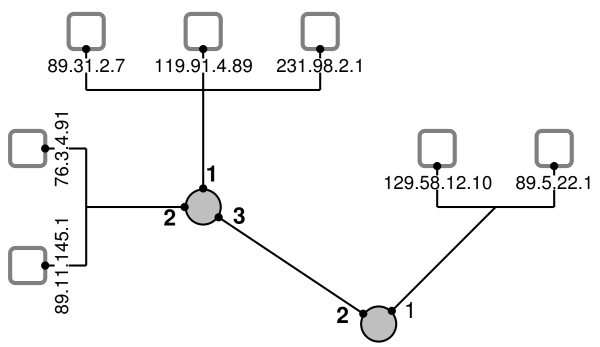
address class	prefix length
A	8
B	16
C	24

- Why is the idea of the common prefix so important ?
- Routers outside a (sub)network can ignore the specifics of each address within the network
  - there might be some 64 thousands hosts in 128.138.0.0/16, but they all appear as one address from the outside

Example : Good Address Allocation



Example : Bad Address Allocation



Longest-Prefix Matching

- In choosing where to forward a datagram, a router chooses the entry that matches the destination address with the longest prefix E.g.,
  - 123.4.1.69→ ?
  - 68.142.226.44→ ?
  - 98.7.2.71→ ?
  - 200.100.2.1→ ?
  - 128.138.207.167→ ?
  - 123.4.20.11→ ?
  - 123.4.21.10→ ?

forwarding table	
network	**port**
123.4.0.0/16	1
98.7.1.0/16	2
123.4.20.0/24	2
128.0.0.0/1	3
66.249.0.0/16	3
0.0.0.0/1	4
128.138.0.0/16	4

Special Addresses

IPv4 defines a number of special addresses or address blocks

- “Private,” non-routable address blocks
  10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16
- Default route
  0.0.0.0/0
- Loopback (a.k.a., localhost)
  127.0.0.0/8
- IP Multicast
  224.0.0.0/4
- Broadcast
  255.255.255.255/32

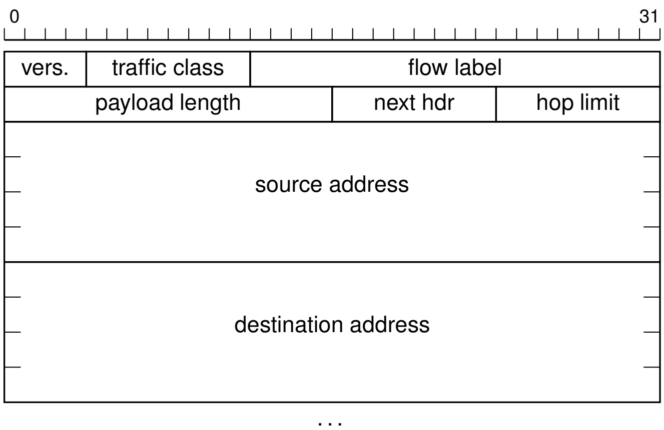
and there are many more... (RFC5735)

```
Address Block       Present Use                Reference
------------------------------------------------------------------
0.0.0.0/8           "This" Network             RFC 1122, Section 3.2.1.3
10.0.0.0/8          Private-Use Networks       RFC 1918
127.0.0.0/8         Loopback                   RFC 1122, Section 3.2.1.3
169.254.0.0/16      Link Local                 RFC 3927
172.16.0.0/12       Private-Use Networks       RFC 1918
192.0.0.0/24        IETF Protocol Assignments  RFC 5736
192.0.2.0/24        TEST-NET-1                 RFC 5737
192.88.99.0/24      6to4 Relay Anycast         RFC 3068
192.168.0.0/16      Private-Use Networks       RFC 1918
198.18.0.0/15       Network Interconnect
                    Device Benchmark Testing   RFC 2544
198.51.100.0/24     TEST-NET-2                 RFC 5737
203.0.113.0/24      TEST-NET-3                 RFC 5737
224.0.0.0/4         Multicast                  RFC 3171
240.0.0.0/4         Reserved for Future Use    RFC 1112, Section 4
255.255.255.255/32  Limited Broadcast          RFC 919, Section 7
                                               RFC 922, Section 7
```


IPv6

- “New-generation IP”
- Why ?
  - the IPv4 address space is too small
- Given the obvious difficulty of replacing IPv4, the short-term benefits of IPv6 are debatable
- Nobody questions the long-term vision
- Also, IPv6 improves various design aspects of IPv4

IPv6 Datagram Format



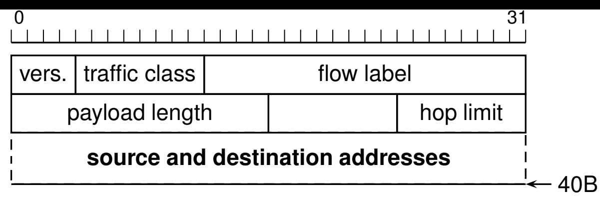
IPv6 Main Design Features

- Expanded addressing
  - 128-bit addresses
  - **anycast** address
- Header format simplification
  - efficiency : reducing the processing cost for the common case
  - bandwidth : reducing overhead due to header bytes
- Improved support for extensions and options
- Flow labeling
  - special handling and non-default quality of service
  - e.g., video, voice, real-time traffic, etc.

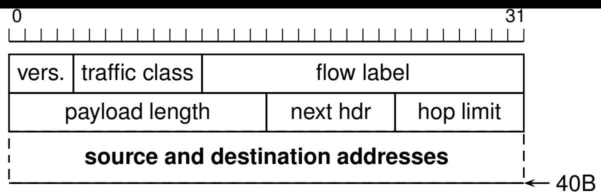
What is Missing from IPv4 ?

- Fragmentation
  - IPv6 pushes fragmentation onto the end-systems
  - efficiency
- Header checksum
  - efficiency
    - ★ how does the checksum in IPv4 behave with respect to the time-to-live field ?
    - ★ the checksum must be recomputed at every hop, so IPv6 avoids that by getting rid of the checksum altogether
  - avoid redundancy : both link-layer protocols and transport protocols already provide error-detection features
- Options
  - efficiency : a fixed-length header is easier to process
  - better modularity for extensions and options

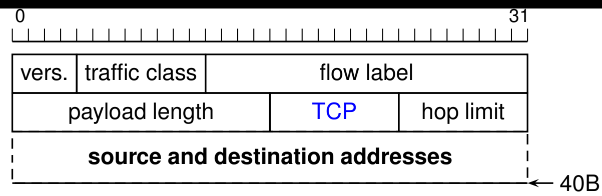
Higher-Level Protocol and Extensions



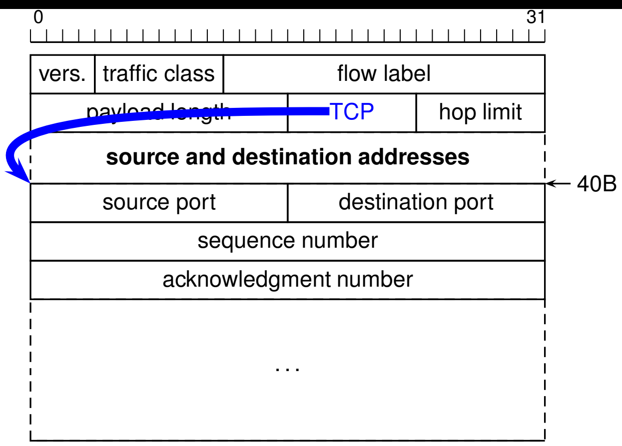
Higher-Level Protocol and Extensions



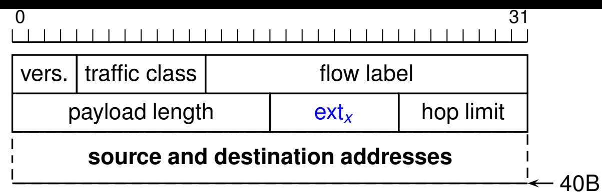
Higher-Level Protocol and Extensions



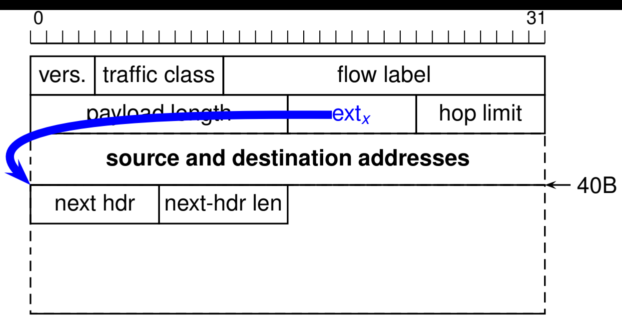
Higher-Level Protocol and Extensions



Higher-Level Protocol and Extensions



Higher-Level Protocol and Extensions



Higher-Level Protocol and Extensions