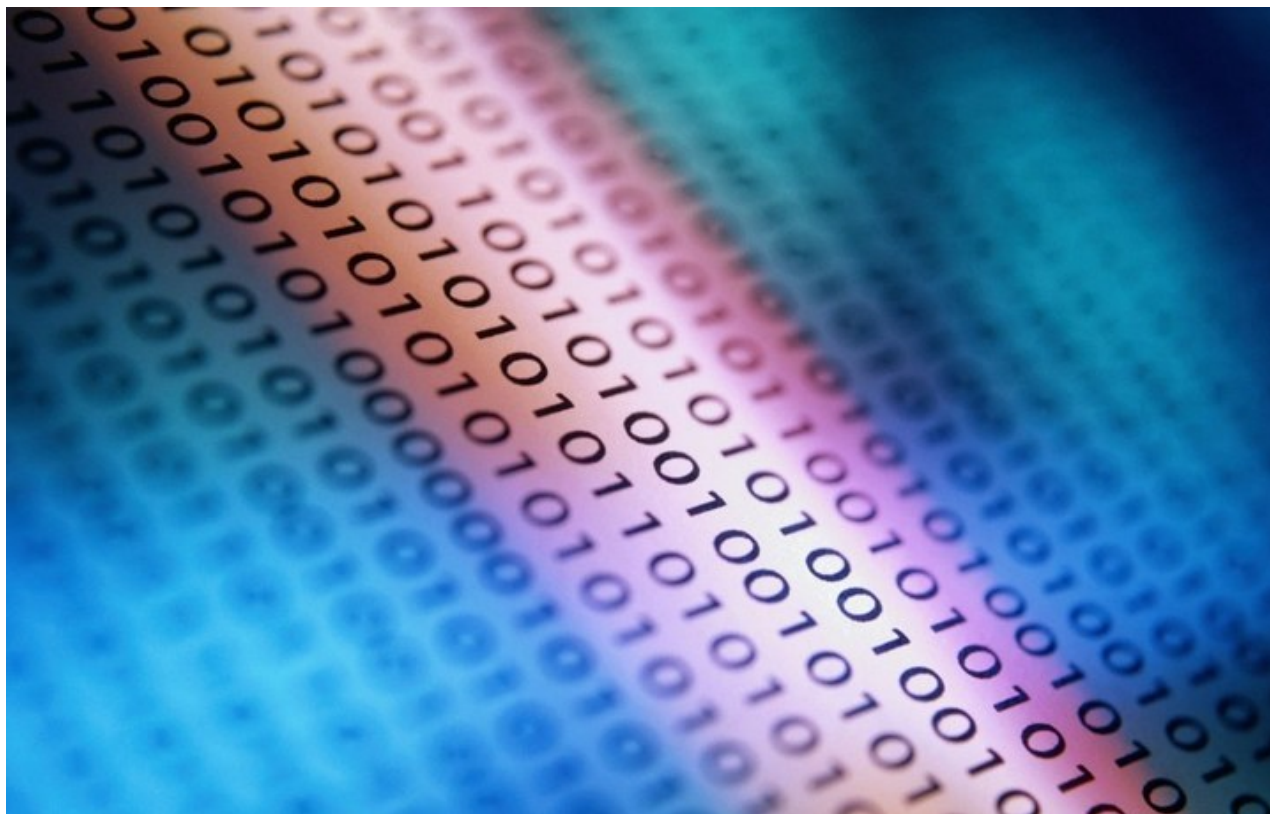


# Основы компьютерных сетей. Тема №5. Понятие IP адресации, масок подсетей и их расчет / Хабр

 [habr.com/ru/articles/314484](https://habr.com/ru/articles/314484)

Денис

15 декабря 2016 г.



Приветствую вас на очередном выпуске. И сегодня речь пойдет о том, какие бывают IP-адреса, и как ими пользоваться. Что такое маска подсети, как она считается, и для чего она нужна. Как делить сети на подсети и суммировать их. Заинтересовавшихся приглашаю к прочтению.

## Содержание

Начнем, или уже продолжим, с самого популярного, заезженного и больного. Это IP-адреса. На протяжении 4-х статей это понятие встречалось по несколько раз, и скорее всего вы уже либо сами поняли для чего они, либо наугуглили и почитали о них. Но я обязан вам это рассказать, так как без ясного понимания двигаться дальше будет тяжело.

Итак IP-адрес — это адрес, используемый узлом на сетевом уровне. Он имеет иерархическую структуру. Что это значит? Это значит, что каждая цифра в его написании несет определенный смысл. Объясню на очень хорошем примере. Примером будет номер обычного телефона — +74951234567. Первой цифрой идет +7. Это говорит о том, что номер принадлежит зоне РФ. Далее следует 495. Это код

Москвы. И последние 7 цифр я взял случайными. Эти цифры закреплены за районной зоной. Как видите здесь наблюдается четкая иерархия. То есть по номеру можно понять какой стране, зоне он принадлежит. IP адреса придерживаются аналогично строгой иерархии. Контролирует их организация IANA(англ. Internet Assigned Numbers Authority). Если на русском, то это «Администрация адресного пространства Интернет». Заметьте, что слово «Интернет» с большой буквы. Мало кто придает этому значение, поэтому объясню разницу. В англоязычной литературе термин «internet» используется для описания нескольких подключённых друг к другу сетей. А термин «Internet» для описания глобальной сети. Так что примите это к сведению.

Несмотря на то, что тема статьи больше теоретическая, нежели практическая, я настоятельно рекомендую отнестись к ней со всей серьезностью, так как от нее зависит понимание дальнейших тем, а особенно маршрутизации. Не для кого, я думаю, не секрет, что мы привыкли воспринимать числовую информацию в десятичном формате (в числах от 0-9). Однако все современные компьютеры воспринимают информацию в двоичном (0 и 1). Не важно при помощи тока или света передается информация. Вся она будет воспринята устройством как есть сигнал (1) или нет (0). Всего 2 значения. Поэтому был придуман алгоритм перевода из двоичной системы в десятичную, и обратно. Начну с простого и расскажу, как выглядят IP адреса в десятичном формате. Вся эта статья посвящена IP адресам версии 4. О версии 6 будет отдельная статья. В предыдущих статьях, лабах, да и вообще в жизни, вы видели что-то вроде этого «193.233.44.12». Это и есть IP адрес в десятичной записи. Состоит он из 4-х чисел, называемых октетами и разделенных между собой точками. Каждое такое число (октет) может принимать значение от 0 до 255. То есть одно из 256 значений. Длина каждого октета равна 8 битам, а суммарная длина IPv4 = 32 битам. Теперь интересный вопрос. Каким образом этот адрес воспримет компьютер, и как будет с ним работать?

Можно конечно набить это в калькулятор, коих навалом в Интернете, и он переведет его в двоичный формат, но я считаю, что переводить вручную должен уметь каждый. Особенно это касается тех, кто планирует сдавать экзамен. У вас не будет под рукой ничего, кроме бумаги и маркера, и полагаться придется только на свои навыки. Поэтому показываю, как это делать вручную. Строится таблица.

---

128	64	32	16	8	4	2	1
x	x	x	x	x	x	x	x

---

Вместо «х» записывается либо 1, либо 0. Таблица разделена на 8 колонок, каждая из которых несет в себе 1 бит (8 колонок = 8 бит = 1 октет). Расположены они по старшинству слева направо. То есть первый (левый) бит — самый старший и имеет номер 128, а последний (правый) — самый младший и имеет номер 1. Теперь объясню, откуда эти числа взялись. Так как система двоичная, и длина октета равна 8-ми битам, то каждое число получается возведением числа 2 в степень от 0 до 7. И

каждая из полученных цифр записывается в таблицу от большего к меньшему. То есть слева направо. От 2 в 7-ой степени до 2 в 0-ой степени. Приведу таблицу степеней 2-ки.

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
128	64	32	16	8	4	2	1

Думаю теперь понятно, каким образом строится таблица. Давайте теперь разберем адрес «193.233.44.12» и посмотрим, как он выглядит в двоичном формате. Разберем каждый октет отдельно. Возьмем число 193 и посмотрим, из каких табличных комбинаций оно получается.  $128 + 64 + 1 = 193$ .

<b>128</b>	<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>
1	1	0	0	0	0	0	1

Те числа, которые участвовали в формировании комбинации получают 1, а все остальные получают 0.

Берем первый октет 233.  $128 + 64 + 32 + 8 + 1$ .

<b>128</b>	<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>
1	1	1	0	1	0	0	1

Для 44 — это  $32 + 8 + 4$ .

<b>128</b>	<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>
0	0	1	0	1	1	0	0

И напоследок 12.  $8 + 4$ .

<b>128</b>	<b>64</b>	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>
0	0	0	0	1	1	0	0

Получается длинная битовая последовательность

11000001.11101001.00101100.00001100. Именно с данным видом работают сетевые устройства. Битовая последовательность обратима. Вы можете так же вставить каждый октет (по 8 символов) в таблицу и получить десятичную запись. Я представлю совершенно случайную последовательность и приведу ее к десятичному виду. Пусть это будет 11010101.10110100.11000001.00000011. Строю таблицу и заносу в нее первый блок.

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

Получаю  $128 + 64 + 16 + 4 + 1 = 213$ .

Вычисляю второй блок.

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

Считаю  $128 + 32 + 16 + 4 = 180$ .

Третий блок.

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

$128 + 64 + 1 = 193$ .

И напоследок четвертый.

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

$2 + 1 = 3$

Собираем результаты вычислений и получаем адрес 213.180.193.3. Ничего тяжелого, чистая арифметика. Если тяжело и прям невыносимо трудно, то попрактикуйтесь. Сначала может показаться страшным, так как многие закончили учебу лет 10 назад и многое позабыли. Но уверяю, что как только набьете руку, считать будет гораздо легче. Ну а для закрепления дам вам несколько примеров для самостоятельного расчета (под спойлером будут ответы, но открывайте их только когда прорешаете сами).

## Задача №1

- 1) 10.124.56.220
- 2) 113.72.101.11
- 3) 173.143.32.194
- 4) 200.69.139.217
- 5) 88.212.236.76
- 6) 01011101.10111011.01001000.00110000
- 7) 01001000.10100011.00000100.10100001
- 8) 00001111.11011001.11101000.11110101
- 9) 01000101.00010100.00111011.01010000
- 10) 00101011.11110011.10000010.00111101

## Ответы

Теперь IP-адреса не должны быть чем-то страшным, и можно углубиться в их изучение.

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 я пробежусь по этой теме и покажу, чем она блистала с 1981 по 1995 год.

Класс	Первые биты	Начальный адрес	Конечный адрес
A	0	0.0.0.0	127.255.255.255
B	10	128.0.0.0	191.255.255.255
C	110	192.0.0.0	223.255.255.255
D	1110	224.0.0.0	239.255.255.255
E	1111	240.0.0.0	255.255.255.255

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

**Начнем с класса А.** 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.

Класс А			
Сеть	Хост	Хост	Хост

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. Чуть ниже, после разбора классов, я покажу, как с ней работать. Сейчас главное знать, что маска класса А — 255.0.0.0. В таблице еще упомянут какой-то первый бит и для класса А 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 А. Например, возьмем адрес 44.58.63.132. Мы знаем, что у класса А первый октет отдается под адрес сети. То есть «44» — это адрес сети, а «58.63.132» — это адрес хоста.

## Поговорим про класс В

Класс В			
Сеть	Сеть	Хост	Хост

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 В 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 С

Класс С			
Сеть	Сеть	Сеть	Хост

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.** Я неспроста объединил их в один. 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.

Напомню, 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. Теперь озвучу вопрос, который мне недавно задали. 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). Напомню, 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.

Десятичная запись IP-адреса	192	168	1	1
Двоичная запись IP-адреса	11000000	10101000	00000001	00000001
Десятичная запись маски	255	255	255	0
Двоичная запись маски	11111111	11111111	11111111	00000000

Я записал 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. Объясню 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. Меняю маску. Заимствую первый бит из хостовой части(то есть 1-ый бит 4-ого октета выставлю в единицу). Получается следующая маска.

Двоичная запись маски	11111111.11111111.11111111.10000000
Десятичная запись маски	255.255.255.128

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 посмотрю, что изменится в целом с адресами.

Двоичная запись IP-адреса 192.168.1.1	11000000.10101000.00000001.00000001
Двоичная запись маски 255.255.255.128	11111111.11111111.11111111.10000000

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

Начало сети или адрес сети	11000000.10101000.00000001.00000000
Конец сети или широковещательный адрес	11000000.10101000.00000001.01111111

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 посмотрю на границы.

Начало сети или адрес сети	11000000.10101000.00000001.10000000
Конец сети или широковещательный адрес	11000000.10101000.00000001.11111111

Приведу в десятичный вид.

Начало сети или адрес сети в 10-ом формате	192.168.1.128
Конец сети или широковещательный адрес в 10-ом формате	192.168.1.255

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 формуле  $2^H - 2$

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.

Внесу ясность и объясню, 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, я ниже подготовил таблицу степеней двойки.

n	2 <sup>n</sup>	n	2 <sup>n</sup>
0	1	16	65536
1	2	17	131072
2	4	18	262144
3	8	19	524288
4	16	20	1048576
5	32	21	2097152
6	64	22	4194304
7	128	23	8388608
8	256	24	16777216
9	512	25	33554432
10	1024	26	67108864
11	2048	27	134217728
12	4096	28	268435456
13	8192	29	536870912
14	16384	30	1073741824
15	32768	31	2147483648

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 В. 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  $N + S + H = 32 \Rightarrow H = 32 - (N + S) \Rightarrow H = 32 - (16 + 7) = 9$ . 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. Объясняю, 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  $\Rightarrow$  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.

Думаю 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 задачах, которые я оставляю.

## 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.

## Ответы на задачи

На этом разговор про классы сети начну закруглять и подведу итоги. 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 была 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.

Поняли ведущие умы, что использовать классы сети не удобно и нужно от них отказываться. Это привело к созданию бесклассовой адресации и маскам переменной длины, о чем мы ниже поговорим. Но перед этим пару слов о видах IP-адресов. Несмотря на то, что переход от классовой адресации к бесклассовой предполагал экономию IP-адресов, на деле эта проблема все равно решалась не

полностью. Все упиралось в саму технологию IPv4. Объясню почему. Выше я говорил, что длина IP адреса равна 32 бита. Каждый бит может принимать значение 0 или 1, то есть два значения. Соответственно, чтобы вычислить все комбинации, надо возвести 2 в 32-ую степень. Получаем 4294967296 адресов. Если вычесть отсюда зарезервированные для специальных нужд и прочего, то останется примерно 4.2 млрд. адресов, когда на Земле проживает около 7.3 млрд. человек. Поэтому ведущие умы быстро просекли эту фишку и начали искать решение. Они решили выделить некое адресное пространство, которое будет использоваться только в пределах локальной сети и не будет использоваться в Интернете. Это разделило адреса на 2 лагеря: белые или публичные (англ. public) и серые или частные (англ. private).

Привожу диапазон адресов, которые выделены под локальные сети:

- 1) 10.0.0.0 — 10.255.255.255 с маской 255.0.0.0 (или кратко 10/8).
- 2) 172.16.0.0 — 172.31.255.255 с маской 255.240.0.0 (или кратко 172.16/12).
- 3) 192.168.0.0 — 192.168.255.255 (или кратко 192.168/16).

Если честно, я мало где видел применение адресации 172.16.X.X. Обычно в корпоративной среде всегда используется 10.X.X.X, а в домах/квартирах и мелких офисах 192.168.X.X.

Теперь прошу обратить внимание на очень важную вещь, которую многие путают. Не путайте классовую адресацию и диапазон частных адресов. Очень много людей наступают на эти грабли и свято верят, что диапазон частных адресов 10.0.0.0 — 10.255.255.255 — это диапазон А класса.

Разобрались, что такое частные адреса или private адреса. Но это еще не все. Есть еще список зарезервированных адресов, которые не могут светиться в Интернете. По ним написана целая документация на IETF. Привожу [ссылку](#), где можете прочитать оригинал. Я кратко опишу часто встречающиеся.

1) 0.0.0.0/8 — диапазон адресов, используемый хостами для самоидентификации. Обычно это можно увидеть, когда хост пытается получить IP-адрес от DHCP сервера. Так как изначально у него нету IP-адреса, то в поле источника он вставляет адрес из данного диапазона.

2) 127.0.0.0/8 — loopback или localhost адреса. Это IP-адреса, используемые компьютером, чтобы обратиться к самому себе. Очень полезно для проверки работы TCP/IP. Дело в том, что независимо от наличия соединения с Интернетом или локальной сетью, адреса из этого пула должны всегда пинговаться. Если этого не происходит, значит система накрылась или накрывается медным тазом.

3) 169.254.0.0/16 — link-local address или локальные адреса. Автоматически используются хостами при отсутствии DHCP-сервера или его недоступности. Это позволяет быстро организовать локальную сеть и проверить работу узлов. Однако данный пул адресов не маршрутизируется. Следовательно, выйти в Интернет с них не получится.

4) 224.0.0.0/4 — блок адресов, зарезервированный под многоадресную рассылку или multicast. Для тех, кто хочет побольше узнать про multicast, оставляю [ссылку](#).

### **Бесклассовая адресация (англ. Classless Inter-Domain Routing или CIDR).**

Описана была в стандарте RFC1519 в 1993 году. Она отказалась от классовых рамок и фиксированной маски. Адреса делятся только на публичные и зарезервированные, о которых написано выше. Если в классовой адресации маска нарезалась единой для всех подсетей, то в бесклассовой — у каждой подсети может быть своя маска. На теории все хорошо и красиво, но нет ничего лучше, чем практика. Поэтому перехожу к ней и объясню, как можно делить на подсети с разным количеством хостов.

В качестве шпаргалки приведу список всех возможных масок.

Представим ситуацию. Вам выдали сеть 192.168.1.0/24 и поставили следующие условия:

- 1) Подсеть на 10 адресов для гостей.
- 2) Подсеть на 42 адреса для сотрудников.
- 3) Подсеть на 2 адреса для соединения 2 маршрутизаторов.
- 4) Подсеть на 26 адресов для филиала.

Ок. Данная маска показывает, что в нашем распоряжении находятся 256 адресов. По условию эту сеть надо каким-то образом разделить на 4 подсети. Давайте попробуем. 256 очень хорошо делится на 4, давая в ответе 64. Значит один большой блок в 256 адресов можно поделить на 4 равных блока по 64 адреса в каждом. И все было бы прекрасно, но это порождает большое число пустых адресов. Для сотрудников, которым нужно 42 адреса, ладно, может в дальнейшем компания еще наймет. Но вот подсеть для маршрутизаторов, которая требует всего 2 адреса, оставит 60 пустых адресов. Да, вы можете сказать, что это private адреса, и кому дело до них. А теперь представьте, что это публичные адреса, которые маршрутизируются в Интернете. Их и так мало, а тут мы еще будем их отбрасывать. Это не дело, тем более, когда мы можем гибко управлять адресным пространством. Поэтому возвращаемся к примеру и нарежем подсети так, как нам нужно.

Итак, какие подсети должны быть нарезаны, чтобы вместились все адреса, заданные по условию?!

- 1) Для 10 хостов, наименьшей подсетью будет блок из 16 адресов.
- 2) Для 42 хостов, наименьшей подсетью будет блок из 64 адресов.
- 3) Для 2 хостов, наименьшей подсетью будет блок из 4 адресов.
- 4) Для 26 хостов, наименьшей подсетью будет блок из 32 адресов.

Я понимаю, что не все могут с первого раза в это вникнуть, и в этом нет ничего страшного. Все люди разные и по-разному воспринимают информацию. Для полноты эффекта покажу деление на картинке.

Вот у нас блок, состоящий из 256 адресов.

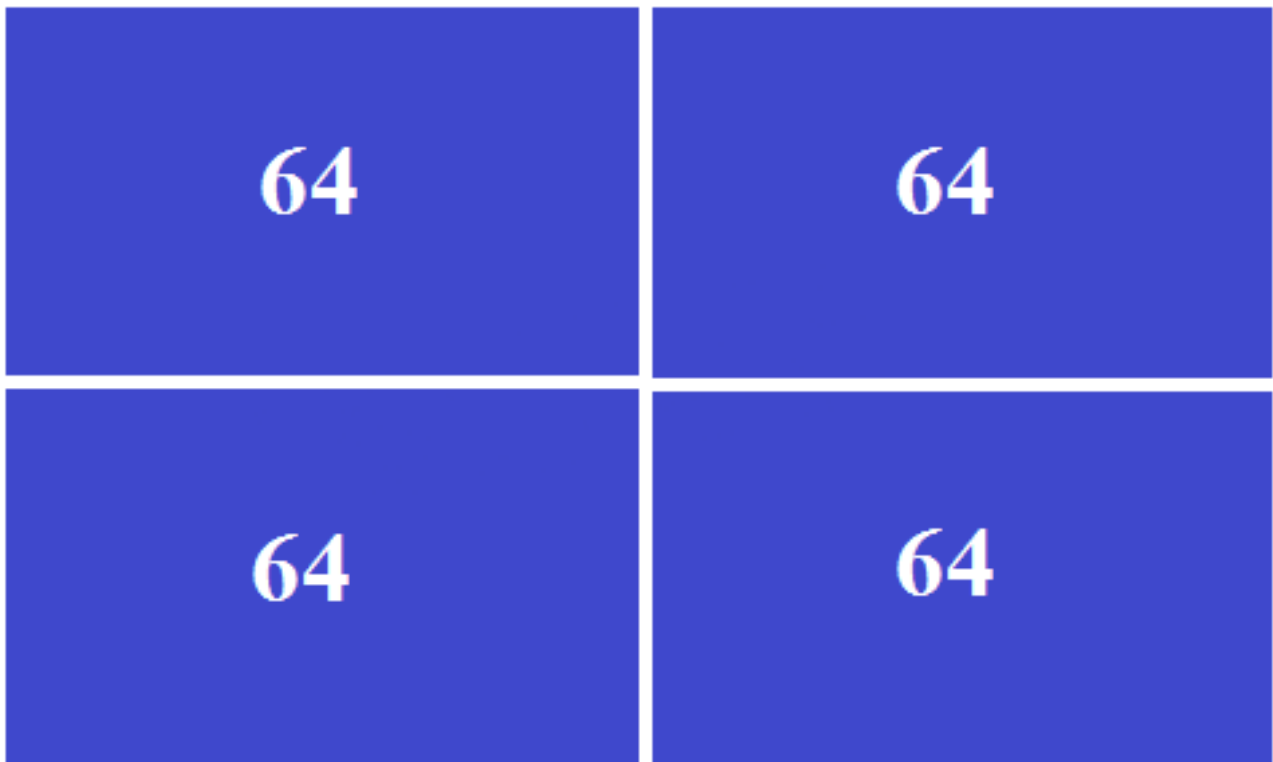
255.255.255.255	/32
255.255.255.254	/31
255.255.255.252	/30
255.255.255.248	/29
255.255.255.240	/28
255.255.255.224	/27
255.255.255.192	/26
255.255.255.128	/25
255.255.255.0	/24
255.255.254.0	/23
255.255.252.0	/22
255.255.248.0	/21
255.255.240.0	/20
255.255.224.0	/19
255.255.192.0	/18
255.255.128.0	/17
255.255.0.0	/16
255.254.0.0	/15
255.252.0.0	/14
255.248.0.0	/13
255.240.0.0	/12
255.224.0.0	/11
255.192.0.0	/10
255.128.0.0	/9
255.0.0.0	/8
254.0.0.0	/7
252.0.0.0	/6
248.0.0.0	/5
240.0.0.0	/4
224.0.0.0	/3
192.0.0.0	/2



192.0.0.0	/2
128.0.0.0	/1
0.0.0.0	/0



После деления на 4 части получается следующая картинка.



Выше мы выяснили, что при таком раскладе адреса используются не рационально.

Теперь обратите внимание, как стало выглядеть адресное пространство после нарезки подсетей разной длины.



Как видите, в свободном доступе осталось куча адресов, которые мы в дальнейшем сможем использовать. Можно посчитать точную цифру.  $256 - (64 + 32 + 16 + 4) = 140$  адресов.

Вот столько адресов мы сэкономили. Двигаемся дальше и ответим на следующие вопросы:

- Какими будут сетевые и широковещательные адреса?
- Какие адреса можно будет назначить хостам?
- Как будут выглядеть маски?

Механизм деления на подсети с разной маской получил название **VLSM (от англ. Variable Length Subnet Mask)** или **маска подсети переменной длины**. Дам важный совет! Начинайте адресацию с самой большой подсети. Иначе вы можете попасть на то, что адреса начнут перекрываться. Поэтому сначала планируйте сеть на бумаге. Нарисуйте ее, изобразите в виде фигур, просчитайте вручную или на калькуляторе и только потом переходите на настройку в боевых условиях.

Итак, самая большая подсеть состоит из 64 адресов. С нее и начнем. Первый пул адресов будет следующий:

Адрес подсети — 192.168.1.0.

Широковещательный адрес — 192.168.1.63.

Пул адресов для назначения хостам от 192.168.1.1 до 192.168.1.62.

Теперь выбор маски. Тут все просто. Отнимаем от целой сети нужный кусок и полученное число записываем в октет маски. То есть  $256 - 64 = 192 \Rightarrow$  маска 255.255.255.192 или /26.

Дальше идет подсеть поменьше. Состоит она из 32 адресов. Если первая заканчивалась на .63, то эта будет начинаться с .64:

Адрес подсети — 192.168.1.64.

Широковещательный адрес — 192.168.1.95.

Пул адресов для назначения хостам будет от 192.168.1.65 до 192.168.1.94.

Маска:  $256 - 32 = 224 \Rightarrow$  255.255.255.224 или /27.

3-я подсеть, которая предназначена для филиала, начнет старт с .96:

Адрес подсети — 192.168.1.96.

Широковещательный адрес — 192.168.1.111.

Пул адресов для назначения хостам будет от 192.168.1.97 до 192.168.1.110.

Маска:  $256 - 16 = 240 \Rightarrow$  255.255.255.240 или /28.

Ну и для последней подсети, которая уйдет под интерфейсы, соединяющие роутеры, будет начинаться с .112:

Адрес подсети — 192.168.1.112.

Широковещательный адрес — 192.168.1.115.

Разрешенными адресами будут 192.168.1.113 и 192.168.1.114.

Маска:  $256 - 4 = 252 \Rightarrow$  255.255.255.252 или /30.

Замечу, что адрес 192.168.1.115 является последним используемым адресом. Начиная с 192.168.1.116 и до .255 свободны.

Вот таким образом, при помощи VLSM или масок переменной длины, мы экономно создали 4 подсети с нужным количеством адресов в каждой. Думаю это стоит закрепить задачей для самостоятельного решения.

### **Задача №3**

Разделите сеть 192.168.1.0/24 на 3 разные подсети. Найдите и запишите в каждой подсети ее адреса, широковещательный адрес, пул разрешенных к выдаче адресов и маску. Указываю требуемые размеры подсетей:

- 1) Подсеть на 120 адресов.
- 2) Подсеть на 12 адресов.
- 3) Подсеть на 5 адресов.

### **Ответ**

Теперь, когда вы знаете, как делить сети на подсети, самое время научиться

собирают подсети в одну общую подсеть. Иначе это называется **суммированием** или **summarization**. Суммирование чаще всего используется в маршрутизации. Когда у вас в таблице маршрутизатора несколько соседних подсетей, маршрутизация которых проходит через один и тот же интерфейс или адрес. Скорее всего этот процесс лучше объяснить при разборе маршрутизации, но учитывая то, что тема маршрутизации и так большая, то я объясню процесс суммирования в этой статье. Тем более, что суммирование это сплошная математика, а в этой статье мы ею и занимаемся. Ну что же, приступлю.

Представим, что у меня компания состоящая из главного здания и корпусов. Я работаю в главном здании, а в корпусах коллеги. Хотя у меня и главное здание, но в нем всего 4 подсети:

— 192.168.0.0/24  
 — 192.168.1.0/24  
 — 192.168.2.0/24  
 — 192.168.3.0/24

Тут коллеги с соседнего здания очухались и поняли, что у них слетела конфигурация на маршрутизаторе, а бекапов нет. Наизусть они не помнят, какие в главном здании подсети, но помнят, что они находятся рядом друг с другом, и просят прислать одну суммированную. Теперь у меня возникает задача, как их суммировать. Для начала я переведу все подсети в двоичный вид.

Подсеть №1	192	168	0	0
Двоичный вид	11000000	10101000	00000000	00000000
Подсеть №2	192	168	1	0
Двоичный вид	11000000	10101000	00000001	00000000
Подсеть №3	192	168	2	0
Двоичный вид	11000000	10101000	00000010	00000000
Подсеть №4	192	168	3	0
Двоичный вид	11000000	10101000	00000011	00000000
Подсеть №5	192	168	4	0
Двоичный вид	11000000	10101000	00000100	00000000

Посмотрите внимательно на таблицу. Как видите, у 4 подсетей первые 22 бита одинаковые. Соответственно, если я возьму 192.168.0.0 с маской /22 или 255.255.252.0, то покрою свои 4 подсети. Но обратите внимание на 5 подсеть, которую я специально ввел. Это подсеть 192.168.4.0. 22-ой бит у нее отличается от предыдущих 4-х, а значит выше выбранное не покрывает эту подсеть. Ок. Теперь я отправлю коллегам суммированную подсеть, и, если они все правильно пропишут, то маршрутизация до моих подсетей будет работать без проблем.

Возьмем тот же пример и немного изменим условия. Нас попросили прислать суммарный маршрут для подсетей 192.168.0.0 и 192.168.1.0. Я не поленюсь и создам еще одну таблицу.

Подсеть №1	192	168	0	0
Двоичный вид	11000000	10101000	00000000	00000000
Подсеть №2	192	168	1	0
Двоичный вид	11000000	10101000	00000001	00000000
Подсеть №3	192	168	2	0
Двоичный вид	11000000	10101000	00000010	00000000
Подсеть №4	192	168	3	0
Двоичный вид	11000000	10101000	00000011	00000000

Обратите внимание, что у 2 первых подсетей одинаковые не 22 бита, а 23 бита. Это значит, что их можно просуммировать еще компактнее. В принципе работать будет и так, и так. Но как говорилось в одной рекламе: «Если нет разницы — зачем платить больше?». Поэтому старайтесь суммировать, не задевая при этом соседние подсети.

Таким образом, переводя подсети в двоичный формат и находя одинаковые биты, можно их суммировать.

Вообще суммирование полезно применять, когда надо объединить несколько подсетей, расположенных вблизи друг с другом. Это позволит сэкономить ресурсы маршрутизаторов. Однако это не всегда возможно. Просуммировать, например, подсеть 192.168.1.0 и 192.168.15.0, не захватив при этом соседние подсети, невозможно. Поэтому перед суммированием стоит подумать над ее целесообразностью. Поэтому повторяюсь еще раз, что начинать какую-либо революцию надо на бумажке. Ну и для закрепления материала оставлю небольшую задачу.

#### Задача №4

Даны 4 подсети:

- 1) 10.3.128.0
- 2) 10.3.129.0
- 3) 10.3.130.0
- 4) 10.3.131.0

Просуммируйте подсети и найдите маску, которая сможет покрыть их, не задевая при этом соседние подсети.

#### Ответ

Подсеть №1	10	3	128	0
Двоичный вид	00001010	00000011	10000000	00000000
Подсеть №2	10	3	129	0
Двоичный вид	00001010	00000011	10000001	00000000
Подсеть №3	10	3	130	0
Двоичный вид	00001010	00000011	10000010	00000000
Подсеть №4	10	3	131	0
Двоичный вид	00001010	00000011	10000011	00000000

Исходя из этого, ответом будет 10.3.128.0/22 (255.255.252.0)

Пришло время закругляться. Статья получилась не очень длинной. Я бы даже сказал наоборот. Но все, что требует знать Cisco про IPv4, мы рассмотрели. Самое главное, что требуется от вас — это научиться работать с адресами и масками и уметь конвертировать их из десятичной в двоичную и обратно. Ну и, конечно, правильно делить на подсети и распределять адресное пространство. Спасибо, что дочитали. А если еще и задачки все сами прорешали, то цены вам нет) А если еще не прорешали, то приятного времяпровождения.