Building an IPv6 Number plan

The goal of this exercise

- Not only show a numberplan
- But explain the reasoning behind it
- Use a small example network and address it

Hexadecimal counting

- These go to 15..uhh ‘f’
- Powers of 16
- 0,1,2,3,4,5,6,7,9,a,b,c,d,e,f
- f1 = 17
- 3 + 9 = c
- Need for additional exercise ?

Hexadecimal Numbers

June 2010

Ferenc Csorba

Why hexadecimal for IPv6?

- Hexadecimals used for representing IPv6 addresses
- Information in computers is stored in binary numbers.
- Binary numbers very easily converted to hexadecimal numbers
  - easier than converting to decimal

Number Systems

Overview

- Decimal
- Binary
- Hexadecimal
- Conversion with examples

Decimal (dec, base 10)

- Uses positional representation
- Digits 0,1,2,3,4,5,6,7,8,9 are used
- Each digit corresponds to a power of 10 based on its position in the number
- The powers of 10 increment from 0,1,2 etc as you move from right to left

1000s	100s	10s	1s
3	**1**	**2**	**4**

- **3124**$_{dec}$ = $3x10^3+1x10^2+1x10^1+4x10^0$

Binary (bin, base 2)

- Uses positional representation
- Only digits 0 and 1 used
- Each digit corresponds to a power of 2 based on its position in the number
- The powers of 2 increment from 0,1,2 etc as you move from right to left

8s	4s	2s	1s
1	**1**	**0**	**1**

$$\mathbf{1101}_{bin} = 1x2^3+1x2^{2+}0x2^1+1x2^0 = 13_{dec}$$

Hexadecimal (hex, base 16)

- Uses positional representation
- Digits 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F (0 to15_{dec}) used
- Each digit corresponds to a power of 16 based on its position in the number
- The powers of 16 increment from 0,1,2 etc as you move from right to left

16x16x16 = 4096s	16x16 = 256s	16s	1s
0	**1**	**2**	**b**

012b$_{hex}$ = 0x16^3+1x16^2+2x16^1 +Bx16^0 = 299_{dec}

Correspondences

Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
a	1010	10
b	1011	11
c	1100	12
d	1101	13
e	1110	14
f	1111	15
10	1 0000	16

Conversion: Binary to Decimal

Conversion: binary to decimal

- Multiply and add the positional values:
- **10110101**$_{bin}$ = ? $_{dec}$

128s	64s	32s	16s	8s	4s	2s	1s
1	0	1	1	0	1	0	1

- 10110101=1x(128)+0x(64)=1x(32)+1x(16)+0x(8)+1x(4)+0x(2)+1x(1) = 181$_{dec}$
- **10110101**$_{bin}$ = **181** $_{dec}$

Conversion: Decimal to Binary

Conversion: decimal to binary

- List the increasing powers of 2 from right to left
- Subtract the largest possible power of 2
- Keep subtracting the next largest power from the remainder
- Mark ‘1’ in each column where this is possible, ‘0’ where it is not

Conversion example: decimal to binary

- Binary to decimal:
- **371**$_{dec}$ = ? $_{bin}$
- .

512s	256s	128s	64s	32s	16s	8s	4s	2s	1s

- Largest power of 2 that fits into **371** is 256 (=2^8)
- Remainder is 115 (= 371-256)
- Keep subtracting largest powers from remainders
- Largest power of 2 that fits into 115 is 64 (=2^6)
- ....and so on....

Conversion example dec to bin (continued)

371
- 256

115
- 64

51
- 32

19
- 16

3
- 2

512	256	128	64	32	16	8	4	2	1
0	1	0	1	1	1	0	0	1	1

$371_{dec} = 101110011_{bin}$

Conversion: Hex to Decimal

Conversion: hexadecimal to decimal

- Multiply and add the positional values:
- **f2b**$_{hex}$ = ? $_{dec}$

256s	16s	1s
f	**2**	**b**
15	2	11

- **f2b**$_{hex}$ = 15x(256)+2x(16)+11x(1) = 3883$_{dec}$
- **f2b**$_{hex}$ = **3883** $_{dec}$

Conversion: Decimal to Hex

Conversion: Decimal to Hexadecimal

1. List the increasing powers of 16 from right to left
2. Find the two powers of 16 in the list between which your decimal number fits
3. The smaller of these two powers of 16 will be the largest position/ digit of your hexadecimal number
4. Now find out how many multiples (1-f) of this power fit in your decimal number. That is the value of the largest digit of your hexadecimal number. Write down this value in the position.
5. Subtract this value from your original decimal number
6. Repeat the procedure with the remainder: Find the two powers of 16 in the list between which the remainder decimal number fits etc
7. Keep going until there are no remainders.
8. Fill out all unused positions with 0.

Conversion example: dec to hex

- hex to decimal:
- **4235**$_{dec}$ = ? $_{hex}$

65536s	4096s	256s	16s	1s

- Largest power of 16 that fits into **4235** is 4096 (=16^3)
- So the largest digit of the hexadecimal nr will be the 3rd position from the right, the 4096s
- Value of the largest digit is =1, because 4096 fits into 4235 only once
- Remainder is 139 (= 4235-4096)
- 256 doesn't fit into 139, so that position is = 0
- Largest power of 16 that fits into 139 is 16
- 16 fits into 139 8 times, so the value of that position is 8
- Remainder is 11 so last digit is b

Conversion example dec to hex (continued)

4235 → 134 → 11

65536s	4096s	256s	16s	1s
0	**1**	0	**8**	**b**

4235

- **1**x4096

remainder: 139

139

- 8x16 = -128

remainder: 11

$11_{dec} = B_{hex}$

139:16 = 8,...

8 x16 = 128

$4235_{dec} = 108b_{hex}$

Conversion: Binary to Hex

Conversion: binary to hexadecimal

- Replace each set of four binary digits with the corresponding hexadecimal digit :
- **101101101**$_{bin}$ = ? $_{hex}$

0 0 0 1	0 1 1 0	1 1 0 1
1	6	d

- **101101101**$_{bin}$ = **16d** $_{hex}$

Conversion: Hex to Binary

Conversion example: hex to binary

- Replace each hexadecimal with the corresponding set of four binary digits:
- **103d**$_{hex}$ = ? $_{bin}$

1	0	3	d
0 0 0 1	0 0 0 0	0 0 1 1	1101

- **103d**$_{hex}$ = **0001 0000 0011 1101** $_{bin}$

Note

Quicker (less intuitive) conversion method

- From decimal to binary or hexadecimal
  - Keep dividing by 2 (or 16)
  - Remainders give the digits, starting from the lowest power on the right

Subnetting

- IPv6 subnetting is the same as in IPv4
- But slightly larger (128 bits)
- CIDR rules apply !

Try to stick to 4 bit boundaries !

Subnetting (2)

- 2001:DB8:16ae:f00d:217:f2ff:fe0a:cd48/128
- part of 2001:DB8:16ae:f00d::/64
- /48 ?
  - 2001:DB8:16ae:xxxx
- /56 ?
  - 2001:DB8:16ae:f0xx
- /58 ?
  - 2001:DB8:16ae:f00 to 2001:DB8:16ae:f03, f04 - f07, f08 - f0b, f0c - f0?

Try and assign 4 bits at a time

- 2^4 = 16 = f
- One nibble at a time keeps it simple
- Also matches reverse DNS layout
  - x.x.x.x.d.0.0.f.8.b.d.0.1.0.0.2.ip6.arpa

The rules

- RIPE-481
- Goals:
  - Uniqueness
  - Registration
  - Aggregation
  - Conservation
  - Fairness
  - Minimized overhead

The rules (2)

- Conflict of goals ?

‘In IPv6 address policy, the goal of aggregation is considered to be the most important.’

Why aggregation ?

- Keep the number of routes low, both internal and external
- try and keep it to 1 route per network in the DFZ
- But also keep your internal table small
- Bottom line: routes cost memory
- and therefor cost money

The risk of deaggration

- In IPv4 land
  - assume you have a /16
  - Will deaggregate into 65k routes
- In IPv6 land
  - You get a /32
  - Worst case scenario is 4.2 billion routes !!!
  - The equivalence of the full theoretical v4 space

Assignment rules

- Anybody can get upto a /48
- When in doubt, go big
- /60 is 16 subnets which may seem enough
- But you won’t know what the future brings
- Try and avoid to ever add an additional block
- IPv6 is not a scarce resource (for the moment)

Internal assignments

- RIPE-481, section 5.4.3

  An organisation (i.e. ISP/LIR) may assign a network prefix per PoP as the service infrastructure of an IPv6 service operator. Each assignment to a PoP is regarded as one assignment regardless of the number of users using the PoP. A separate assignment can be obtained for the in-house operations of the operator.

Assignments

- up to /48 per PoP
- 1 additional assignment ‘internal operations’ (/48) (often referred to as ‘core network’)
- Customers up to a /48
- More possible, have to talk to IPRA

Let’s do some work

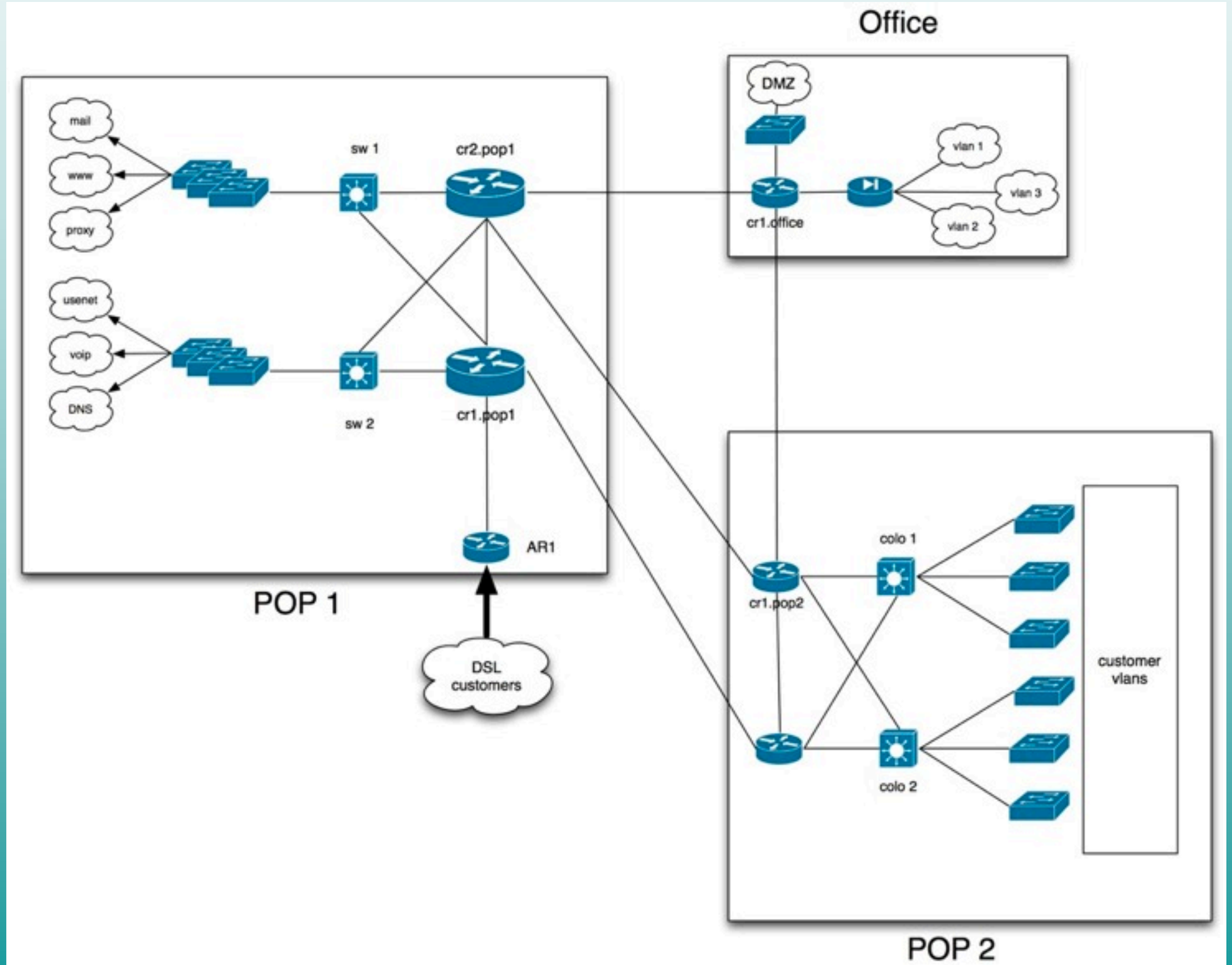
- Keep in mind this is just an example
- It will always be a custom job
- It has to fit your needs

Keep it simple

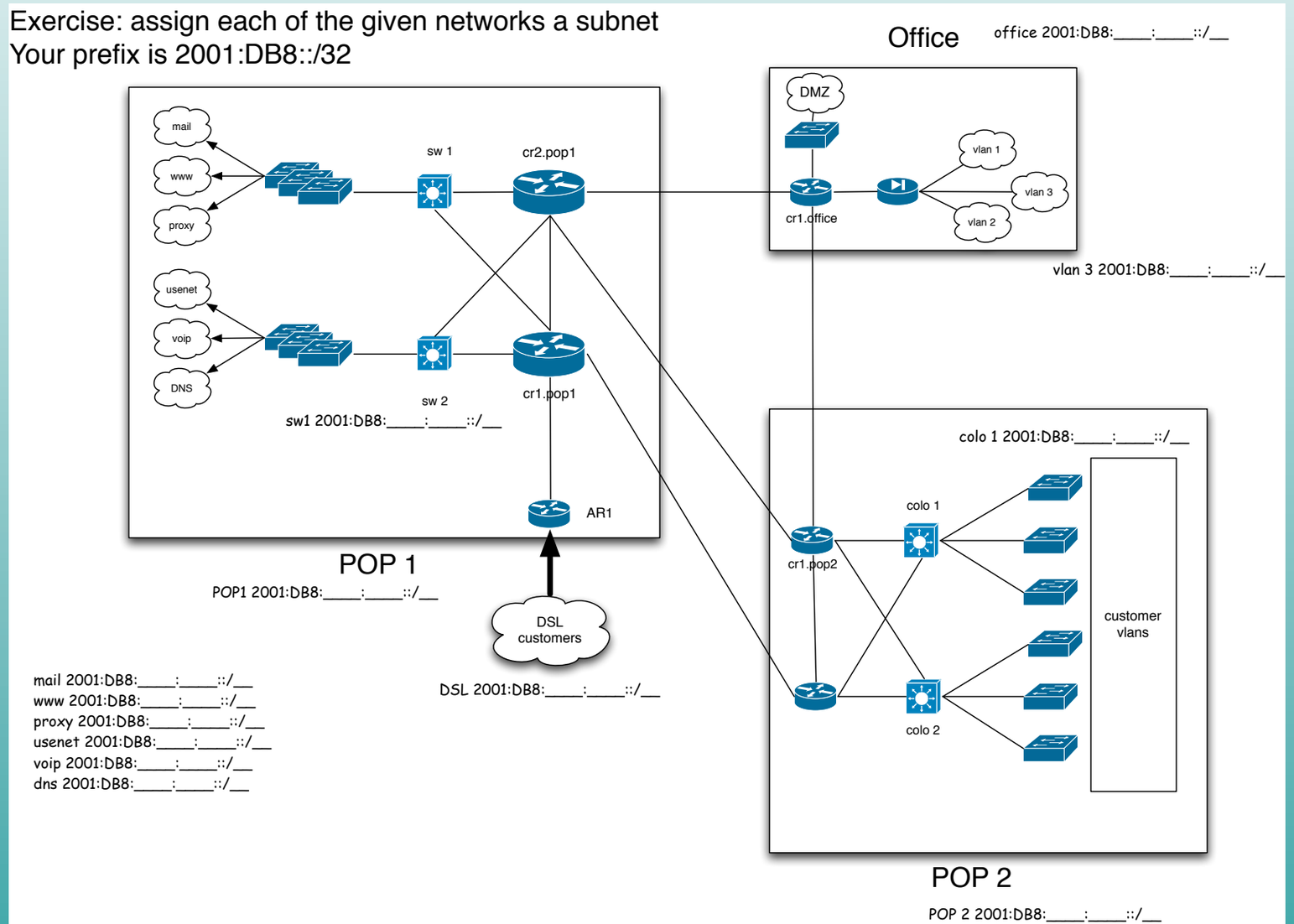
- Use a clear structure
- Try not to leave to many holes
- Things like 2001:xxxx:dead:beef look nice for now, but one day will come back and haunt you
- Imagine how it would look when your DNS is down

Leading zeros can be skipped!

The example network: ISP X



Exercise 1


Exercise: assign each of the given networks a subnet
Your prefix is 2001:DB8::/32
Office
office 2001:DB8:____:____::/__
DMZ
vlan 1
vlan 3
vlan 2
cr1.office
vlan 3 2001:DB8:____:____::/__
mail
www
proxy
usenet
voip
DNS
sw 1
cr2.pop1
sw 2
cr1.pop1
sw1 2001:DB8:____:____::/__
AR1
POP 1
POP1 2001:DB8:____:____::/__
DSL customers
DSL 2001:DB8:____:____::/__
mail 2001:DB8:____:____::/__
www 2001:DB8:____:____::/__
proxy 2001:DB8:____:____::/__
usenet 2001:DB8:____:____::/__
voip 2001:DB8:____:____::/__
dns 2001:DB8:____:____::/__
colo 1 2001:DB8:____:____::/__
colo 1
cr1.pop2
colo 2
customer vlans
POP 2
POP 2 2001:DB8:____:____::/__

Exercise 1

- Most of these are ‘internal’ assignments
- Assign a subnet to the networks/pops listed
- Assume a maximum of 4000 DSL customers

Receiving a prefix

- Of course you won’t have a prefix yet
- For now we assume you will get 2001:DB8::/32
- In your number plan, probably want to note xxxx:yyyy::/32 to keep it clear
- Prevents typos or configuration errors

The first chop

- a /32 contains 2^32 /64 subnets (4.2 billion)
- a /32 contains a minimum of 2^16 assignments
- It might be bigger (almost 17 million /56’s)
- Anybody wants do the math on /127 ?

The first chop (2)

- Let’s divide by 16 (hint again 4 bits)
- 2001:DB8:0000:/36 to 2001:DB8:f000::/36
- Let’s focus on the first one
- And put the others aside
- Assign internals first

Try and minimize the amount of typing

Making internal assigments

- Try and follow the hierarchy of the network
- Always keep in the back of mind DNS might break
- And you end up in the middle of the night ‘guessing’ what the management IP was.

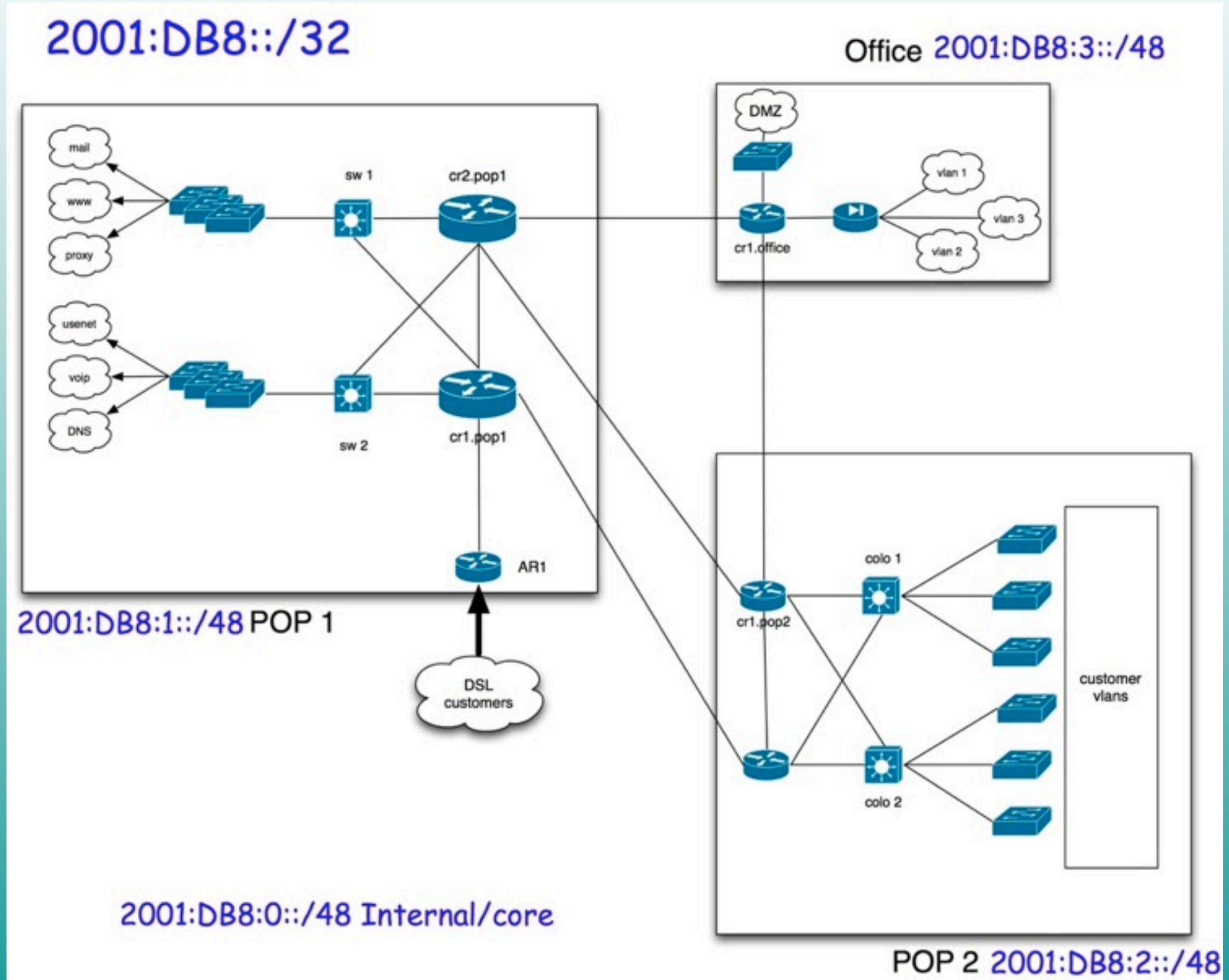
The shorter the easier

- 2001:0DB8:0000::/48 -> 2001:DB8::/48
- 2001:0DB8:0001::/48 -> 2001:DB8:1::/48
- 2001:0DB8:0002::/48 -> 2001:DB8:2::/48
- etc

PoP X

- Let’s use logic
  - PoP 1 -> 2001:DB8:1::/48
  - PoP 2 -> 2001:DB8:2::/48
  - Office has several options
    - 2001:DB8:3::/48
    - 2001:DB8:f::/48
    - 2001:DB8:fff::/48
- For now we’ll take 2001:DB8:3::/48

The initial picture



Assignment within a PoP

- Take a look at the structure
- And try and maintain that in the plan
- Work inside out or vice versa
- Take a look at what you expect from a node
  - A core router might never exceed 256 interfaces
  - A layer 3 access switch might run into thousands

Wash, rinse, repeat

- Again take a 4 bit boundary
  - /48 = 16 * /52 (4096 subnets)
  - /48 = 256 * /56 (256 subnets)
  - /48 = 4096 * /60 (16 subnets)
- So let’s take the first one (think big)

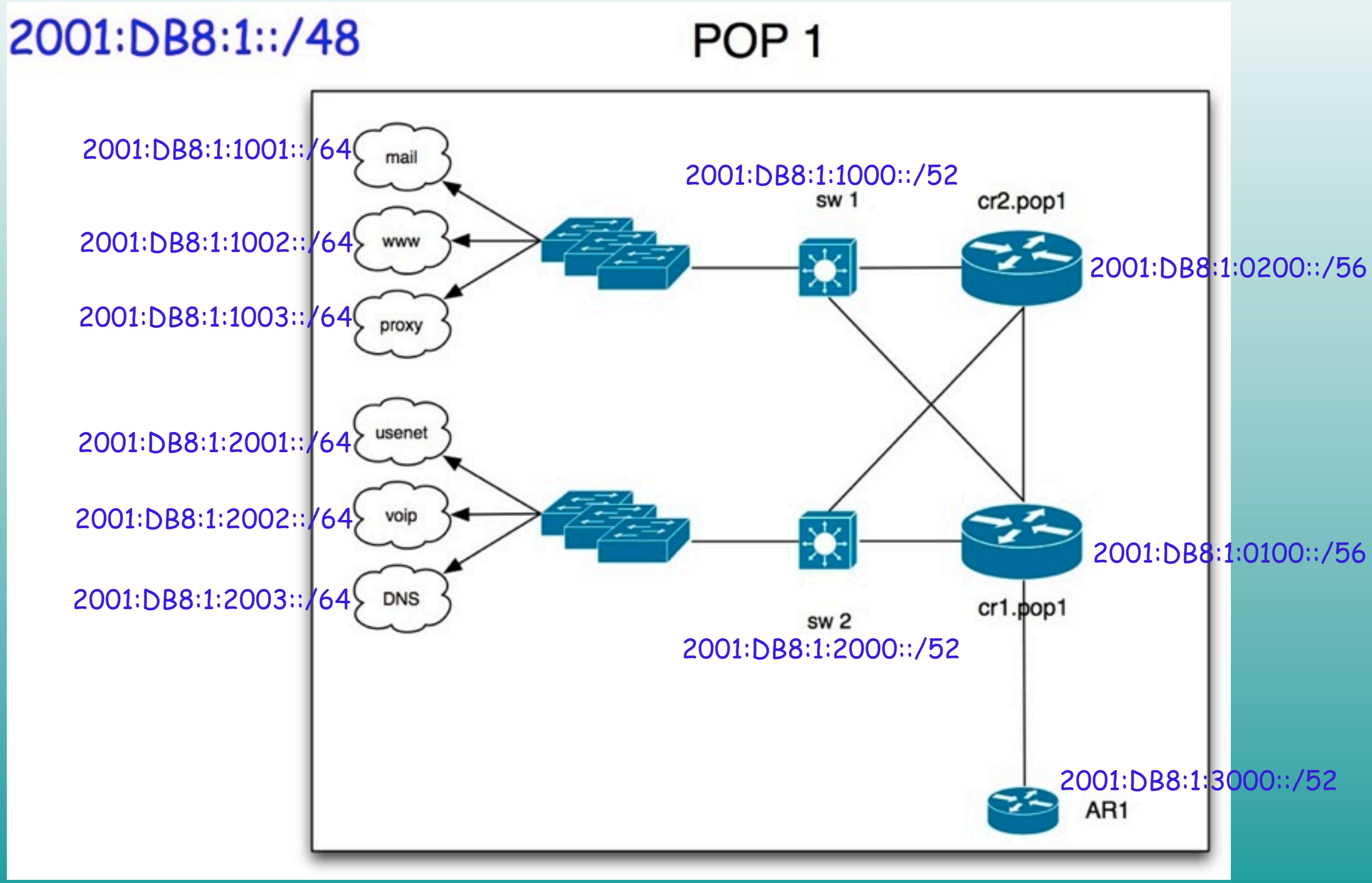
2001:DB8:1::/48

- 2001:DB8:1:0000::/52
- 2001:DB8:1:1000::/52
- 2001:DB8:1:2000::/52
- 2001:DB8:1:3000::/52
- …
- 2001:DB8:1:f000::/52

You can always split it down again

- 2001:DB8:1:0000::/52
  - 2001:DB8:1:0000::/56
  - 2001:DB8:1:0100::/56
  - 2001:DB8:1:0200::/56
  - …
  - 2001:DB8:1:0f00::/56
- 2001:DB8:1:1000::/52
- 2001:DB8:1:2000::/52
- 2001:DB8:1:3000::/52
- …
- 2001:DB8:1:f000::/52

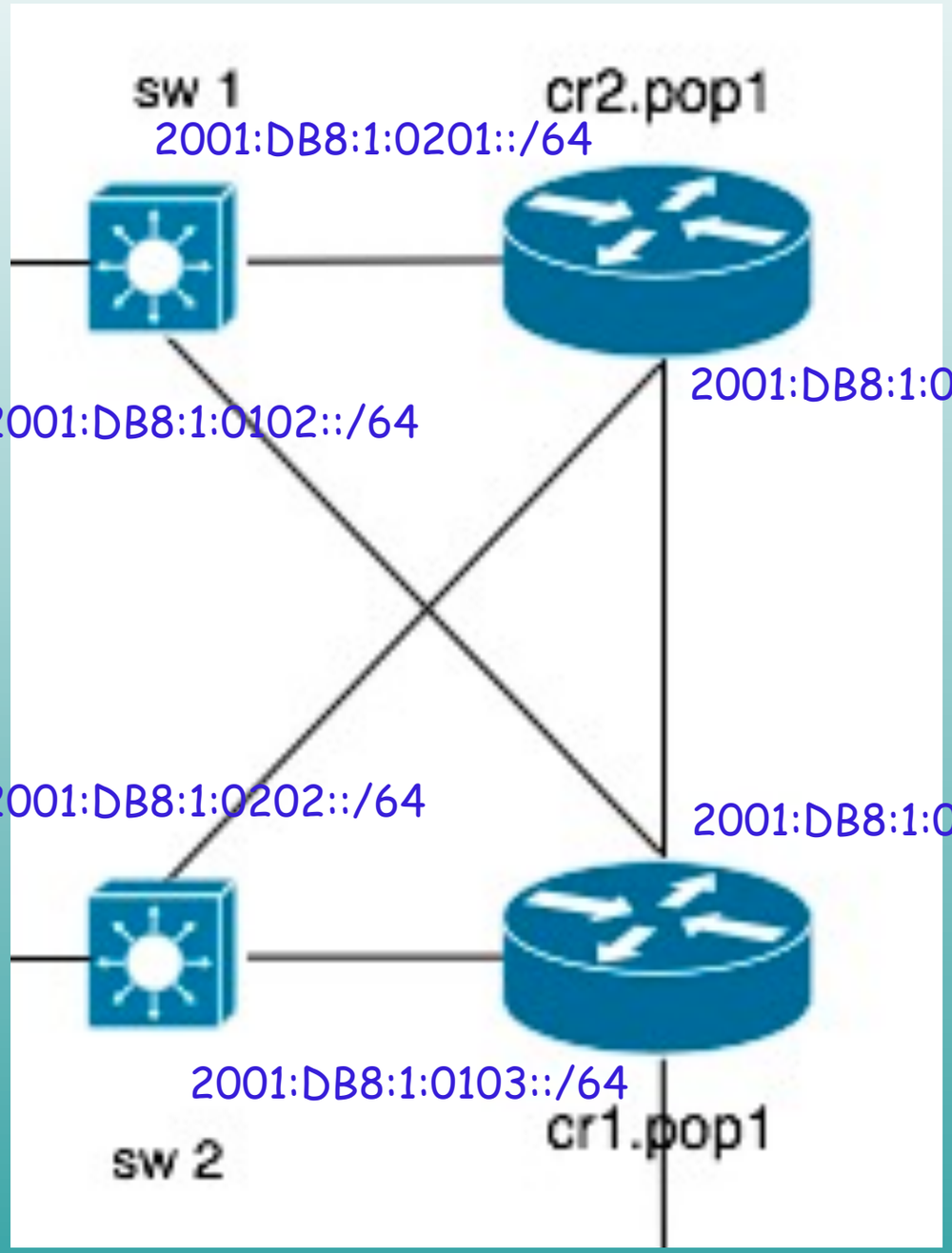
Assign PoP1


2001:DB8:1::/48
POP 1
2001:DB8:1:1001::/64
mail
2001:DB8:1:1002::/64
www
2001:DB8:1:1003::/64
proxy
2001:DB8:1:1000::/52
sw 1
cr2.pop1
2001:DB8:1:0200::/56
2001:DB8:1:2001::/64
usenet
2001:DB8:1:2002::/64
voip
2001:DB8:1:2003::/64
DNS
2001:DB8:1:0100::/56
sw 2
cr1.pop1
2001:DB8:1:2000::/52
2001:DB8:1:3000::/52
AR1

Assigning the links

- Even if you use /127 maintain /64 for assignment
- You really don’t wanna go counting subnets in the middle of the night.
- Use ::1 and ::2 instead of EUI-64
- For reasons explained above
- Follow the natural hierarchy of your network
  - cr1 -> cr 2 comes from ‘cr1 range’
  - cr2 -> sw1 comes from ‘cr2 range’

Link assignments



What happened to 2001:DB8::/48 ?

- Remember the ‘internal operations’ ?
- Which are the addresses you type most often
    1) DNS resolvers
    2) DNS authorative servers
    3) Loopback addresses
    4) Other servers

Aggregation might be the goal

- But you will always find there must be some leaks
- (Loopback) interfaces to keep protocols going
- Anycasted services or other redundancy
- The odd customer who refused to renumber
- So we'll accept the fact that 2001:DB8::/48 will be broken up
- At the cost of a few additional routes
  - Have easy to remember addresses
  - Which are unlikely to change often

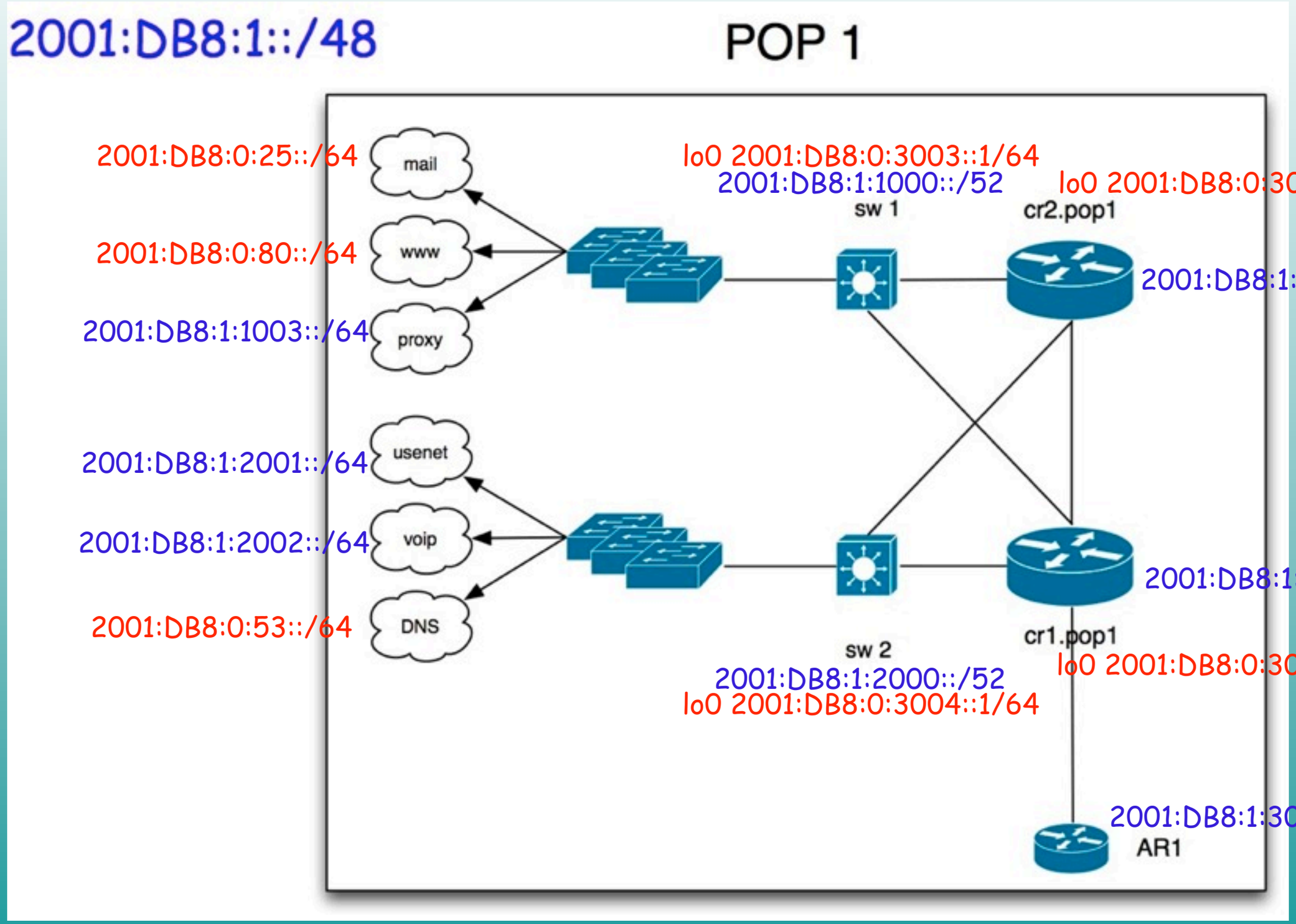
Keep it simple and short (KISS)

- Remember how we are lazy
- And try to limit the amount of typing
- So why not put the resolvers at:
  - 2001:DB8:0:0053::/64
  - maybe keep 2001:DB8:0:1053::/64 as a backup
  - While busy, why not drop
    - mailservers in 0025 and 1025
    - webservers in 0080 and 1080
- May still aggregate into /52 or /56
- Unlikely you fill all 65k subnets

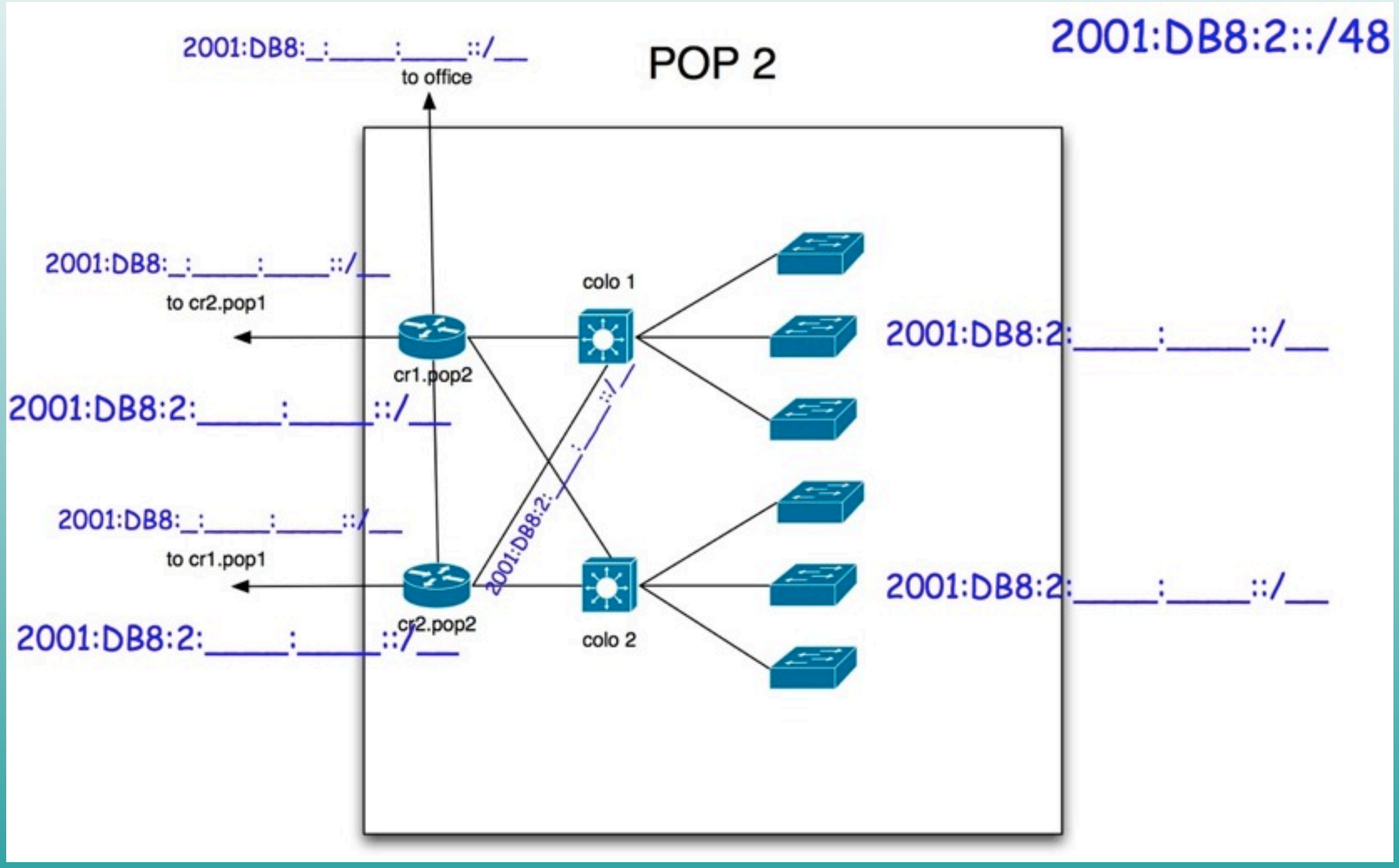
2001:DB8::/48

- 2001:DB8:0:0000::/52 -> servers
- 2001:DB8:0:1000::/52 -> backup servers
- 2001:DB8:0:2000::/52 -> loopbacks
- 2001:DB8:0:3000::/52 -> spare ?
- …
- 2001:DB8:0:f000::/52

Easy addressing in practice



Exercise 2: Assign PoP 2



PoP 2 answers

- cr1.pop2 -> 2001:DB8:2:0100::/56
- cr2.pop2 -> 2001:DB8:2:0200::/56
- colo1 -> 2001:DB8:2:1000::/56
- colo2 -> 2001:DB8:2:2000::/56
- Link to cr1.pop1 -> 2001:DB8:1:01xx::/64
- Link to cr2.pop1 -> 2001:DB8:1:02xx::/64
- Link to office -> 2001:DB8:2:01xx::/64
- Link cr2 to colo1 -> 2001:DB8:02xx::/64

Customer assignments

- 2 types: colo and DSL
- For colo there are 2 options:
  - customers who hook directly to your access box
  - Customers who bring their own router
- For DSL you have to make a decision on how big a single assignment will be
  - Mixing can be done, but could get messy

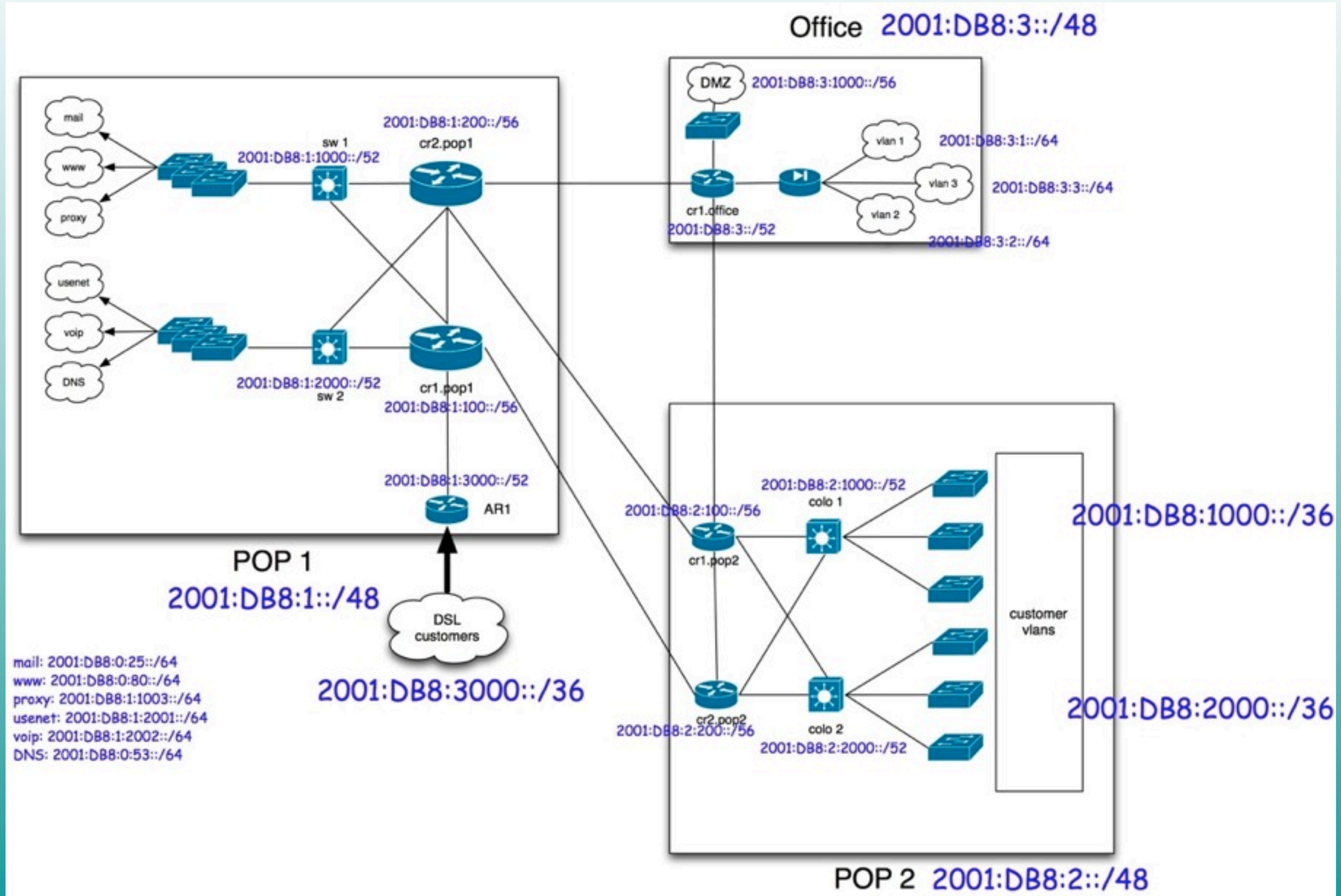
Colocation

- For the links we will use the router's block
  - Allows for 4096 customers per box
  - Allow the customer to address servers in the /64
- For customers with a router
  - Use the router's block for the link
  - Use something completely different and assign a /48
- 2001:DB8:1000::/36 is empty
  - And allows for 4k customers
  - Can be routed entirely to a colo box
- So we'll take 2001:DB8:2000::/36 for the other
  - Or split them up and go /40

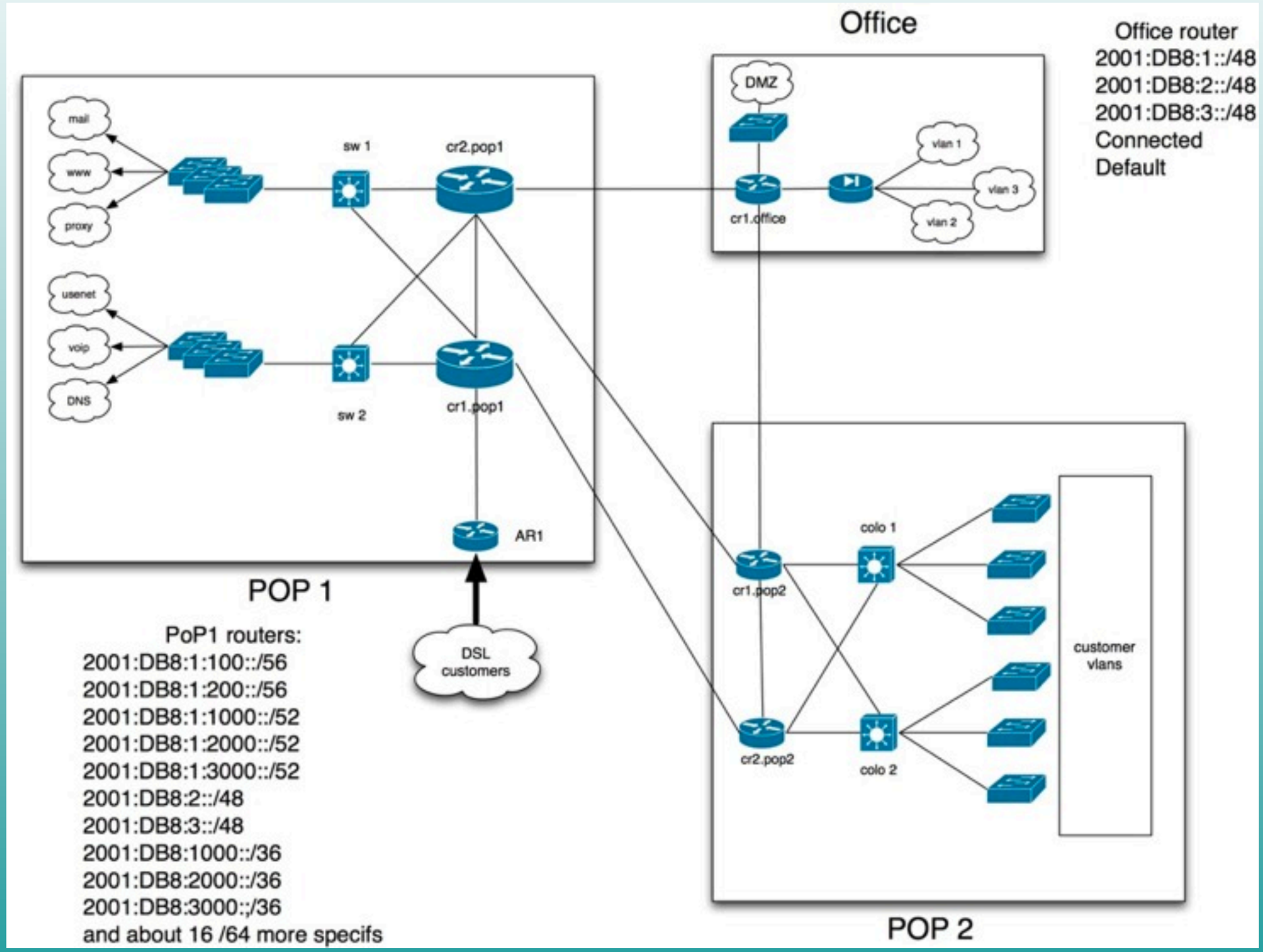
DSL customers

- /36 allows for
  - 4096 /48 assignments
  - and about a million /56’s
- So we’ll take 2001:DB8:3000::/36Worst case we later have to add an additional /36

The completed picture



Did aggregation work ?



Did aggregation work ?

- 2 points of presence and the office
- 8000 colocation customers
- 4000 DSL customers
- In less then 64 routes floating around

Writing it down

- Text might be the easiest
- Remember it goes to ‘F’

```
2001:0DB8::/32                          ALLOCATED
    2001:0DB8:0000::/36                         INTERNAL ASSIGNMENTS
        2001:0DB8:0000:0000::/48                INTERNAL OPERATIONS
            2001:0DB8:0000:0000::/52            CORE SERVICES
            2001:0DB8:0000:1000::/52            BACKUP SERVICES
            2001:0DB8:0000:2000::/52            LOOPBACK INTERFACES
    2001:0DB8:0001:0000::/48                    PoP 1
            2001:0DB8:0001:0100::/56            CR1.POP1
            2001:0DB8:0001:0200::/56            CR2.POP1
        2001:0DB8:0001:1000::/52                SW1.POP1
        2001:0DB8:0001:2000::/52                SW2.POP1
    2001:0DB8:0002:0000::/48                    PoP 2
2001:0DB8:1000::/36                             COLOCATION CUSTOMERS
```

Thank you for attending

- Any questions ?