IPv4 and IPv6 Addressing

Campus Network Design & Operations Workshop

UNIVERSITY OF OREGON

Last updated 23rd October 2022

Network Startup Resource Center

IP Addresses

- Internet connected networks use two types of IP Addressing
  - IPv4 – legacy Internet protocol
  - IPv6 – new Internet protocol
- Presentation describes IPv4 addresses and IPv6 addresses & addressing
- The Campus Network Design Workshop labs use both IPv4 and IPv6 for all exercises
  - Dual stack network (both protocols running in parallel)

IPv4 Addresses

- 32-bit binary number
  - How many unique addresses in total?

IPv4 Addresses

- 32-bit binary number
  - How many unique addresses in total?
  - 2^{32} which is 4,294,967,296 addresses
- Conventionally represented as four dotted decimal octets
- If you turn on all bits this is:

11111111111111111111111111111111

255 . 255 . 255 . 255

Can you explain why 11111111 = 255 in decimal?

IPv4 Addresses

- Remember binary mathematics!
- Each bit is basically to the power of 2. First bit from right is 2^0, second bit is 2^1 and so on to the eighth bit which is 2^7.

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$

11111111

- This means that :
- $11111111 = 1x2^7 + 1x2^6 + 1x2^5 + 1x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 1x2^0$
- 11111111 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255

IPv4 Addresses

- 32-bit binary number
- Conventionally represented as four dotted decimal octets

10000000110111111001110100010011

128 . 223 . 157 . 19

Can you explain why 00010011 = 19 in decimal?

IPv4 Addresses

- 32-bit binary number
- Conventionally represented as four dotted decimal octets

10000000110111111001110100010011

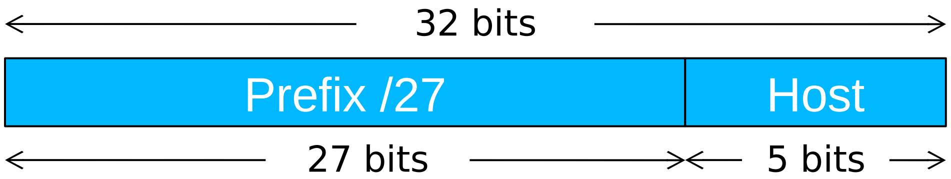
128 . 223 . 157 . 19

$2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$

00010011

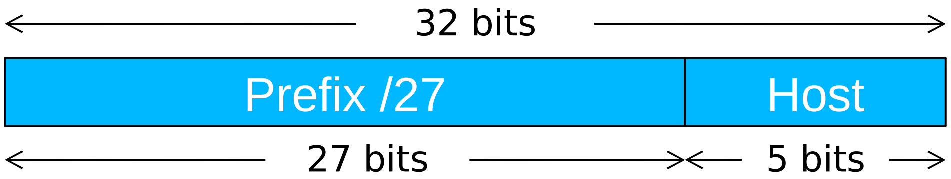
- $00010011 = 0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
- $00010011 = 0 + 0 + 0 + 16 + 0 + 0 + 2 + 1 = 19$

Prefixes



- A range of IP addresses is given as a prefix, e.g. 192.0.2.128/27
- In this example:
  - How many addresses are available?
  - What are the lowest and highest addresses?

Prefixes



- A range of IP addresses is given as a prefix, e.g. 192.0.2.128/27
- In this example:
  - How many addresses are available?
    - Number of bits for the host = 32 – 27 = 5 bits
    - Number of available addresses = 2^5 = 32

Prefix Calculation

192 . 0 . 2 . 128

11000000000000000000001010000000

Prefix length /27 → First 27 bits are fixed

Prefix Calculation

192 . 0 . 2 . 128

11000000000000000000001010000000

Prefix length /27 → First 27 bits are fixed

Lowest address:

11000000000000000000001010000000

192 . 0 . 2 . 128

Prefix Calculation

192 . 0 . 2 . 128

1100000000000000**00000010**100|00000

Prefix length /27 → First 27 bits are fixed

Lowest address:

11000000000000000000001010000000

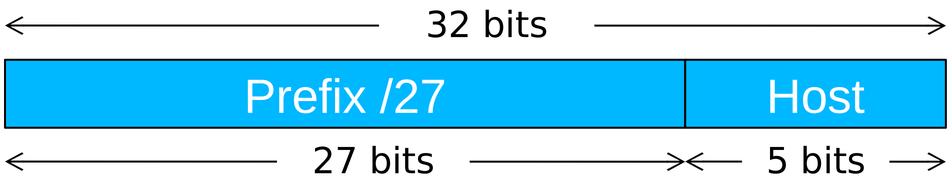
192 . 0 . 2 . 128

Highest address:

11000000000000000000001010011111

192 . 0 . 2 . 159

IPv4 “Golden Rules”



1. All hosts on the same L2 network must share the same prefix
2. All hosts with the same prefix have different host part
3. Host part of all-zeros and all-ones are reserved

Golden Rules for 192.0.2.128/27

- Lowest 192.0.2.128 = network address
- Highest 192.0.2.159 = broadcast address
- Usable: 192.0.2.129 to 192.0.2.158
- Number of usable addresses: 32 - 2 = 30

Exercises

- Network 10.10.10.0/25
  - How many addresses in total?
  - How many usable addresses?
  - What are the lowest and highest usable addresses?

Exercises

- Network 10.10.10.0/25
  - How many addresses in total?
  - How many usable addresses?
  - What are the lowest and highest usable addresses?

Hint...

10 . 10 . 10 . 0

```
000010100000101000001010 0|0000000
000010100000101000001010 0|1111111
```

Prefix length /25 → First 25 bits are fixed

An Edge Case

- How many usable addresses in a /30 prefix?
- What is this used for?
  - (Note: modern routers support /31 for this purpose to reduce IPv4 address wastage)

An Edge Case

- How many usable addresses in a /30 prefix?
  - Number of host bits is 32 – 30 = 2
  - Number of addresses is $2^2 = 4$
  - Number of usable address is 4 – 2 = 2
- What is this used for?
  - Used for Point-to-Point links

Netmask

- Netmask is just an alternative (old) way of writing the prefix length
- A ‘1’ for a prefix bit and ‘0’ for a host bit
- Hence N x 1’s followed by (32-N) x 0’s

/27 =

11111111111111111111111111100000

255 . 255 . 255 . 224

How did we get to 224?

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

Netmask

/27 =

11111111111111111111111111100000

255 . 255 . 255 . 224

How did we get 224?

128 + 64 + 32 = 224

Or: 5 bits = 32 IPs (2^5)

256 – 32 = 224

What about a "/26" ?

What about a "/28" ?

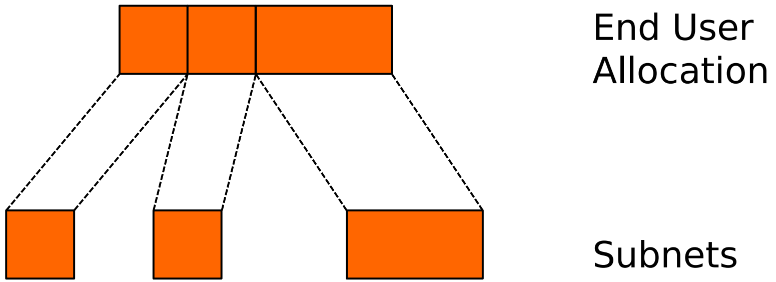
https://nsrc.org/workshops/2009/summer/ref/netmask-table.html

UNIVERSITY OF OREGON

NSRC Network Startup Resource Center

Subnetting

- Since each L2 network needs its own prefix, then if you route more than one network you need to divide your allocation
- Ensure each prefix has enough IPs for the number of hosts on that network



Subnetting Example

- You have been given 192.0.2.128/27
- However, you want to build two Layer 2 networks and route between them
- The Golden Rules demand a different prefix for each network
- Let’s split this address space into two equal-sized pieces

Subnetting /27

192 . 0 . 2 . 128

110000000000000000000010100|00000

Move one bit from host part to prefix

Subnetting /27

192 . 0 . 2 . 128

11000000000000000000001010000000

Move one bit from host part to prefix

We now have two /28 prefixes

11000000000000000000001010000000

192 . 0 . 2 . 128

Second prefix:

11000000000000000000001010010000

192 . 0 . 2 . 144

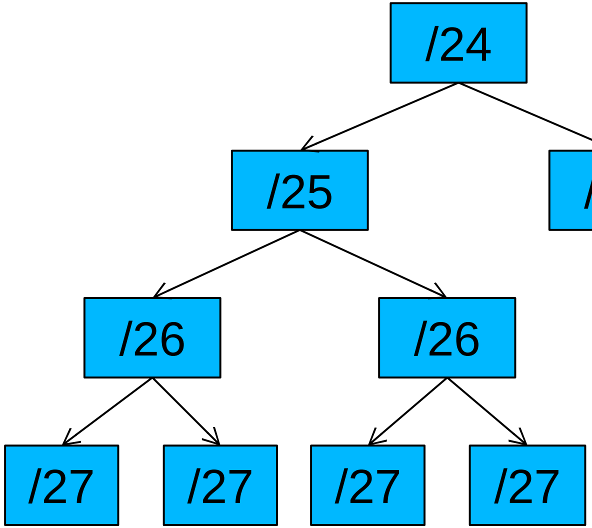
How did we get “144” for the second prefix?

Check correctness

- Expand each new prefix into lowest and highest
- Ranges should not overlap
  - 192.0.2.128/28
    - Lowest (network) = 192.0.2.128
    - Highest (broadcast) = 192.0.2.143
  - 192.0.2.144/28
    - Lowest (network) = 192.0.2.144
    - Highest (broadcast) = 192.0.2.159
  - How many usable addresses now?

Aggregation tree

- Continue to divide prefixes as required
- Can visualize this as a tree



Questions about IPv4?

IPv6 addresses

- 128-bit binary number
  - How many unique addresses in total?

IPv6 addresses

- 128-bit binary number
  - How many unique addresses in total?
  - 2^{128}
  - $3.402823669209 \times 10^{38}$
  - 340,282,366,920,938,463,463,374,607,431,768,211,456

IPv6 addresses

- 128-bit binary number
  - How many unique addresses in total?
  - 2^{128}
  - $3.402823669209 \times 10^{38}$
  - 340,282,366,920,938,463,463,374,607,431,768,211,456
- Conventionally represented in hexadecimal – 8 words of 16 bits, separated by colons

2607:8400:2880:0004:0000:0000:80DF:9D13

Hexadecimal

4 bits
= 1 hex digit

0000	**0**	1000	**8**
0001	**1**	1001	**9**
0010	**2**	1010	**A**
0011	**3**	1011	**B**
0100	**4**	1100	**C**
0101	**5**	1101	**D**
0110	**6**	1110	**E**
0111	**7**	1111	**F**

4 hex digits
= 16 bits

0000 = 0000 0000 0000 0000
FFFF = 1111 1111 1111 1111

IPv6 addresses

- Our example address:

  2607:8400:2880:0004:0000:0000:80DF:9D13

- Leading zeros can be dropped

IPv6 addresses

- Our example address:

2607:8400:2880:0004:0000:0000:80DF:9D13

- Leading zeros can be dropped
- The largest contiguous run of all-zero words can be replaced by "::" (see RFC5952)

2607:8400:2880:~~000~~4:~~0000:0000~~:80DF:9D13

IPv6 addresses

- Our example address:

2607:8400:2880:0004:0000:0000:80DF:9D13

- Leading zeros can be dropped
- The largest contiguous run of all-zero words can be replaced by "::" (see RFC5952)

2607:8400:2880:~~000~~4:~~0000:0000~~:80DF:9D13

2607:8400:2880:4::80DF:9D13

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

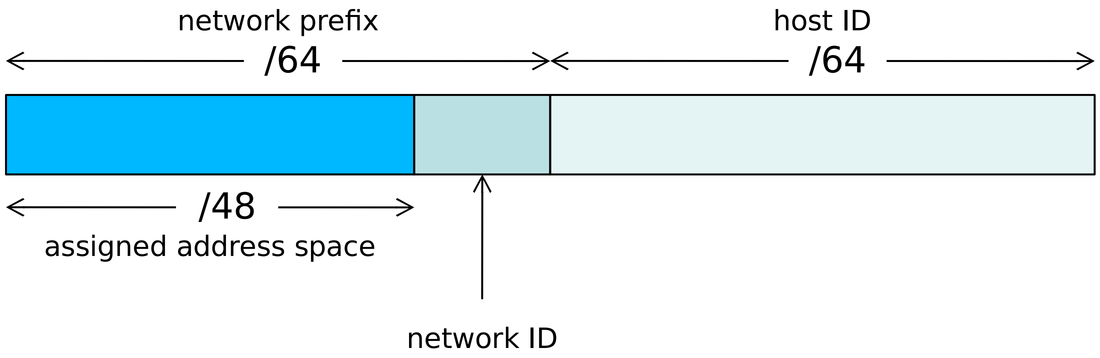
IPv6 rules

- With IPv6, every subnet is /64 (*1)
- The remaining 64 bits can be assigned by hand, or picked automatically
  - all-zeros address is reserved (*1) - *Subnet-Router Anycast address*
- There are special prefixes
  - e.g. link-local addresses start with FE80::
- Total available IPv6 space is ≈ 2^{61} subnets
  - Global unicast addresses have first 3 bits set to 001

(*1) Except /127 recommended for point-to-point links (RFC 6164), in which case the all-zeros address is allowed

IPv6 addressing

- Typical end-user allocation is /48 or /56 *(dependent on ISP policy)*



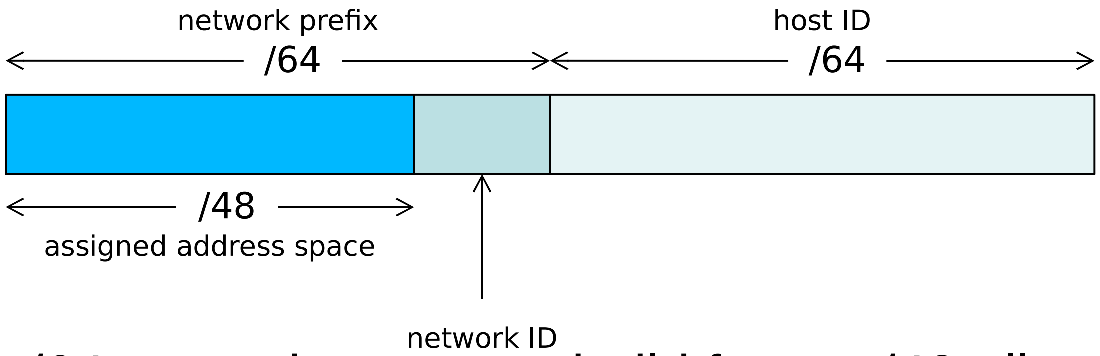
- How many /64 networks can you build from a /48 allocation?

UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

IPv6 addressing

- Typical end-user allocation is /48



- How many /64 networks can you build from a /48 allocation?
  - IPv6 address is 128 bits which means you have 128 – 64 – 48 = 16 bits
  - Number of networks = 2^{16} = 65,536

IPv6 addressing

- You are assigned 2001:DB8:123::/48
  - 2001:0DB8:0123:0000:0000:0000:0000:0000
- Lowest /64 network?

IPv6 addressing

- You are assigned 2001:DB8:123::/48
  - 2001:0DB8:0123:0000:0000:0000:0000:0000
- Lowest /64 network?
  - 2001:DB8:123:0000::/64
  - written simply 2001:DB8:123::/64

IPv6 addressing

- You are assigned 2001:DB8:123::/48
  - 2001:0DB8:0123:0000:0000:0000:0000:0000
- Lowest /64 network?
  - 2001:DB8:123:0000::/64
  - written simply 2001:DB8:123::/64
- Highest /64 network?

IPv6 addressing

- You are assigned 2001:DB8:123::/48
  - 2001:0DB8:0123:0000:0000:0000:0000:0000
- Lowest /64 network?
  - 2001:DB8:123:0000::/64
  - written simply 2001:DB8:123::/64
- Highest /64 network?
  - 2001:DB8:123:FFFF::/64

Ways to allocate the host part

- We recommend manual configuration for servers
  - Gives a persistent and predictable address
- Choose any scheme that you like, e.g.
  - Can number sequentially from ::1
  - Can use the last octet of the IPv4 address
  - Can embed the whole IPv4 address in the lower 32 bits
    - e.g. 2607:8400:2880:4::80DF:9D13
    - 80DF9D13 hex = 128.223.157.19 in decimal
    - Can also be written as 2607:8400:2880:4::128.223.157.19

Ways to allocate the host part

- Automatic: “stateless address autoconfiguration” (SLAAC)
  - Prefix is learned from Router Advertisement messages, and client derives the low 64 bits from the network card MAC address
  - Design problem: MAC address is persistent and means you can be tracked around the Internet
  - Now most clients generate random, changing "privacy" addresses

Ways to allocate the host part

- Automatic: “stateless address autoconfiguration” (SLAAC)
  - Prefix is learned from Router Advertisement messages, and client derives the low 64 bits from the network card MAC address
  - Design problem: MAC address is persistent and means you can be tracked around the Internet
  - Now most clients generate random, changing "privacy" addresses
- Or use DHCPv6
  - Gives you DHCP logs and non-changing addresses
  - Consistent with how you manage IPv4 address allocation
  - Unfortunately, Android does not support DHCPv6

Additional client configuration

- Default gateway is learned via Router Advertisements
  - even if you are using DHCPv6
- Router advertisements can provide recursive DNS servers (RDNSS) and DNS search list (DNSSL) settings
  - But clients may need additional configuration that DHCP traditionally does
- This means that there are three variations you may come across
  - SLAAC only
  - SLAAC + stateless DHCPv6 for extra client configuration
  - Stateful DHCPv6 + Router Advertisements for default gateway

Link local addresses

- All clients also assign themselves a link-local address
- Starts with FE80:
  - May see explicitly scoped to an interface, e.g. FE80::1%eth0
- Remainder of address auto-generated from MAC address (like SLAAC), or can be manually overridden
- Can only be used for communication on the same LAN segment
  - But link-local addresses can be used for next hop, e.g. default gateway

Other notes on IPv6

- Mostly similar to IPv4, e.g. forwarding, prefix matching
- Loopback address is ::1
  - Equivalent to IPv4 127.0.0.1
- “ARP” is replaced by “NDP”
- Beware of "Duplicate Address Detection" (DAD)
  - Many vendors implement this in such a way that any address conflict causes one side to *permanently* disable the address until the interface is shutdown and re-enabled
  - Cisco shows the disabled address as "[DUP]"

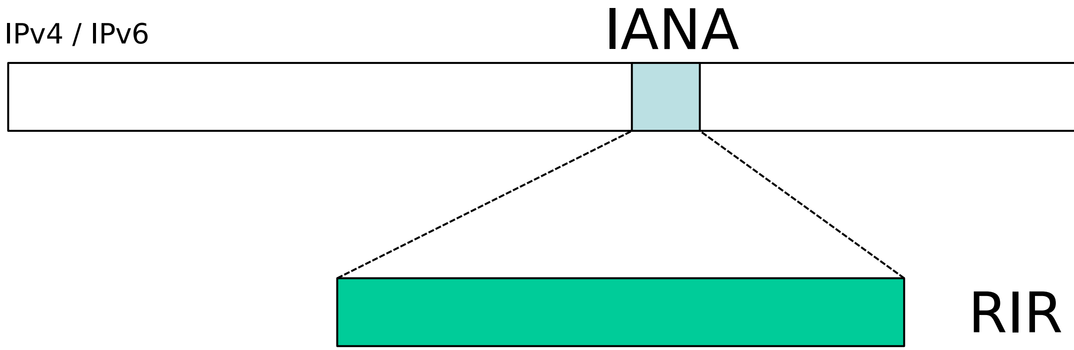
Questions about IPv6?

Hierarchical address allocation

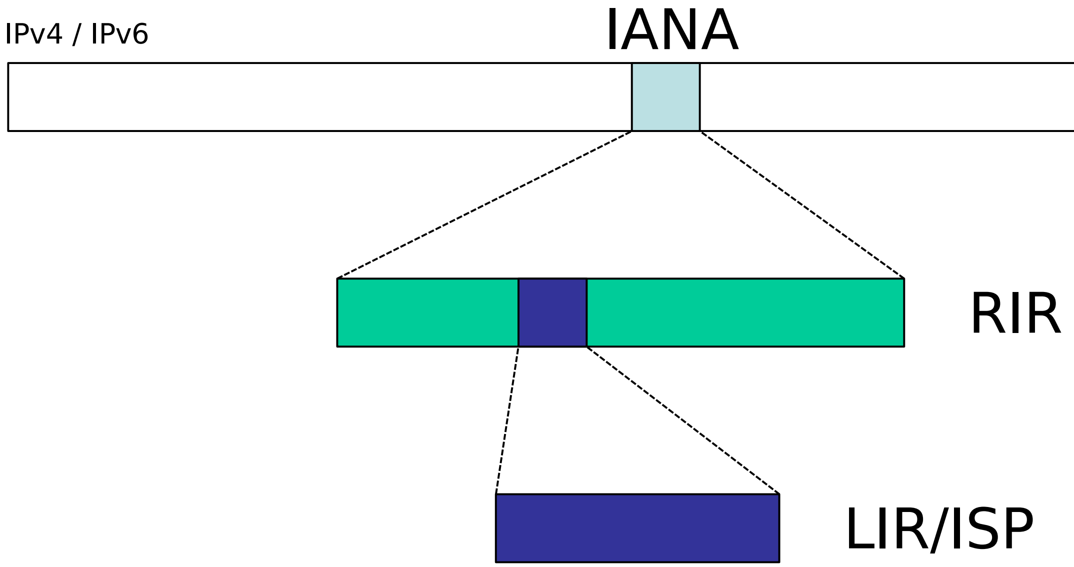
IPv4 / IPv6

IANA

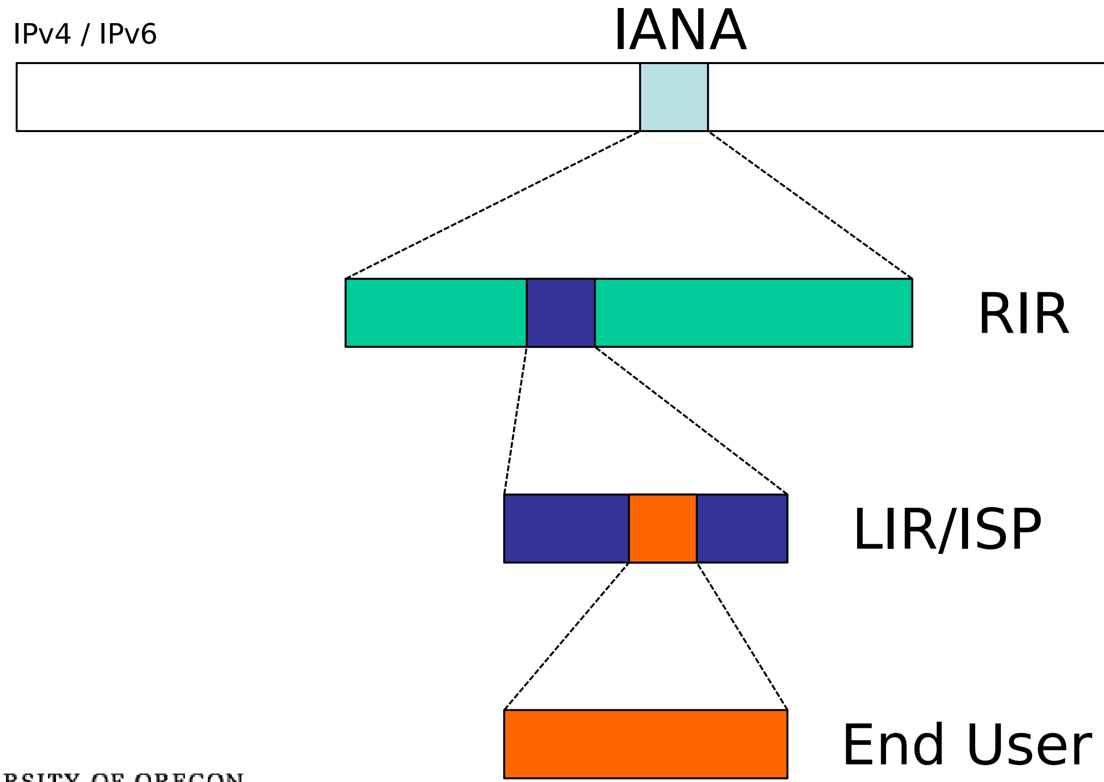
Hierarchical address allocation



Hierarchical address allocation



Hierarchical address allocation



UNIVERSITY OF OREGON

NSRC
Network Startup Resource Center

IPv4 Address Distribution

- IPv4 addresses
  - Distributed by RIRs according to demonstrated need
  - Almost completely exhausted
  - RIRs have IPv4 run-out policies
    - E.g. one-off assignment from a limited pool, or waiting list
- Typical Campus:
  - Small public address block
    - For public servers, NAT pools
    - Anything from /28 to /21 depending on RIR region/upstream
  - Private address block
    - For internal end users, network management, etc

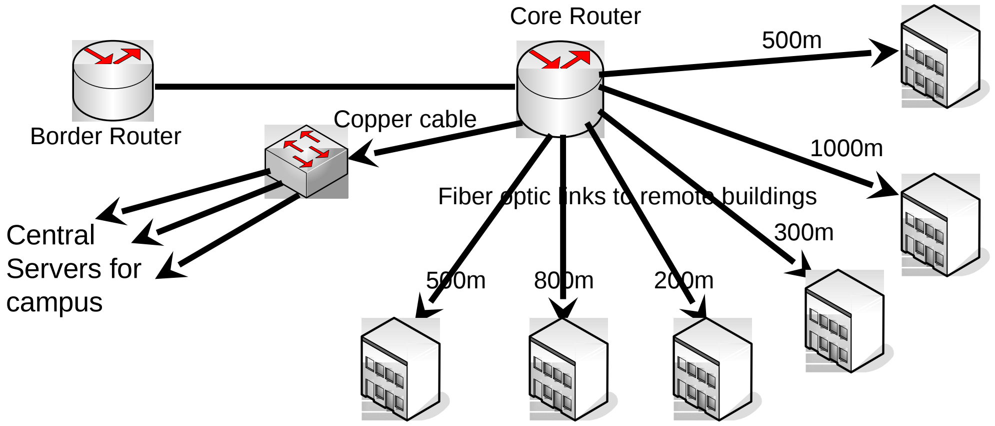
IPv6 Address Distribution

- IPv6 addresses
  - Network operators receive minimum of /32
    - Includes RENs, University Campuses, etc
  - End-sites receive /48 or /56
  - Smallest subnet size is /64
- Typical Single Campus:
  - /48 divided out amongst buildings
- Typical Multi-Campus or Multi-Faculty:
  - /32 divided out amongst Campuses
    - /48 per campus

Questions about IP Address Distribution?

Designing an Address Plan

- Now we will look at how to design an address plan for a simple campus



Designing an Address Plan

- There are different numbers of hosts in each part of the campus

Network	Number of Devices
Border Router to Core Router	2
Server Network	23
Science Building	120
Arts Building	52
Engineering Building	200
Library	80
Administration Building	40
Languages Building	30
Staff & Student Hostel	60

IPv4 Plan

- **You will not get public IPv4 space for all this!!**
  - In the old days you could (and would have to plan and subnet carefully to justify your allocation)
- Today, make the best of whatever you can get
  - Got a /28? Use a /29 for a few servers with public IPs and the other /29 for a NAT pool
  - Got a /24? You're lucky! Maybe two /26 server networks and /25 for NAT?
- Everything else has to be private, but that's easy with 10.0.0.0/8
- Make a logical IPv4 address plan that's easy to manage

Suggested campus IPv4 plan

- Allocate 10.***X***.0.0/16 to each building and subdivide consistently

Network	Container subnet
Core	10.0.0.0/16
Building 1	10.1.0.0/16
Building 2 … etc …	10.2.0.0/16 ... etc …
Building 255	10.255.0.0/16

Use	Subnet	VLAN
Network Management	10.x.0.0/24	x0
Wired (staff/general)	10.x.1.0/24	x1
Wired (computer lab)	10.x.2.0/24	x2
Wireless *	10.x.8.0/21	x8

Use	Subnet	VLAN
Loopbacks (further divided into /32's)	10.0.0.0/24	-
Point-to-points (divided into /30's or /31's)	10.0.1.0/24	-
Public servers	192.0.2.72/29	2
Student servers	10.0.3.0/24	3
Finance servers	10.0.4.0/24	4

** Larger subnet for whole-building wireless and to allow for DHCP churn*

Advantages of a consistent plan

- You can look at any IP address and immediately tell:
  - Which building it's in
  - Which subnet / application within that building
  - Which VLAN it's on
- Very helpful for troubleshooting and tracing
- No need to think when creating new configs or allocations
  - Lends itself well to templated configs
- Feel free to tailor our suggestions to your use case

Modified plan for ACLs

- Protect switch management IPs, video cameras, BMS etc from general access by aggregating them, for example under 10.128/9
  - Network management = 10.128.0.0/16
    - Building 1 network management = 10.128.1.0/24
    - Building 2 network management = 10.128.2.0/24 … etc
  - IP cameras = 10.129.0.0/16 … etc
- This allows for short, simple ACLs in the core router
  - allow from [monitoring systems] to 10.128.0.0/16
  - allow from [video recorders] to 10.129.0.0/16
  - deny from everywhere else to 10.128.0.0/9

Multi-campus university?

- Allocate blocks out of 10/8 to each campus, e.g. 8 lots of /11
  - 10.0.0.0/11 (10.0 – 10.31)
  - 10.32.0.0/11 (10.32 – 10.63)
  - …
  - 10.224.0.0/11 (10.224 – 10.255)
- More than 32 buildings in a campus? Give it another /11 block, and/or subdivide each building block

Building 33A	10.33.0.0/18	Building 33C	10.33.64.0/18
Building 33B	10.33.128.0.0/18	Building 33D	10.33.192.0/18

IPv6 Plan

- This time we have unique public IPs everywhere!
- Campus gets a /48
  - That's 2^{16} = 65,536 subnets of /64
  - Same as 2^{16} subnets from 10.0.0.0/24 to 10.255.255.0/24
  - A happy alignment
- We will use 2001:db8:abc::/48 as our example

Suggested campus IPv6 plan

- Allocate a /60* to each building and subdivide consistently

Network	Container subnet
Core	2001:db8:abc::/60
Building 1	2001:db8:abc:10::/60
Building 2 … etc …	2001:db8:abc:20::/60
Building 255	2001:db8:abc:2550::/60

Use	Subnet	VLAN
Loopbacks (divided into /128's)	2001:db8:abc::/64	-
Point-to-points (divided into /127's)	2001:db8:abc:1::/64	-
Public servers	2001:db8:abc:2::/64	2
Student servers	2001:db8:abc:3::/64	3
Finance servers	2001:db8:abc:4::/64	4

Use	Subnet	VLAN
Device Management	2001:db8:abc:x0::/64	x0
Wired (staff/general)	2001:db8:abc:x1::/64	x1
Wired (computer lab)	2001:db8:abc:x2::/64	x2
Wireless	2001:db8:abc:x8::/64	x8

* This avoids you having to think in hexadecimal
e.g. building 255 = :2550: instead of :ff00:

VLAN tags

- VLAN IDs 2-4094 are available. You are unlikely to run out!
- If your core router supports "no switchport" and sub-interfaces, then you could use the *same* VLAN tags to every building
  - Worth considering even if you are not short of VLANs
  - Makes edge/dist switch configs nearly identical everywhere

Use	Subnet (v4)	Subnet (v6)	VLAN
Network Management	10.x.0.0/24	2001:db8:abc:xx00::/64	10
Wired (staff/general)	10.x.1.0/24	2001:db8:abc:xx01::/64	11
Wired (computer lab)	10.x.2.0/24	2001:db8:abc:xx02::/64	12
Wireless	10.x.8.0/21	2001:db8:abc:xx08::/64	18

Same on all switches in all buildings

UNIVERSITY OF OREGON

NSRC Network Startup Resource Center

Questions?