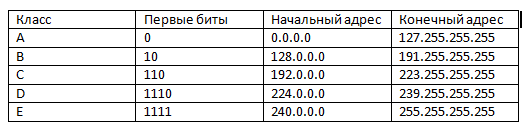
Выше мы говорили о структуре телефонных номеров и их иерархии. И вот на заре рождения Интернета в том представлении, в каком мы его привыкли видеть, возник вопрос. Вопрос заключался в том, что IP-адреса нужно как-то сгруппировать и контролировать выдачу. Решением было разделить все пространство IP-адресов на классы. Это решение получило название **классовая адресация (от англ. Classful)**. Она уже давно устарела, но практически в любой книге на нее отводятся целые главы и разделы. Cisco тоже не забывает про это и в своих учебных материалах рассказывает про нее. Поэтому пробежимся по этой теме и покажем, чем она блистала с 1981 по 1995 год.



Пространство было поделено на 5 классов. Каждому классу был назначен блок адресов.

**Начнем с класса A**. Если внимательно посмотреть на таблицу, то можно заметить, что этому блоку дан самый большой блок адресов, а если быть точным, то половина всего адресного пространства. Предназначался данный класс для крупных сетей. Структура этого класса выглядит следующим образом.

https://habrastorage.org/files/b69/191/ec7/b69191ec7a8e45688114a0055856515e.PNG

В чем суть. Первый октет, то есть 8 бит, остаются за адресом сети, а 3 последних октета (то есть оставшиеся 24 бита) назначаются хостам. Вот для того, чтобы показать, какой кусок относится к сети, а какой к хостам, используется **маска**. По структуре записи она аналогична записи IP-адреса. Отличие маски от IP-адресов в том, что 0 и 1 не могут чередоваться. Сначала идут 1, а потом 0. Таким образом, там, где есть единица, значит это участок сети. Сейчас главное знать, что маска класса A — 255.0.0.0. В таблице еще упомянут какой-то первый бит и для класса A он равен 0. Этот бит как раз нужен для того, чтобы сетевое устройство понимало, к какому классу оно принадлежит. Он же еще задает начальный и конечный диапазон адресов. Если в двоичном виде записать на всех октетах единицы, кроме первого бита в первом октете (там всегда 0), то получится 127.255.255.255, что является границей класса A, например, возьмем адрес 44.58.63.132. Мы знаем, что у класса A первый октет отдается под адрес сети. То есть «44» — это адрес сети, а «58.63.132» — это адрес хоста.

**Поговорим про класс B**

https://habrastorage.org/files/c2f/017/171/c2f01717133f4e829cd29c42f9b44f81.PNG

Этому классу был дан блок поменьше. И адреса из этого блока предназначались для сетей средних масштабов. 2 октета отданы под адрес сети, и 2 — под адрес хостов. Маска у B класса — 255.255.0.0. Первые биты строго 10. А остальные меняются. Перейдем к примеру: 172.16.105.32. Два первых октета под адрес сети — «172.16». А 3-ий и 4-ый под адрес хоста — «105.32».

**Класс C**

https://habrastorage.org/files/35d/2ed/dad/35d2eddadfe24607b5cceb8b7e1bbdff.PNG

Этот класс обделили адресами и дали ему самый маленький блок. Он был предназначен для мелких сетей. Зато этот класс отдавал целых 3 октета под адрес сети и только 1 октет — под хосты. Маска у него — 255.255.255.0. Первые биты 110. На примере это выглядит так — 192.168.1.5. Адрес сети «192.168.1», а адрес хоста «5».

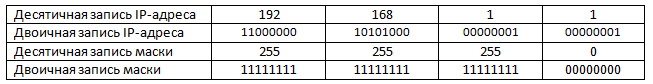
**Классы D и E**. Адреса из этих блоков зарезервированы и не могут назначаться сетям и хостам. Класс D предназначен для многоадресной рассылки. Аналогию можно привести с телевидением. Телеканал вещает группе лиц свой эфир. И те, кто подключены, могут смотреть телепередачи. То есть в распоряжение администраторов могут попасть только 3 первых класса.

Напомним, что первые биты у класса D — это 1110. Пример адреса — 224.0.0.5.

А первые биты у класса E — это 1111. Поэтому, если вдруг увидите адрес вида 240.0.0.1, смело говорите, что это адрес E класса.

Про классы обмолвились. Так зачем тогда маски? У нас итак хосты понимают в каком они классе. Но суть вот в чем. Например, у вас есть маленький офис, и вам нужен блок IP-адресов. Никто не будет вам выдавать все адреса класса C. А дадут только его кусок. Например, 192.168.1.0 с маской 255.255.255.0. Так вот эта маска и будет определять вашу границу. Мы уже говорили, что октет варьируется в значении от 0 до 255. Вот этот 4 октет полностью в вашем распоряжении. За исключением первого адреса и последнего, то есть 0 и 255 в данном случае. Первый адрес — это адрес сети (в данном случае 192.168.1.0), а последний адрес — широковещательный адрес (192.168.1.255). Напомним, что широковещательный адрес используется в том случае, когда надо передать информацию всем узлам в сети. Поэтому есть правило. Если вам надо узнать номер сети, то все биты относящиеся к хосту обращаете в 0, а если широковещательный, то все биты — в 1. Поэтому, если из 256 адресов забирается 2 адреса, то на назначение хостам остается 254 адреса (256 — 2).

*На собеседованиях и экзаменах часто любят спрашивать: «Количество IP-адресов в сети?» и «Сколько доступных IP-адресов в сети для назначения хостам?». Два разных вопроса, которые могут поставить в тупик. Ответом на первый будет — все адреса, включая адрес сети и широковещательный адрес, а на второй вопрос — все адреса, кроме адреса сети и широковещательного адреса.*  
Теперь углубимся в изучении маски.



Адрес класса C 192.168.1.1 с маской 255.255.255.0 в десятичном и двоичном формате. Обратите внимание на то, как выглядит IP-адрес и маска в двоичном формате. Если в IP-адресе 0 и 1 чередуются, то в маске сначала идут 1, а потом 0. Эти биты фиксируют адрес сети и задают размер. По таблице выше можно сделать вывод, что в двоичном виде маска представлена последовательностью 24 единиц подряд. Это говорит о том, что целых 3 октета выделено под сеть, а 4 октет свободен под адресацию для хостов. Здесь ничего необычного. Это стандартная маска класса C.

Но вот в чем загвоздка. Например, в вашем офисе 100 компьютеров, и расширяться вы не планируете. Зачем плодить сеть из 250+ адресов, которые вам не нужны?! На помощь приходит разделение на подсети. Это очень удобная вещь. Объясним принцип на примере того же класса C. Как бы вы не хотели, но трогать 3 октета нельзя. Они фиксированы. Но вот 4 октет свободен под хосты, поэтому его можно трогать. Заимствуя биты из хостового куска, вы дробите сеть на n-ое количество подсетей и, соответственно, уменьшаете в ней количество адресов для хостов.

Попробуем это воплотить в реальность. Меняем маску. Заимствуя первый бит из хостовой части (то есть 1-ый бит 4-ого октета выставляю в единицу). Получается следующая маска.

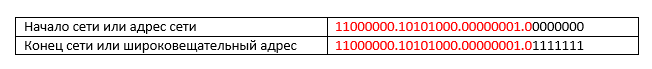
https://habrastorage.org/files/f6f/e49/d41/f6fe49d411c54896a4e0a2244752cb91.PNG

Данная маска делит сеть на 2 части. Если до дробления у сети было 256 адресов (от 0 до 255), то после дробления у каждого куска будет по 128 адресов (от 0 до 127 и от 128 до 255).

Теперь посмотрим, что изменится в целом с адресами.

https://habrastorage.org/files/d97/53f/6ac/d9753f6acbdd4425a8cf0eefd52e10ce.PNG

Красным цветом показано те биты, которые зафиксированы и не могут изменяться. То есть маска ей задает границу. Соответственно биты, помеченные черным цветом определены для адресации хостов. Теперь вычисляем эту границу. Чтобы определить начало, надо все свободные биты (помеченные черным цветом) обратить в ноль, а для определения конца обратить в единицы. Приступаем.



То есть в четвертом октете меняются все биты, кроме первого. Он жестко фиксирован в рамках этой сети.

Теперь посмотрим на вторую половину сети и вычислим ее адреса. Деление у нас производилось заимствованием первого бита в 4-ом октете, значит он является делителем. Первая половина сети получалась, когда этот бит принимал значение 0, а значит вторая сеть образуется, когда этот бит примет значение 1. Обращаю этот бит в 1 и посмотрим на границы.

https://habrastorage.org/files/450/d3c/6e4/450d3c6e48f44beaac8d87fb627578e9.PNG

Приведем в десятичный вид.

https://habrastorage.org/files/600/97d/719/60097d71927d4ac9be95b8dadee07798.PNG

Соответственно .128 и .255 назначать хостам нельзя. Значит в доступности 128-2=126 адресов.

Вот таким образом можно при помощи маски управлять размером сети. Каждый заимствованный бит делит сеть на 2 части. Если откусить 1 бит от хостовой части, то поделим на 2 части (по 128 адресов), 2 бита = 4 части (по 64 адреса), 3 бита = 8 (по 32 адреса) и так далее.

Если вы рассчитали количество бит, отдаваемые под хосты, то количество доступных IP-адресов можно вычислить по формулеhttps://habrastorage.org/files/7ea/bae/3be/7eabae3be92742d6ac822267d18099ba.PNG

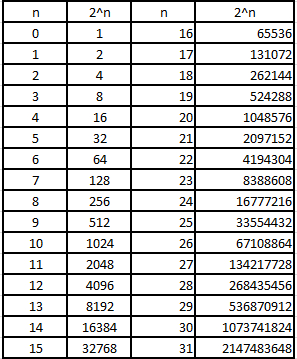
В книге У. Одома по подготовке к CCNA R&S приведена хорошая формула для расчета битов, отдаваемых на подсеть и хосты:  
**N + S + H = 32**, где **N** — кол-во битов сети (класс A — 8 бит, B — 16 бит, C — 24 бита), **S** — кол-во заимствованных битов на подсеть (это то, что мы делали выше, когда заимствовали 1 бит из хостовой части), **H** — кол-во бит отводимых хостам.

Внесем ясность и объясним, как и где применять эти формулы.

Возьмем пример:

Нам выдали сеть 172.16.0.0 и попросили создать 120 подсетей со 180 хостами и записать маску. Приступим.

В качестве шпаргалки, и для быстроты вычисления, ниже воспользуемся таблицей степеней двойки.



Двигаемся дальше. Первое главное условие, при использовании классовой адресации — это то, что должна использоваться одна маска для всех подсетей. То есть, если у вас для одной подсети маска 255.255.255.0, то для другой подсети она не может быть 255.255.255.128.

Теперь смотрим на выданную сеть. Путем логических размышлений понимаем, что это адрес класса B. А значит его N (кол-во битов сети) = 16. Ок. Значит на хосты выделено тоже 16 бит. Вспоминаем условия задачи. Нужно создать 120 подсетей. «Откусывать» биты от сетевой части запрещено, значит кусаем от хостовой части.

Теперь нужно взять такое кол-во бит, чтобы хватило для 120 подсетей, однако оставляло достаточное кол-во под биты для хоста. Смотрим на таблицу выше. Если взять 7 бит, то получим 128. 128>120, следовательно, попадаем под условие. Если возьмем 6 бит, то получим 64. 64<128, поэтому не попадаем под условие и отбрасываем этот вариант.

Ок. Выяснили, что S надо выделить не меньше 7 бит. Теперь посмотрим, что осталось под хосты. Если N + S + H = 32 => H = 32 — (N + S) => H = 32 — (16 + 7) = 9. Смотрим на таблицу выше (или возводим 2 в 9 степень в уме) и получаем число 512. Отнимаем 2 (адрес сети и широковещательный адрес) и получаем 510 адресов. Нам нужно 180, а значит под условие мы попадаем причем с большим запасом. В таких случаях вам предоставляется право выбора. Сделать больше подсетей или хостов на подсеть. Объясняем, что это значит. У нас есть 9 бит на хосты. Если мы возьмем 8 бит, то получим число 256. 256 — 2 = 254 адреса. Этот вариант нам тоже подходит. Возьмем 7 бит. Получаем 128. Даже не отнимая 2 адреса, становится понятно, что это меньше 180 => данный вариант отбрасывается сразу. Итого получаем, что минимальное количество для подсети — 7 бит, а для хостов — 8 бит. Поэтому свободный бит можно отдать либо на подсеть, либо на хосты. Маска получается сложением N и S. В нашем случае получаем, если под подсеть отдаем 7 бит, то получаем 23. В десятичном виде маска будет выглядеть 255.255.254.0. А если отдадим под подсеть 8 бит, то получим 24 (или в десятичном виде 255.255.255.0). Иногда бывает, что под задачу существует всего одна маска. Ну и, конечно, могут быть случаи, когда маска не попадает не под какие условия. В этих случаях нужно брать сеть другого класса или доказывать заказчику, что это невозможно.

Теперь понятно, как работала классовая адресация, и как ее рассчитывали. Возможно с первого раза голова не переварит этого, поэтому перечитывайте еще раз и повнимательнее. Как только начнет что-то проясняться, потренируйтесь на задачках.

**Задача №2**

1) Записать маску для проекта: сеть 172.16.0.0. 250 подсетей и 220 хостов.  
2) Записать маску для проекта: сеть 10.0.0.0. 2000 подсетей и 1500 хостов.  
3) Записать маску для проекта: сеть 192.168.0.0. 4 подсети и 60 хостов.

*Классовая адресация* — это зарождение сегодняшнего интернета, и именно с нее все началось. Поэтому плюсов у нее много, и за это создателям спасибо. Но, как вы могли заметить, у нее было жесткая привязка к одной маске. За счет этого IP-адреса использовались не экономно и расточительно. А в связи с бурным ростом Интернета адресов стало не хватать, и срочно нужно было вносить изменения.