

Правила назначения IP адресов.

Из рассмотренного ранее материала известно, что для взаимодействия с использованием протокола Internet Protocol необходимо чтобы IP адреса были назначены каждому сетевому интерфейсу узла и каждому сетевому интерфейсу маршрутизатора. При этом необходимо придерживаться следующих обязательных правил (об исключениях мы поговорим позже):

- Адреса не должны дублироваться: IP адрес – уникальный идентификатор узла или порта маршрутизатора
- Если узлы и порты маршрутизаторов находятся в одной канальной сети, то они должны иметь такие IP адреса, которые принадлежат одной IP сети
- Если узлы и порты маршрутизаторов находятся в разных канальных сетях, то они должны иметь такие IP адреса, которые принадлежат разным IP сетям

Первое правило – очевидно. Рассмотрим подробнее второе и третье правило.

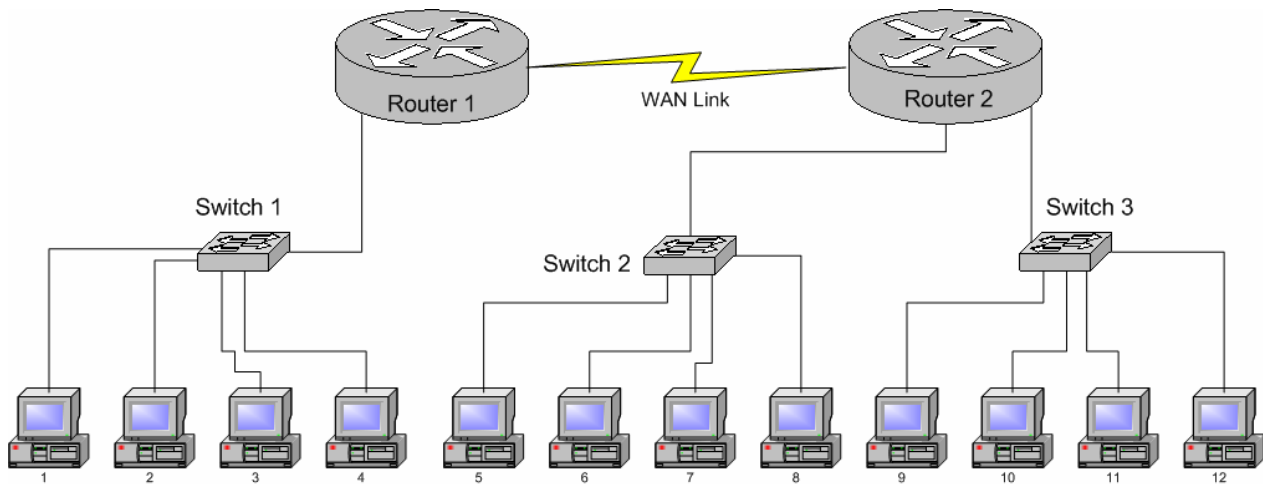
Когда узел передает данные другому узлу, он не знает, в одной канальной сети он с ним находится или в разных. Единственное, что знает отправитель пакета о получателе, это его IP адрес, кроме того, отправитель пакета знает свой IP адрес. Из своего адреса отправитель может узнать номер своей сети, пользуясь техникой классов, из IP адреса получателя отправитель может узнать номер его сети: если они совпадают – значит два узла в одной IP сети, значит можно передавать кадр канального уровня непосредственно получателю, иначе его необходимо передавать маршрутизатору.

Если два узла будут находиться в разных канальных сетях, но иметь адреса из одной IP сети, то отправитель подумает, что получатель с ним в одной канальной сети, пошлет ему кадр канального уровня, который, очевидно не будет получен. И обратное: если отправитель и получатель находятся в одной канальной сети, но их IP адреса заданы таким образом, что отправитель думает, что они в разных канальных сетях, то вместо передачи кадра непосредственно получателю, кадр будет передан маршрутизатору, что по меньшей мере не желательно а в худшем случае приведет к невозможности доставки IP пакета.

То же касается и маршрутизаторов: из канальных адресов узлов не следует ничего об их принадлежности к той или иной сети, лишь IP адреса позволяют маршрутизаторам делать выводы о составе КАНАЛЬНЫХ сетей, и соответственно IP адреса должны быть назначены таким образом, чтобы отражать, кто из узлов принадлежит какой канальной сети.

Рассмотрим пример назначения IP адресов в небольшой сети, используя указанные выше правила:

Пусть у нас есть небольшое предприятие, имеющее офис в центре города и подразделение на окраине. Очевидно, что построить одну канальную сеть, охватывающую все узлы предприятия невозможно (АТМ исключаем для упрощения задачи), для решения задачи нам потребуется привлечение средств третьего уровня. В главном офисе сеть разделена на две сети Ethernet, так как того требует политика безопасности, сеть склада состоит из единственной сети Ethernet. В офисе и на складе установлены маршрутизаторы, которые связаны между собой арендованным каналом 4DS0. Задача – адресовать эту сеть. Нам понадобится присвоить адреса всем узлам и портам маршрутизаторов данной составной сети в соответствии с рассмотренной выше стратегией.



В качестве адресов сетей нужно выбрать адреса класса C, так как все наши сети имеют размер меньше 254 узлов. Обратите внимание – использование адресного пространства в данном случае не является рациональным, так как три наши сети содержат всего по четыре узла (+порт маршрутизатора!), но более мелких IP сетей, нежели размером 254 узла не существует. Впрочем, при добавлении новых узлов в наши сети у нас не будет проблем с присвоением IP адресов новым узлам. И еще один нюанс - глобальная связь между маршрутизаторами – это тоже сеть и тоже нуждается в адресации, но в ней по определению два узла, однако она тоже потребляет сеть класса C.

Итак, результатом назначения адресов стали четыре сети класса C, присвоим им соответствующие номера 201.1.1.0, 201.1.2.0, 201.1.3.0 и 201.1.4.0.

Пусть сеть на коммутаторе Switch1 получает номер 201.1.1.0. В этой сети необходимо присвоить 5 адресов: порту маршрутизатора и четырем узлам. Впредь договоримся присваивать портам маршрутизаторов младшие адреса узлов (это не является обязательным условием - просто обычная договоренность для удобства). Т.е. присвоим порту маршрутизатора Router1, который подключен к коммутатору Switch1 адрес 201.1.1.1. Узлам можно присвоить адреса:

- 1: 201.1.1.10
- 2: 201.1.1.11
- 3: 201.1.1.12
- 4: 201.1.1.13

Пусть сеть на коммутаторе Switch2 получает номер 201.1.2.0. И в этой сети необходимо присвоить 5 адресов: порту маршрутизатора и четырем узлам. Присвоим порту маршрутизатора Router2, который подключен к коммутатору Switch2 адрес 201.1.2.1. Тогда узлам можно присвоить адреса:

- 5: 201.1.2.10
- 6: 201.1.2.11
- 7: 201.1.2.12
- 8: 201.1.2.13

Пусть сеть на коммутаторе Switch3 получает номер 201.1.3.0. И в этой сети необходимо присвоить 5 адресов: порту маршрутизатора и четырем узлам. Присвоим порту маршрутизатора Router2, который подключен к коммутатору Switch3 адрес 201.1.3.1. Тогда узлам можно присвоить адреса:

- 9: 201.1.3.10
- 10: 201.1.3.11
- 11: 201.1.3.12
- 12: 201.1.4.13

Глобальной сети между маршрутизаторами Router1 и Router2 присвоим номер сети 201.1.4.0. Порт маршрутизатора Router1 в этой сети получит адрес 201.1.4.1, порт маршрутизатора Router2 в этой сети получит номер 201.1.4.2.

Таким образом, вся составная сеть адресована. Рассмотренный выше набор правил является базовым для присвоения адресов в сети любого масштаба и схемы подключения канального уровня. В случае несоблюдения указанных рекомендаций, работа сетевого уровня может быть нарушена вследствие различий логики протокола IP и логики используемой администратором при назначении адресов.

Например, в случае присвоения хостам лежащим в разных канальных широкополосных сегментах адресов IP лежащих в одной сети – работа сетевого уровня по объединению указанных хостов нарушится. Исправить ситуацию, в этом случае можно двумя путями: назначить адреса в соответствии с рекомендациями или дополнить таблицу маршрутизации на каждом из хостов + таблицу на маршрутизаторе (но это уже из области особых ситуаций).

Из рассмотренного выше примера следуют недостатки классовой техники (RFC791) назначения IP адресов. Сформулируем основные из этих недостатков:

- Небольшие сети (20-100 машин) потребляют идентификатор класса C, который позволяет адресовать до 254 узлов
- Умеренно большие сети (300-500 машин) потребляют идентификатор класса B так как им не хватает размера сетей класса A, а сети класса B позволяют адресовать до 65534 узлов!
- Глобальные сети точка-точка (2 узла) потребляют идентификатор класса C, так как более мелких сетей классовой техника все равно не позволяет
- Потребность в сетях класса B сомнительна – такого числа узлов обычно не используют в одной сети
- Потребность в сетях класса A сомнительна – такого числа узлов никогда не используют в одной сети
- Слишком большая доля адресного пространства используется для гигантских сетей класса A (половина)
- Пусть у компании есть идентификатор сети любого класса и одна сеть из некоторого небольшого количества узлов. При необходимости создать еще несколько сетей, соединив их маршрутизаторами, это не возможно сделать без применения дополнительных номеров сетей, так как одна большая IP сеть некоторого класса не может быть использована как несколько мелких.

Все эти недостатки приводят к одному и тому же – нерациональному использованию адресного пространства, неэффективному расходованию адресов. Все эти недостатки были бы не принципиальны, если бы адресное пространство IP было бы весьма избыточным. Однако, как уже говорилось, это не так, напротив, IP адресов не слишком уж и много (около 4 млрд.). Но при этом нужно вычесть 256 млн. адресов класса D, 256 млн. адресов класса E, 32 млн. адресов в сетях 0.0.0.0 и 127.0.0.1, так что адресов, которые можно присваивать узлам оказывается около 3,5 млрд. Это число не так уж и мало, однако неэкономичное классовое распределение адресов значительно истощило доступное адресное пространство, возникла потребность в более гибком способе адресации, впрочем, таком, который бы можно было использовать, совместно с существующей классовой технологией, не переделывая работающие сети полностью. И такое решение было предложено в RFC917 (октябрь 1984, статус: неизвестно) и в RFC950 (август 1985, статус: стандарт). Эти RFC описывали принципы создания подсетей – деления классовых сетей на более мелкие части с целью более гибкого использования адресного пространства. Указанные действия осуществляются с использованием так называемых масок IP адресов.

Маски IP адресов.

Изучение классового метода деления IP адреса началось с рассмотрения недостатков самого первого метода деления IP адреса – т.н. доклассового метода (RFC 760). Недостаток заключался в нерациональном делении IP адреса на адрес сети и адрес хоста. Вот как выглядел доклассовый метод.

Адрес сети	Адрес хоста	Адрес хоста (продолжение)	Адрес хоста (продолжение)
------------	-------------	---------------------------	---------------------------

Таким образом, на адресацию сетей отводилось 8 бит и их количество составляло $2^8 = 256$

На адресацию хостов отводилось остальные 24 бита, и их количество составляло $2^{24} - 2 = 16\,777\,214$

Введение классов IP адресов решало проблему нерационального деления за счет использования идентификаторов IP адресов. При этом каждому классу соответствовала оригинальная комбинация деления на номер сети и номер хоста. Вроде бы проблема была решена, но так могло показаться лишь в начале - на первый взгляд. И дело было вот в чем: в случае если необходимо разделить адресное пространство сети принадлежащей к одному из классов (А, В или С) хотя бы на две подсети – задача не имеет решения даже с использованием классов IP адресов.

В качестве примера возьмем сеть класса А 11.0.0.0. Предположим, что данная сеть была выдана некоторой компании. Спустя время возникла необходимость разделения сети на две подсети в связи с реорганизацией структуры компании. С одной стороны это вполне реальная для выполнения задача, так как адресное пространство в 16 777 214 узлов, скорее всего ☺, можно поделить между двумя подразделениями одной компании. Например, возьмем 2-й байт в адресе сети и будем его использовать как идентификатор номера подсети – тогда получается

11.1.0.0 – сеть первого подразделения

11.2.0.0 – сеть второго подразделения

Для адресации хостов в каждой из полученных подсетей остается 16 бит, что удовлетворяет количеству хостов компании «с запасом». И задача вроде бы решена – но, в случае если хосты и маршрутизаторы на которых построена сеть компании анализируют IP адрес используя классовый метод – то предложенный вариант деления – работать не будет – т.к. и первая 11.1.0.0 и вторая 11.2.0.0 подсети принадлежат к классу А – следовательно номер сети 11, а 11.1.0.0 и 11.2.0.0 – в этом случае просто хосты.

Решение возникшей проблемы заключалось в способе выделения бит IP адреса отвечающих за номер сети и бит IP адреса отвечающих за номер хоста. Наиболее просто это можно было бы сделать путем введения дополнительно поля в заголовок IP пакета. В этом поле можно было бы указывать число – обозначающее количество бит IP адреса слева направо отвечающих за номер сети. Например, 8 – восемь бит слева на право указывают номер сети, 12 – двенадцать бит слева на право указывают номер сети и т.д. Но формат заголовка IP пакета был уже принят и его изменение вызвало бы массу проблем, в том числе и несовместимость с уже работающим оборудованием сетей (нарушение принципа обратной совместимости). Поэтому идея выделения бит IP адреса указывающих номер сети была воплощена в т.н. **масках** или **масках IP подсети**.

Маска подсети это 32-битное число, записываемое в точечно-десятичной форме, как и IP адрес. Маска устроена следующим образом: сначала последовательность из n единиц, затем – последовательность из [32 минус n] нулей.

Каждому интерфейсу в составной сети присваивается IP адрес и вместе с ним маска подсети. Те биты IP адреса, соответствующие которым биты маски равны «1» означают в IP

адресе номер сети, те же биты IP адреса, соответствующие которым биты маски равны «0» означают в IP адресе номер узла. Это значит, что теперь границу между номером сети и узла в IP адресе можно провести не по границам байтов (как в классовой технике), а между произвольными битами, при этом маску назначает администратор сети, так что выбор размеров сетей отчасти оказывается в руках администратора. Сама по себе техника масок не отменяла технику классов, но была предложена как механизм, дополняющий технику классов, делающий ее более гибкой.

Рассмотрим пример вычисления маски для сети класса А.

Известно, что для сетей класса А номер сети указывается в первой байте, остальные три байта отводятся под адресацию хостов. Обратите внимания, что в данном случае речь об идентификаторе сетей класса А (первый бит первого байта всегда равен 0) не идет, т.к. и это бит входит в номер сети – но всегда в нулевом значении.

Соблюдая правило об устройстве маски (сначала единицы потом нули) и учитывая что первые восемь бит для сети класса А описывают номер сети составим маску в двоичном виде:

```
11111111000000000000000000000000
или
11111111.00000000.00000000.00000000
```

Т.е. в позициях соответствующих номеру сети мы установили двоичные единицы, а в позициях отведенных под хосты – двоичные нули. Форма записи маски, как и форма записи IP может применяться в двоичном, десятичном и шестнадцатеричном виде. Приведем к указанным видам полученную маску

Bin	11111111.00000000.00000000.00000000
Dec	255.0.0.0
Hex	FF:00:00:00

Аналогично, рассмотрим пример вычисления маски для сети класса В.

Известно, что для сетей класса В номер сети указывается в первой и втором байтах, остальные два байта отводятся под адресацию хостов. Обратите внимания, что в данном случае речь об идентификаторе сетей класса В (первые два бита первого байта всегда равны 10) не идет, т.к. и эти биты входят в номер сети – но всегда в значениях 10.

Соблюдая правило об устройстве маски (сначала единицы потом нули) и учитывая что первые шестнадцать бит для сети класса В описывают номер сети составим маску в двоичном виде:

```
11111111111111110000000000000000
или
11111111.11111111.00000000.00000000
```

Т.е. в позициях соответствующих номеру сети мы установили двоичные единицы, а в позициях отведенных под хосты – двоичные нули. Приведем к возможным видам записи полученную маску

Bin	11111111.11111111.00000000.00000000
Dec	255.255.0.0
Hex	FF:FF:00:00

Аналогично, рассмотрим пример вычисления маски для сети класса С.

Известно, что для сетей класса С номер сети указывается в первой, втором и третьем байтах, четвертый байт отводится под адресацию хостов. Обратите внимания, что в данном

случае речь об идентификаторе сетей класса С (первые три бита первого байта всегда равны 110) не идет, т.к. и эти биты входят в номер сети – но всегда в значениях 110.

Соблюдая правило об устройстве маски (сначала единицы потом нули) и учитывая что первые двадцать четыре бита для сети класса С описывают номер сети составим маску в двоичном виде:

```
111111111111111111111111100000000
или
11111111.11111111.11111111.00000000
```

Т.е. в позициях соответствующих номеру сети мы установили двоичные единицы, а в позициях отведенных под хосты – двоичные нули. Приведем к возможным видам записи полученную маску

```
Bin      11111111.11111111.11111111.00000000
Dec      255.255.255.0
Hex      FF:FF:FF:00
```

Далее рассмотрим пример применения маски для выделения номера сети и узла из некоторого IP адреса и маски:

Пусть задан IP адрес: 210.56.78.212. Если техника масок не применяется, то номер сети 210.56.78.0, узел номер 212. Теперь сопоставим с этим адресом маску 255.255.255.224. Что теперь номер сети, а что номер узла? На этот вопрос можно ответить, лишь записав и адрес, и маску в двоичной форме:

```
Адрес: 11010010.00111000.01001110.11010100
Маска: 11111111.11111111.11111111.11100000
```

- обратите внимание – количество единиц 27, что на 3 больше количества единиц маски класса С (в данном случае говорят «длиннее на три бита»)

Дальше мы можем записать номер узла и номер сети. Первые 27 бит записанного IP адреса – номер сети, в которой находится узел, остальные 5 бит – номер самого узла. Запишем номер сети, для этого все биты, отвечающие за номер узла положим равными нулю:

```
Адрес: 11010010.00111000.01001110.110|10100
Маска: 11111111.11111111.11111111.111|00000
Сеть: 11010010.00111000.01001110.110|00000
```

Теперь осталось лишь перевести номер сети в привычную точечно-десятичную форму: 210.56.78.192.

Проанализируем полученную сеть:

Номер сети:

- 210.56.78.192.

Количество узлов в этой сети:

- Судя по количеству нулей в маске, для нумерации узлов остается 5 бит, следовательно в этой сети 2^5 узлов, т.е. 32. Но, адрес, в котором все биты, отвечающие за номер узла равны «0» является номером сети и не может быть назначен узлу, а адрес, в котором все биты, отвечающие за номер узла равны «1» является широковещательным адресом в

данную сеть и так же не может быть назначен узлу. Таким образом, в данной сети 2^5-2 узлов, т.е. 30.

Найдем адрес первого узла в этой сети. В этом случае, биты, отвечающие за номер узла должны принять минимальное значение, но не все быть равны нулям. Т.е. последние пять бит адреса должны принять значение 00001, и тогда адрес первого узла в этой сети выглядит следующим образом:

Сеть: 11010010.00111000.01001110.110|00000
Маска: 11111111.11111111.11111111.111|00000
Узел: 11010010.00111000.01001110.110|00001

Переводим его в точечно-десятичную запись: 210.56.78.193

Найдем теперь самый большой номер узла. В этом случае, биты адреса, отвечающие за номер узла должны быть все кроме последнего равны «1» (иначе получится широковещательный адрес сети).

Сеть: 11010010.00111000.01001110.110|00000
Маска: 11111111.11111111.11111111.111|00000
Узел: 11010010.00111000.01001110.110|11110

Переводим его в точечно-десятичную запись: 210.56.78.222

Наконец найдем широковещательный (broadcast) адрес сети, установив все биты IP адреса, отвечающие за номер равными «1»:

Сеть: 11010010.00111000.01001110.110|00000
Маска: 11111111.11111111.11111111.111|00000
Br-t: 11010010.00111000.01001110.110|11111

Переводим его в точечно-десятичную запись: 210.56.78.224

Итого, имеем узел с IP адресом 210.56.78.212 и маской 255.255.255.224. Этот узел находится в сети 210.56.78.192, всего в сети 30 узлов, первый узел 210.56.78.193, последний узел 210.56.78.222 (наш узел в середине 210.56.78.212), широковещательный адрес сети 210.56.78.223.

Следующий пример.

Условие: дан IP адрес 155.192.68.4 маска 255.255.192.0

Найти: номер сети, количество хостов, первый адрес, последний адрес и широковещательный адрес.

Решение:

1. Выделим номер сети из IP адреса используя приведенную маску

Адрес: 10011011.11000000.01|000100.00000100
Маска: 11111111.11111111.11|000000.00000000

Сеть: 10011011.11000000.01|000000.00000000 = 155.192.64.0

2. Найдем количество хостов в сети

Т.к. длина маски 18 бит, то на хосты отводится $32 - 18 = 14$ бит. Учитывая исключения на широковещательный и адрес сети, рассчитываем количество хостов

$$2^{14} - 2 = 16382$$

3. Найдем первый адрес сети

Сеть: 10011011.11000000.01|000000.00000000

1-й: 10011011.11000000.01|000000.00000001 = 155.192.64.1

4. Найдем последний адрес сети

Сеть: 10011011.11000000.01|000000.00000000

16382-й: 10011011.11000000.01|111111.11111110 = 155.192.127.254

5. Найдем широковещательный адрес

Сеть: 10011011.11000000.01|000000.00000000

Br-t: 10011011.11000000.01|111111.11111111 = 155.192.127.255

Ответ:

Адрес сети 155.192.64.0 255.255.192.0

Количество хостов 16382

1-й 155.192.64.1

16382-й 155.192.127.254

Широковещательный 155.192.127.255

И еще один пример - для закрепления. Дальше будет практическое задание.

Условие: дан IP адрес 11.0.25.165 маска 255.224.0.0

Найти: номер сети, количество хостов, первый адрес, последний адрес и широковещательный адрес.

Решение:

1. Выделим номер сети из IP адреса используя приведенную маску

Адрес: 00001011.000|00000.00011001.10100101

Маска: 11111111.111|00000.00000000.00000000

Сеть: 00001011.000|00000.00000000.00000000 = 11.0.0.0

2. Найдем количество хостов в сети

Т.к. длина маски 11 бит, то на хосты отводится $32 - 11 = 21$ бит. Учитывая исключения на широковещательный и адрес сети, рассчитываем количество хостов

$$2^{21} - 2 = 2\,097\,150$$

3. Найдем первый адрес сети

Сеть: 00001011.000|00000.00000000.00000000

1-й: 00001011.000|00000.00000000.00000001 = 11.0.0.1

4. Найдем последний адрес сети

Сеть: 00001011.000|00000.00000000.00000000

2 097150-й: 00001011.000|11111.11111111.11111110 = 11.31.255.254

5. Найдем широковещательный адрес

Сеть: 00001011.000|00000.00000000.00000000

Br-t: 00001011.000|11111.11111111.11111111 = 11.31.255.255

Ответ:

Адрес сети 11.0.0.1 255.224.0.0

Количество хостов 2 097 150

1-й	11.0.0.1
2 097150-й	11.31.255.254
Широковещательный	11.31.255.255

Задание.

Найти номер сети, количество хостов, первый адрес, последний адрес и широковещательный адрес для следующих IP адресов

192.241.25.69	255.224.0.0
221.156.125.5	255.255.192.0
109.135.221.251	255.255.255.248
34.0.25.1	255.255.252.0
115.128.0.75	255.128.0.0

Деление сетей с помощью масок.

Рассмотрим практическое использование изученного метода, т.е. деление классовых сетей на части с помощью масок.

Начнем с деления на части сети класса C. Изначально сеть класса C может содержать до 254 узлов, и узлы будут использовать маску 255.255.255.0. Это значит, что первые 24 бита адреса – номер сети, а последние 8 бит – номер узла.

Пусть мы хотим разбить сеть на несколько частей, пусть части будут меньше, чем 254 узла, но этих частей (подсетей или сетей) будет более одной. Как это сделать? Получив номер сети класса C, мы исходим из того, что назначение адресов узлов в сети – наше право, а все маршрутизаторы всей составной сети все пакеты на весь наш диапазон адресов в конечном счете переправят нашему маршрутизатору, который передаст их нашим узлам. Иными словами последние 8 бит адреса находятся целиком в нашем распоряжении и мы можем поступать с ними по своему усмотрению: можем использовать все 8 бит для нумерации узлов, а можем некоторое количество из этих 8 бит использовать как номер сети, или как еще говорят «подсети», так как полученные таким образом сети будут частями исходной сети класса C.

Пусть мы позаимствуем в пользу номера подсети из 8 имеющихся в нашем распоряжении бит только 1. Как показать это маской? Очевидно, нужно выбрать маску таковой, чтобы первые 24+1 бита были в адресах номером сети, т.е. соответствующие биты в маске должны быть равны «1», а остальные биты в маске должны равны «0».

Исходная маска: 11111111.11111111.11111111.00000000 (255.255.255.0)

Новая маска: 11111111.11111111.11111111.10000000 (255.255.255.128)

Сколько бит у нас есть в распоряжении для нумерации подсетей, являющихся частями нашего диапазона класса C? Очевидно лишь один бит. Сколько подсетей можно пронумеровать, используя один бит? Очевидно, лишь 2 подсети ($2^1 = 2$). Сколько узлов будет в каждой подсети? Узлы в подсетях нумеруют 7 бит, следовательно, узлов будет $2^7 - 2 = 126$. Запишем номера этих подсетей. (Номер сети принимаем равным 212.1.2.0).

Так как первые три байта не меняются, будем записывать их в десятичной записи, а последний байт – в двоичной форме. Запишем номер сети и маску, выделяющую 1 дополнительный бит для нумерации подсетей.

Номер исходной сети: 212.1.2.00000000

Маска: 255.255.255.10000000

Из этой записи следует, что 1-ый бит четвертого байта адреса является частью номера сети (или номером подсети), а семь остальных нумеруют узел. Можем переписать тогда в виде:

Наш диапазон адресов: 212.1.2.x|ууууууу
Маска: 255.255.255.1|0000000

Где x – биты, нумерующие подсети, а у – биты, нумерующие узлы.

Рассматриваем первую подсеть. Последовательность «x» (в нашем случае, пока один «x») принимает первое возможное значение – «0», а последовательность «у» должна быть положена равной «0» в том случае, если мы хотим получить номер первой подсети:

Номер первой подсети: 212.1.2.0|0000000
Маска: 255.255.255.1|0000000

В точечно-десятичной форме первая подсеть будет записана как 212.1.2.0, маска 255.255.255.128. Сколько узлов в этой подсети? Очевидно, 126 ($2^7 - 2 = 126$, 7 – количество бит на адресацию хостов).

Определим номер первого узла этой подсети. Для этого необходимо приписать последовательности «у» минимальное возможное значение (но не все нули, иначе получится не номер первого узла, а номер сети)

Адрес первого узла: 212.1.2.0|0000001
Маска: 255.255.255.1|0000000

В точечно-десятичной форме первый узел будет иметь адрес 212.1.2.1, маска 255.255.255.128.

Определим номер последнего узла этой подсети. Необходимо приписать последовательности «у» максимально возможное значение (но не все единицы, иначе получится не номер последнего узла, а широковещательный адрес)

212.1.2.0|1111110
Адрес первого узла: 255.255.255.1|0000000
Маска:

В точечно-десятичной форме последний узел будет иметь адрес 212.1.2.126, маска 255.255.255.128.

Найдем широковещательный адрес этой сети: для этого последовательность «у» должна принять значение из всех «1»:

Широковещательный адрес: 212.1.2.0|1111111
Маска: 255.255.255.1|0000000

В точечно-десятичной форме широковещательный адрес будет 212.1.2.127.

Итог по первой подсети:

номер подсети	212.1.2.0
маска	255.255.255.128
первый узел	212.1.2.1
последний узел	212.1.2.126,
диапазон адресов	212.1.2.1 – 212.1.2.126
всего узлов	126
широковещательный	212.1.2.127

Рассмотрим вторую подсеть, для этого первому биту четвертого байта приписываем значение «1», остальное делаем аналогично:

Номер второй подсети: 212.1.2.1|0000000
Маска: 255.255.255.1|0000000

В точечно-десятичной форме первая подсеть будет записана как 212.1.2.128, маска 255.255.255.128.

Определим номер первого узла подсети. Для этого необходимо приписать последовательности «у» минимальное возможное значение (но не все нули, иначе получится не номер первого узла, а номер сети)

Адрес первого узла: 212.1.2.1|0000001
Маска: 255.255.255.1|0000000

В точечно-десятичной форме первый узел будет иметь адрес 212.1.2.129, маска 255.255.255.128.

Определим номер последнего узла подсети. Необходимо приписать последовательности «у» максимально возможное значение (но не все единицы, иначе получится не номер последнего узла, а широковещательный адрес)

Адрес последнего узла: 212.1.2.1|1111110
Маска: 255.255.255.1|0000000

В точечно-десятичной форме последний узел будет иметь адрес 212.1.2.254, маска 255.255.255.128.

Определим широковещательный адрес этой сети: для этого последовательность «у» должна принять значение из всех «1»:

Широковещательный адрес: 212.1.2.1|1111111
Маска: 255.255.255.1|0000000

В точечно-десятичной форме широковещательный адрес будет 212.1.2.255.

Итог по второй подсети:

номер подсети	212.1.2.128
маска	255.255.255.128
первый узел	212.1.2.129
последний узел	212.1.2.254
диапазон адресов	212.1.2.129 – 212.1.2.154
узлов	126
широковещательный	212.1.2.255

Таким образом, мы разделили сеть 212.1.2.0 на две равные части, каждая из которых теперь может называться самостоятельной сетью.

Можно ли разделить исходную сеть не на 2, а на 4 части? Да, для этого необходимо позаимствовать из последнего байта не 1 а 2 бита в пользу номера подсети.

Получим:

Наш диапазон адресов:	212.1.2.xхуууууу
Маска:	255.255.255.11000000 (255.255.255.192)

Номер первой подсети:	212.1.2.00000000 (212.1.2.0) (62 узла)
Адрес первого узла:	212.1.2.00000001 (212.1.2.1)
Адрес последнего узла:	212.1.2.00111110 (212.1.2.62)
Широковещание:	212.1.2.00111111 (212.1.2.63)

Номер второй подсети:	212.1.2.01000000 (212.1.2.64) (62 узла)
Адрес первого узла:	212.1.2.01000001 (212.1.2.65)
Адрес последнего узла:	212.1.2.01111110 (212.1.2.126)
Широковещание:	212.1.2.01111111 (212.1.2.127)
Номер третьей подсети:	212.1.2.10000000 (212.1.2.128) (62 узла)
Адрес первого узла:	212.1.2.10000001 (212.1.2.129)
Адрес последнего узла:	212.1.2.10111110 (212.1.2.190)
Широковещание:	212.1.2.10111111 (212.1.2.191)
Номер четвертой подсети:	212.1.2.11000000 (212.1.2.192) (62 узла)
Адрес первого узла:	212.1.2.11000001 (212.1.2.193)
Адрес последнего узла:	212.1.2.11111110 (212.1.2.254)
Широковещание:	212.1.2.11111111 (212.1.2.255)

Разделим исходную сеть на большее число частей, позаимствуем из последнего байта 3 бита в пользу номера подсети. Получим 8 подсетей по 30 узлов в каждой.

Наш диапазон адресов:	212.1.2.xxxxxxxx
Маска:	255.255.255.11100000 (255.255.255.224)
Номер первой подсети:	212.1.2.00000000 (212.1.2.0) (30 узла)
Адрес первого узла:	212.1.2.00000001 (212.1.2.1)
Адрес последнего узла:	212.1.2.00011110 (212.1.2.30)
Широковещание:	212.1.2.00011111 (212.1.2.31)
Номер второй подсети:	212.1.2.00100000 (212.1.2.32) (30 узлов)
Адрес первого узла:	212.1.2.00100001 (212.1.2.33)
Адрес последнего узла:	212.1.2.00111110 (212.1.2.62)
Широковещание:	212.1.2.00111111 (212.1.2.63)
Номер третьей подсети:	212.1.2.01000000 (212.1.2.64) (30 узлов)
Адрес первого узла:	212.1.2.01000001 (212.1.2.65)
Адрес последнего узла:	212.1.2.01011110 (212.1.2.94)
Широковещание:	212.1.2.01011111 (212.1.2.95)
Номер четвертой подсети:	212.1.2.01100000 (212.1.2.96) (30 узлов)
Адрес первого узла:	212.1.2.01100001 (212.1.2.97)
Адрес последнего узла:	212.1.2.01111110 (212.1.2.126)
Широковещание:	212.1.2.01111111 (212.1.2.127)
Номер пятой подсети:	212.1.2.10000000 (212.1.2.128) (30 узлов)
Адрес первого узла:	212.1.2.10000001 (212.1.2.129)
Адрес последнего узла:	212.1.2.10011110 (212.1.2.158)
Широковещание:	212.1.2.10011111 (212.1.2.159)
Номер шестой подсети:	212.1.2.10100000 (212.1.2.160) (30 узлов)
Адрес первого узла:	212.1.2.10100001 (212.1.2.161)
Адрес последнего узла:	212.1.2.10111110 (212.1.2.190)
Широковещание:	212.1.2.10111111 (212.1.2.191)

Номер седьмой подсети:	212.1.2.11000000 (212.1.2.192) (30 узлов)
Адрес первого узла:	212.1.2.11000001 (212.1.2.193)
Адрес последнего узла:	212.1.2.11011110 (212.1.2.222)
Широковещание:	212.1.2.11011111 (212.1.2.223)

Номер восьмой подсети:	212.1.2.11100000 (212.1.2.224) (30 узлов)
Адрес первого узла:	212.1.2.11100001 (212.1.2.225)
Адрес последнего узла:	212.1.2.11111110 (212.1.2.254)
Широковещание:	212.1.2.11111111 (212.1.2.255)

Для деление на 16 подсетей позаимствуем 4 бита на номер подсети, получим 16 подсетей по 14 узлов в каждой.

Наш диапазон адресов:	212.1.2.xxxxxuuu
Маска:	255.255.255.11110000 (255.255.255.240)

Выпишем только номера подсетей сетей:

1	212.1.2.00000000 (212.1.2.0)
2	212.1.2.00010000 (212.1.2.16)
3	212.1.2.00100000 (212.1.2.32)
4	212.1.2.00110000 (212.1.2.48)
5	212.1.2.01000000 (212.1.2.64)
6	212.1.2.01010000 (212.1.2.80)
7	212.1.2.01100000 (212.1.2.96)
8	212.1.2.01110000 (212.1.2.112)
9	212.1.2.10000000 (212.1.2.128)
10	212.1.2.10010000 (212.1.2.144)
11	212.1.2.10100000 (212.1.2.160)
12	212.1.2.10110000 (212.1.2.176)
13	212.1.2.11000000 (212.1.2.192)
14	212.1.2.11010000 (212.1.2.208)
15	212.1.2.11100000 (212.1.2.224)
16	212.1.2.11110000 (212.1.2.240)

Позаимствовав 5 бит на номер подсети, получим 32 подсети по 6 узлов в каждой

Наш диапазон адресов:	212.1.2.xxxxxuuu
Маска:	255.255.255.11110000 (255.255.255.248)

Позаимствовав 6 бит на номер подсети, получим 64 подсети по 2 узла в каждой

Наш диапазон адресов:	212.1.2.xxxxxuuu
Маска:	255.255.255.11111100 (255.255.255.252)

Позаимствовав 7 бит на номер подсети, получим 128 подсетей, но в этих подсетях не сможет расположится ни одного узла: так как для нумерации узла в этом случае применяется 1 бит, то он принимает всего 2 значения «0» и «1», при последнем бите, равном нулю результат будет номером сети, а при последнем бите, равном единице результат будет широковещанием в данную сеть, видно, что корректных номеров узлов не остается.

Наш диапазон адресов:	212.1.2.xxxxxuuu
Маска:	255.255.255.11111110 (255.255.255.254)

Таким образом маска 255.255.255.254 не используется и разбивать сеть класса C на 128 подсетей не имеет смысла.

Как следует из описанных выше примеров количество подсетей на которые делится сеть всегда равно натуральной степени 2, а число узлов - натуральная степень двойки минус 2.

Далее рассмотрим по одному примеру деления сетей классов В и А.

Разделим сеть класса В 150.150.0.0, таким образом, чтобы получилось не менее 40 подсетей. Главным вопросом, на которых необходимо ответить перед описанием подсетей является вопрос:

- сколько бит необходимо, чтобы они приехали по меньшей мере 40 разных значений ? Или, 2 в какой степени больше либо равно 40 ?

Ответ: минимальное значение показателя степени равно 6.

Т.е. минимум 6 бит необходимо для нумерации 40 подсетей. В сетях класса В изначально на номер сети остается 14 бит, на номер узла – 16 бит. Из этих 16-и бит, предназначенных для нумерации узлов, позаимствуем в пользу номера подсети 6 бит, тогда на номер узла останется еще 10 бит. Таким образом получится 64 подсети, в каждой из них окажется по 1022 узла. И хотя по условию требуется 40 подсетей, выделение 6-ти бит на нумерацию – автоматически «вырезает» $2^6 = 64$ адреса.

Исходный диапазон адресов:	150.150.yyyyyyyy.yyyyyyy
Маска:	255.255.00000000.00000000 (255.255.0.0)

Проводим границу между номером сети и номером узла следующим образом:

Исходный диапазон адресов:	150.150.xxxxxx yy.yyyyyyy
Маска:	255.255.111111 00.00000000 (255.255.252.0)

Для примера рассмотрим некоторую подсеть с xxxxxx=100110

Наша подсеть:	150.150.100110 yy.yyyyyyy
Маска:	255.255.111111 00.00000000 (255.255.252.0)
Номер нашей подсети:	150.150.100110 00.00000000 (150.150.144.0)
Адрес первого узла:	150.150.100110 00.00000001 (150.150.144.1)
Адрес последнего узла:	150.150.100110 11.11111110 (150.150.155.254)
Широковещание:	150.150.100110 11.11111111 (150.150.155.255)

Обратите внимание, номер узла, в данном случае, состоит из двух последних бит третьего байта и всех бит четвертого байта. Точка между байтами не влияет на возможность подобного явления. Не нужно забывать, что точка между байтами применяется лишь для удобства записи – не более. IP адрес по сути – это 32 бита слева на право и в подобных ситуациях (попадание разделяющей точки в номер хоста или подсети) о существовании точки можно забыть до момента формирования ответа в десятично-точечной форме.

Теперь рассмотрим пример с сетью класса А: разделим сеть 17.0.0.0 таким образом, чтобы получились сети размером не менее 400 000 узлов.

В начале, что установим какое количество подсетей получится в результате и какой маской эти сети будут описываться. Так как известно, сколько необходимо иметь узлов в одной сети, оценим, сколько бит необходимо оставить на номер узла, а остальные позаимствовать в пользу номера сети. Учитывая, что 19 бит позволяют перенумеровать около 500 тыс. узлов ($2^{19} = 524\ 288$), принимаем, что на номер узла достаточно оставить 19 бит, тогда остальные 13 бит IP адреса будут являться номером сети. Первый из них фиксирован (сеть класса А), семь следующих нумеруют саму сеть класса А, следовательно оставшиеся 5 бит позволят нам нумеровать подсети.

Исходный диапазон адресов: 17.yyyyyyyyy.yyyyyyyyy.yyyyyyy
Маска: 255.00000000.00000000.00000000 (255.0.0.0)

Проводим границу между номером сети и номером узла следующим образом:

Исходный диапазон адресов: 17.xxxxxxyyy.yyyyyyyyy.yyyyyyy
Маска: 255.11111000.00000000.00000000 (255.248.0.0)

Для примера рассмотрим некоторую подсеть с xxxxx=01010

Наша подсеть: 17.01010yy.yyyyyyy
Маска: 255.1111100.00000000 (255.255.252.0)
Номер нашей подсети: 17.01010000.00000000.00000000 (17.80.0.0)
Адрес первого узла: 17.01010 000.00000000.00000001 (17.80.0.1)
Адрес последнего узла: 17.01010111.11111111.11111110 (17.87.255.254)
Широковещание: 17.01010111.11111111.11111111 (17.87.255.255)

Выводы

При рассмотрении модели составной сети Интернет, построенной только на классах установлено, что в этом случае нерационально расходуется дефицитное адресное пространство, что приводит к его быстрому истощению.

Для каждой своей сети, подключаемой к Интернет, некоторой компании необходимо получать классовый идентификатор, даже если в распоряжении компании есть достаточное количество IP адресов: пусть у компании имеется идентификатор класса C (254 узла), а в ее сети пока 20 компьютеров. Пусть компания хочет разделить свою сеть на две сети, во второй сети будет еще 10 компьютеров. Итого, у компании 30 компьютеров и идентификатор класса C, содержащий в себе 254 адреса узла, но на его базе можно построить только одну сеть, а компании нужно две сети: приходится приобретать еще один идентификатор класса C, что неудобно и приводит к неоправданному расходованию адресного пространства Интернет.

Как отмечалось ранее, если бы адресов было бы избыточно много, то проблема неэффективного использования адресного пространства отсутствовала, впрочем, неудобства, связанные с обращением к сетевому координационному центру за новым номером сети все равно бы оставались. Таким образом, классовая схема деления IP адресов была приемлема лишь на самых ранних стадиях развития Интернет. Она нуждалась в модификации, но! Работавшую систему сложно переделывать кардинально, никого ведь не устроит революционное изменение принципов работы большой системы! Нужно было разработать такой эволюционный путь развития классового подхода, который позволил бы преодолевать описанные выше проблемы, при этом не приводить к необходимости полной переделки всего, что уже работает. В этом случае внедрения новой технологии, как и в большинстве подобных использовался принцип Обратной совместимости – т.е. новые правила не должны были отрицать старых, допускалось лишь улучшение существующей стратегии работы с IP адресами.

Именно таким подходом стали маски. Действительно, компания, описанная выше, могла бы разбить свою сеть на две подсети, просто сконфигурировав СВОЙ маршрутизатор и СВОИ станции. При этом магистраль Интернет не должна знать о том, что сделала компания в своей сети, маршрутизаторы магистрали Интернет перенаправляют все пакеты в сеть класса C на граничный маршрутизатор компании, который уже перераспределяет трафик между своими подсетями. Т.е. применение масок на периферии сети (у клиентов) не затрагивает работу магистрали, которая, вообще говоря, в такой модели вообще может не поддерживать масок, считая, что все сети классовые! И в этом сила подхода, связанного с введением масок. Эта техника помогла компаниям, подключенным к Интернет, более гибко работать со своим адресным пространством, не затрагивая при этом магистрали сети.

Технику масок мог внедрить в СВОЕЙ сети тот, кто имел в этом потребность и соответствующее программное и аппаратное обеспечение, поддержки со стороны магистрали Интернета не требовалось. С точки зрения магистрали Интернета, все сети оставались классовыми.

Все вышесказанное относится к случаю, когда происходит подключение в Интернет. Если некто строит свою сеть на базе IP и не желает подключать ее к Интернету, то у него нет проблем с получением адресов в координационном центре (он сам себе координационный центр ☺), так же нет проблем с нехваткой адресов. Так что для не подключенных к Интернет сетей даже крупных компаний техника масок избыточна и не нужна – масштабируемости техники классов вполне хватает даже на крупные корпоративные сети, ее не хватает на ГИГАНТСКИЕ сети!

Практическое задание:

Задание 1

Часть 1. Выделить номер сети и номер узла из IP адреса с указанной маской.

192.56.78.99	255.255.255.192
132.212.212.212	255.255.254.0
5.199.2.3	255.192.0.0
5.199.2.4	255.255.255.248
208.1.2.219	255.255.255.240
149.149.149.149	255.255.224.0
177.177.177.177	255.255.255.128
222.222.222.213	255.255.255.224
165.165.165.165	255.255.252.0
1.2.3.4	255.128.0.0

Часть 2. Какие маски являются формально допустимыми:

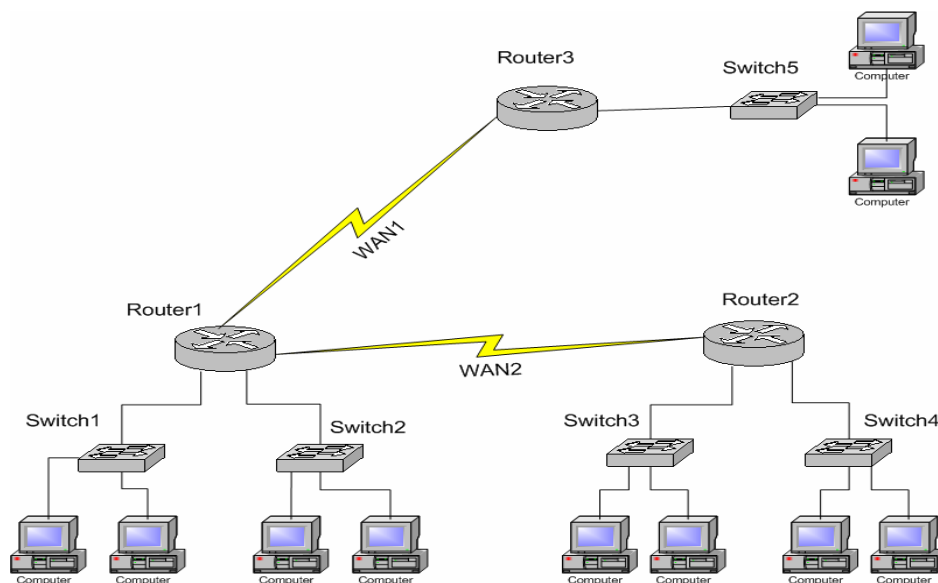
255.0.255.0
255.255.255.252
255.255.0.255
255.255.128.0
255.144.0.0
255.255.191.0
255.254.255.0
255.224.0.0
255.255.253.0
255.255.255.240

192.0.0.0
255.255.188.0
255.254.0.0
255.192.128.0
255.255.255.0
255.255.192.0
255.255.255.248

Часть 3. Разбить сеть класса В 150.150.0.0 на 7 частей, найти номера всех сетей, первых узлов и последних узлов, широковещательных адресов каждой сети.

Задание 2

Пусть компания имеет сеть, изображенную на рисунке ниже. В сети на коммутаторе Switch1 – 20 компьютеров, в сети на коммутаторе Switch2 – 28 компьютеров, в сети на коммутаторе Switch3 – 10 компьютеров, в сети на коммутаторе Switch4 – 26 компьютеров, в сети на коммутаторе Switch5 – 15 компьютеров. Компания получила идентификатор класса C 221.9.8.0. Требуется разделить эту сеть на необходимое количество подсетей, рассчитать маску подсети, номера всех сетей, присвоить адреса всем узлам сети и портам маршрутизаторов, рассчитать для каждой сети широковещательный адрес.

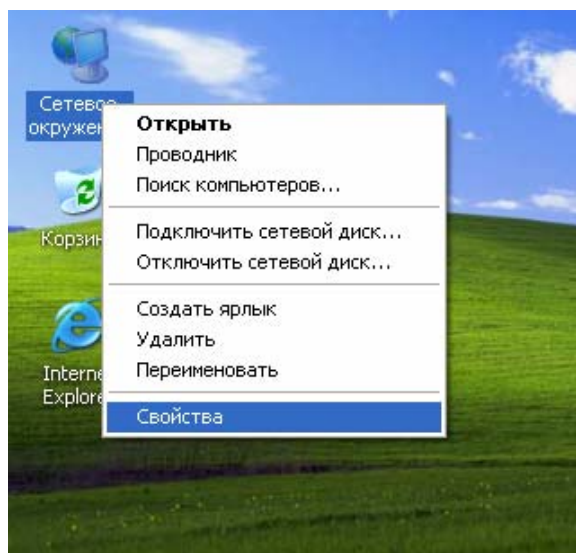


Настройка IP адреса и маски в MS Windows

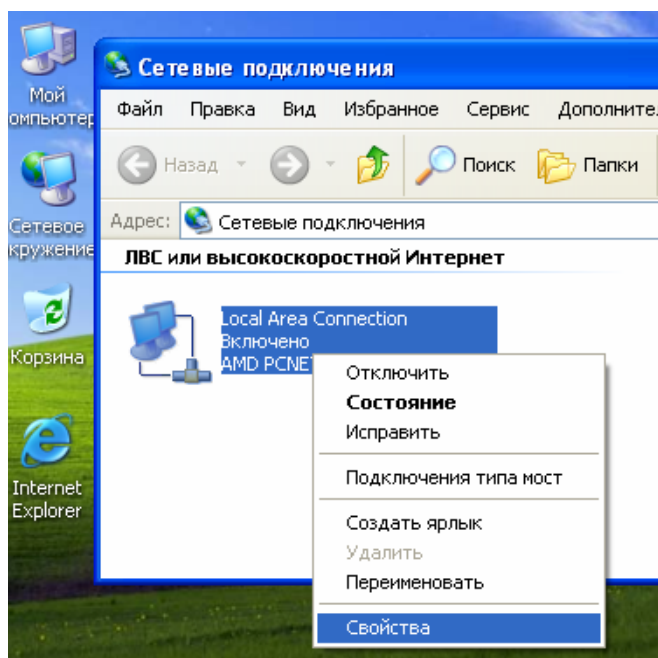
Все современные ОС поддерживают работу с сетью, при этом обязательной является поддержка стека TCP/IP. Рассмотрим способы настройки IP протокола на примере ОС Windows компании Microsoft.

Каждому сетевому адаптеру в ОС Windows соответствует «подключение к локальной сети» в папке «Сетевые подключения». **Свойства** этого подключения позволяют конфигурировать данный адаптер для работы в сети. В этом разделе могут быть установлены сетевые протоколы и службы, которые будет поддерживать данный адаптер. По умолчанию стек TCP/IP устанавливается для каждого сетевого адаптера в процессе инсталляции ОС. Как и большинство настроек Windows, конфигурирование IP интерфейса может осуществляться несколькими способами, рассмотрим один из них.

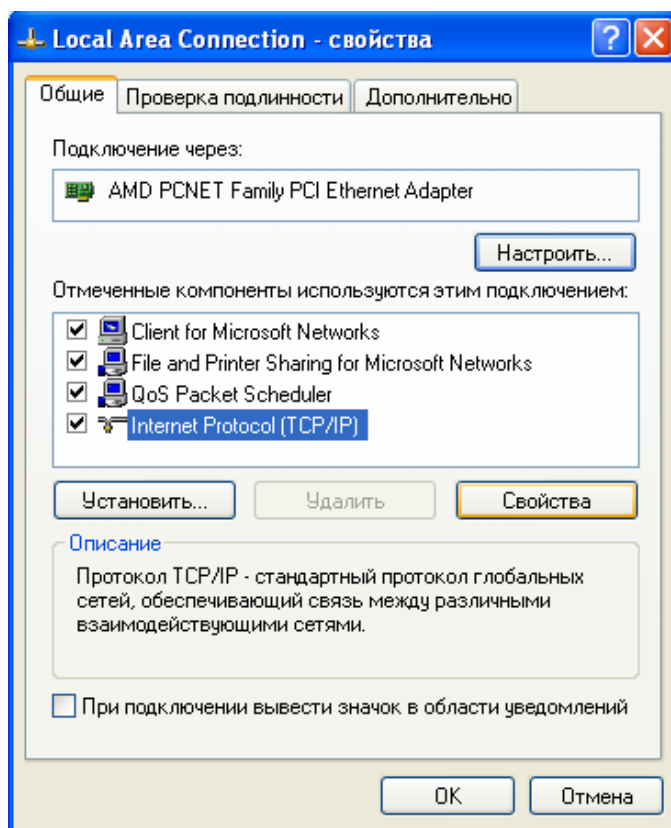
- вызываем свойства «сетевого окружения»



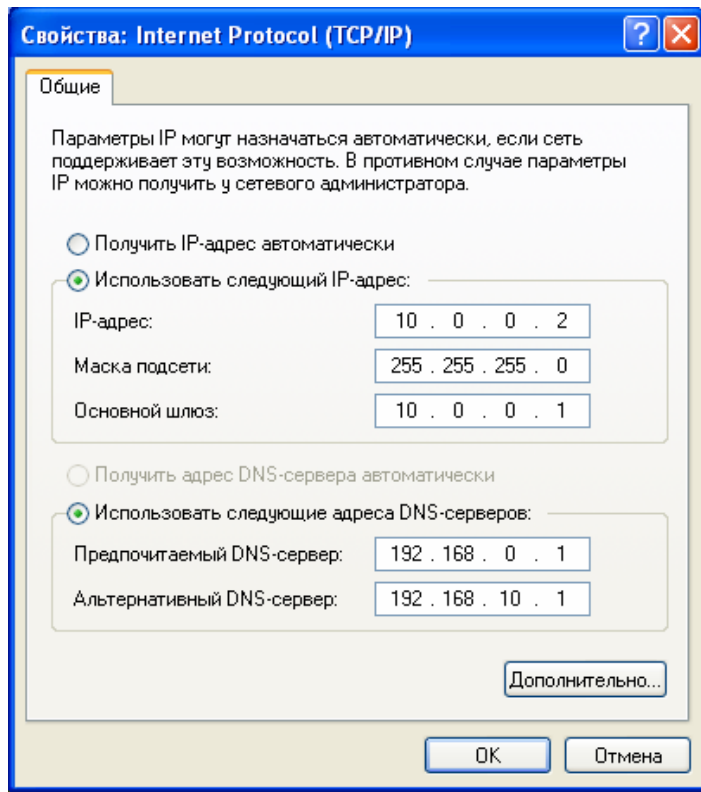
- в появившемся окне вызываем свойства « сетевого подключения», для которого будет осуществляться конфигурация интерфейса



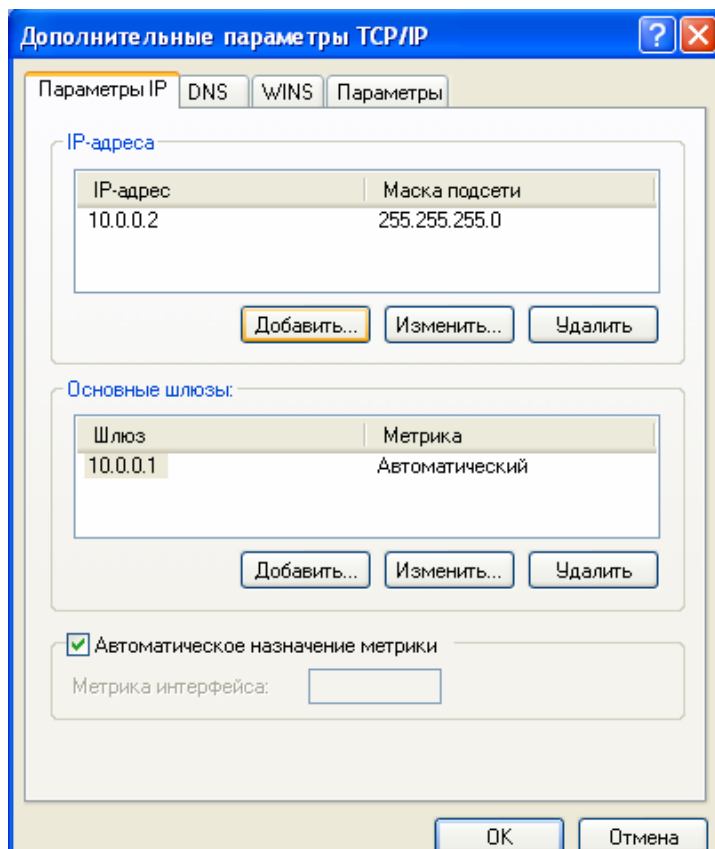
- в свойствах подключения выбираем Internet Protocol (TCP/IP), и нажимаем кнопку «Свойства»



- в появившемся окне описываем параметры IP протокола, а именно «IP-адрес», «Маска подсети» и «Основной шлюз». Так же в настройках присутствует описание DNS сервера – но об этом – позже.



- если конфигурация окончена, нажимаем кнопку «ОК». В случае если физическому интерфейсу (т.е. сетевой карте) необходимо назначить более одного IP адреса, нажимаем кнопку «Дополнительно».



- используем кнопку «Добавить ...» для назначения интерфейсу еще одного IP адреса. Подробнее о назначении интерфейсу нескольких IP адресов далее в курсе.

После конфигурации IP протокола просмотреть текущие настройки можно из консоли (командная строка), используя команду `ipconfig`

```
C:\WINDOWS\System32\cmd.exe

C:\>ipconfig

Настройка протокола IP для Windows

Local Area Connection - Ethernet адаптер:

    DNS-суффикс этого подключения . . . :
    IP-адрес . . . . . : 10.0.0.2
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз . . . . . : 10.0.0.1

C:\>_
```

Более детально, информация может быть получена при использовании ключа /all

```
C:\WINDOWS\System32\cmd.exe

C:\>ipconfig /all

Настройка протокола IP для Windows

    Имя компьютера . . . . . : step
    Основной DNS-суффикс . . . . . :
    Тип узла . . . . . : неизвестный
    IP-маршрутизация включена . . . . . : нет
    WINS-прокси включен . . . . . : нет

Local Area Connection - Ethernet адаптер:

    DNS-суффикс этого подключения . . . :
    Описание . . . . . : AMD PCNET Family PCI Ethernet Adapter
    Физический адрес . . . . . : 00-0C-29-7F-FE-F5
    Dhcp включен . . . . . : нет
    IP-адрес . . . . . : 10.0.0.2
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз . . . . . :
    DNS-серверы . . . . . : 192.168.0.1
    . . . . . : 192.168.10.1

C:\>
```

Так же для конфигурации сетевых интерфейсов в консоли применяется команда netsh.

```
C:\WINDOWS\System32\cmd.exe - netsh

C:\>netsh
netsh>

Применимы следующие команды:

Команды в этом контексте:
.. - Переход на один контекстный уровень вверх.
? - Отображение списка команд.
abort - Отмена изменений, сделанных в автономном режиме.
add - Добавление элемента конфигурации в список элементов.
alias - Добавление псевдонима.
bridge - Изменения в контексте 'netsh bridge'.
bye - Выход из программы.
commit - Применение изменений, сделанных в автономном режиме.
delete - Удаление элемента конфигурации из списка элементов.
diag - Изменения в контексте 'netsh diag'.
dump - Отображение сценария конфигурации.
exec - Запуск файла сценария.
exit - Выход из программы.
help - Отображение списка команд.
interface - Изменения в контексте 'netsh interface'.
offline - Переход в автономный режим.
online - Переход в оперативный режим.
popd - Получение контекста из стека.
pushd - Помещение текущего контекста в стек.
quit - Выход из программы.
ras - Изменения в контексте 'netsh ras'.
routing - Изменения в контексте 'netsh routing'.
set - Обновление параметров конфигурации.
show - Отображение информации.
unalias - Удаление псевдонима.

Доступны следующие дочерние контексты:
bridge diag interface ras routing

Чтобы получить справку по команде, введите эту команду,
затем пробел и "?"

netsh>
```

Назначение нового IP адреса при помощи команды netsh

```
C:\WINDOWS\System32\cmd.exe

C:\>netsh interface ip set address local static 10.0.0.2 255.255.255.0 10.0.0.1 1
OK.

C:\>ipconfig

Настройка протокола IP для Windows

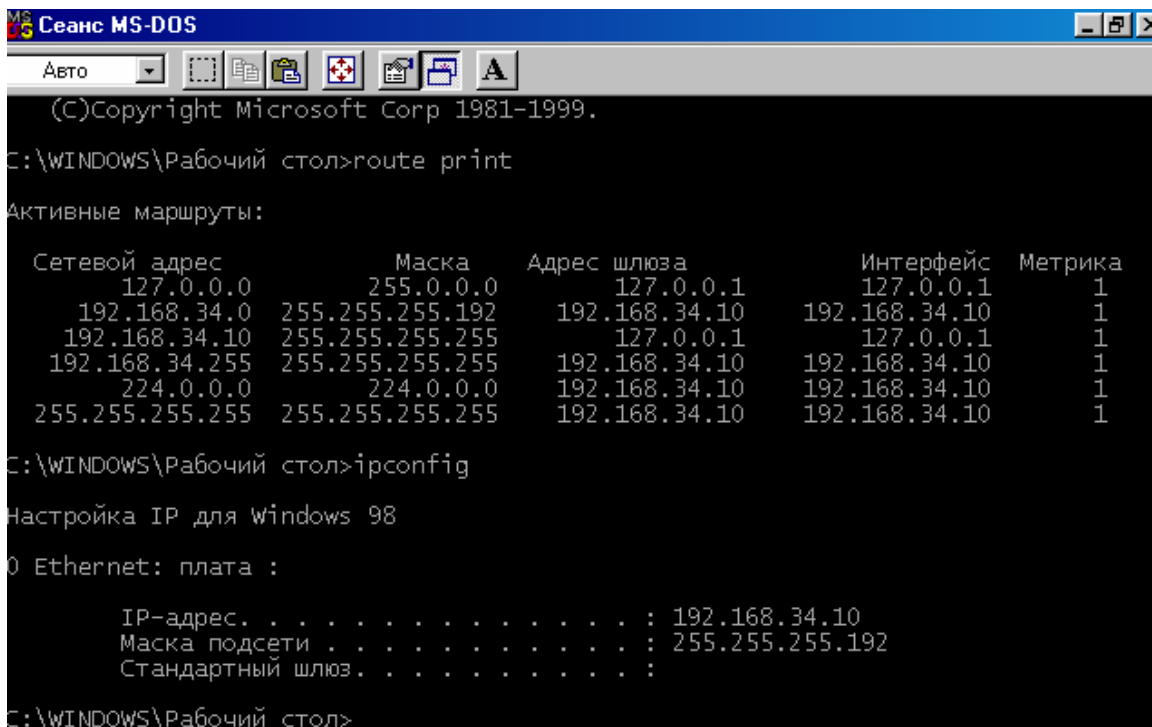
Local Area Connection - Ethernet адаптер:

    DNS-суффикс этого подключения . . . :
    IP-адрес . . . . . : 10.0.0.2
    Маска подсети . . . . . : 255.255.255.0
    Основной шлюз . . . . . : 10.0.0.1

C:\>
```

Старые версии Windows поддерживают только RFC 791, т.е. интерфейсы не нуждаются в настройке маски. И даже при желании, маска узлам, работающим под управлением таких ОС, не может быть назначена – узлы получают адреса и считают, что с ними в одной сети находятся те узлы, которые предполагаются техникой классов.

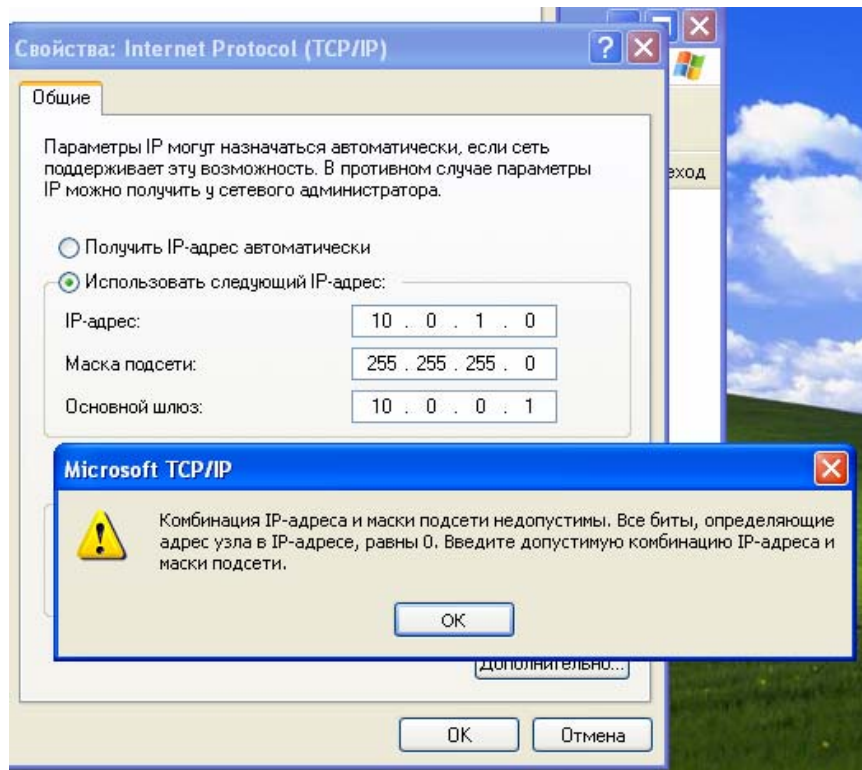
Версии, начиная с Windows 95 поддерживает и RFC791 и RFC 950. Т.е. если нет необходимости пользоваться техникой масок, а использовать только технику классов, то в таком случае просто необходимо присваивать узлам маски, соответствующие технике классов: 255.0.0.0, 255.255.0.0, 255.255.255.0. В случае если необходимо использовать маски заданной битовой длины, в значения маски IP адреса записываются соответствующие числа. Например 255.224.0.0, 255.255.128.0 и т.д. Все современные ОС так же поддерживают и «понимают» такую форму записи масок IP адресов.



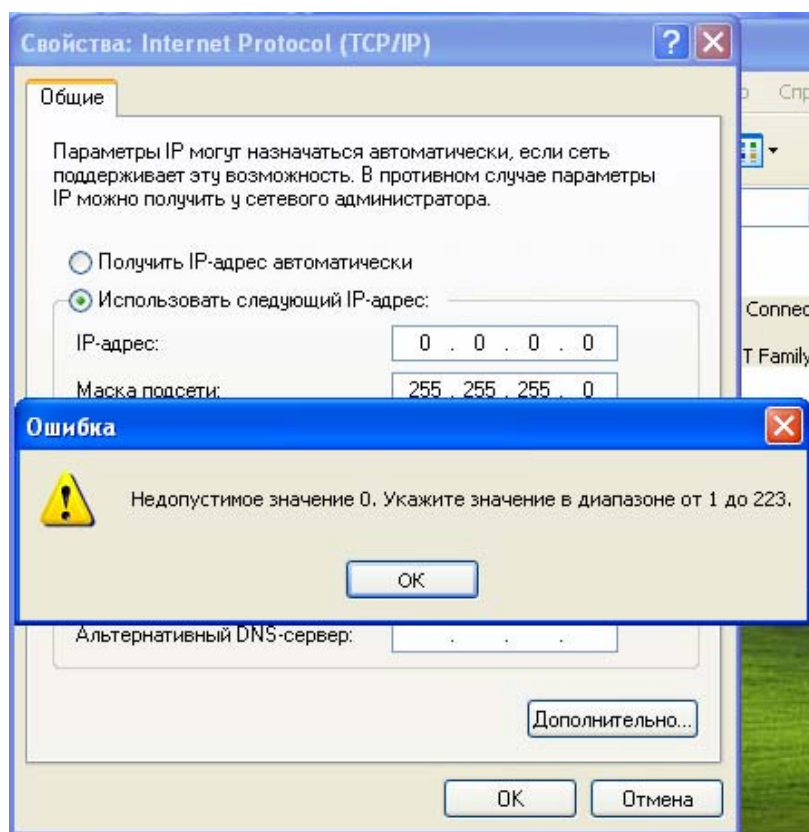
```
Сеанс MS-DOS
(C) Copyright Microsoft Corp 1981-1999.
C:\WINDOWS\Рабочий стол>route print
Активные маршруты:
Сетевой адрес      Маска      Адрес шлюза      Интерфейс      Метрика
127.0.0.0          255.0.0.0   127.0.0.1        127.0.0.1      1
192.168.34.0      255.255.255.192  192.168.34.10   192.168.34.10  1
192.168.34.10     255.255.255.255  127.0.0.1        127.0.0.1      1
192.168.34.255    255.255.255.255  192.168.34.10   192.168.34.10  1
224.0.0.0         224.0.0.0    192.168.34.10   192.168.34.10  1
255.255.255.255   255.255.255.255  192.168.34.10   192.168.34.10  1
C:\WINDOWS\Рабочий стол>ipconfig
Настройка IP для Windows 98
0 Ethernet: плата :
    IP-адрес. . . . . : 192.168.34.10
    Маска подсети . . . . . : 255.255.255.192
    Стандартный шлюз. . . . . :
```

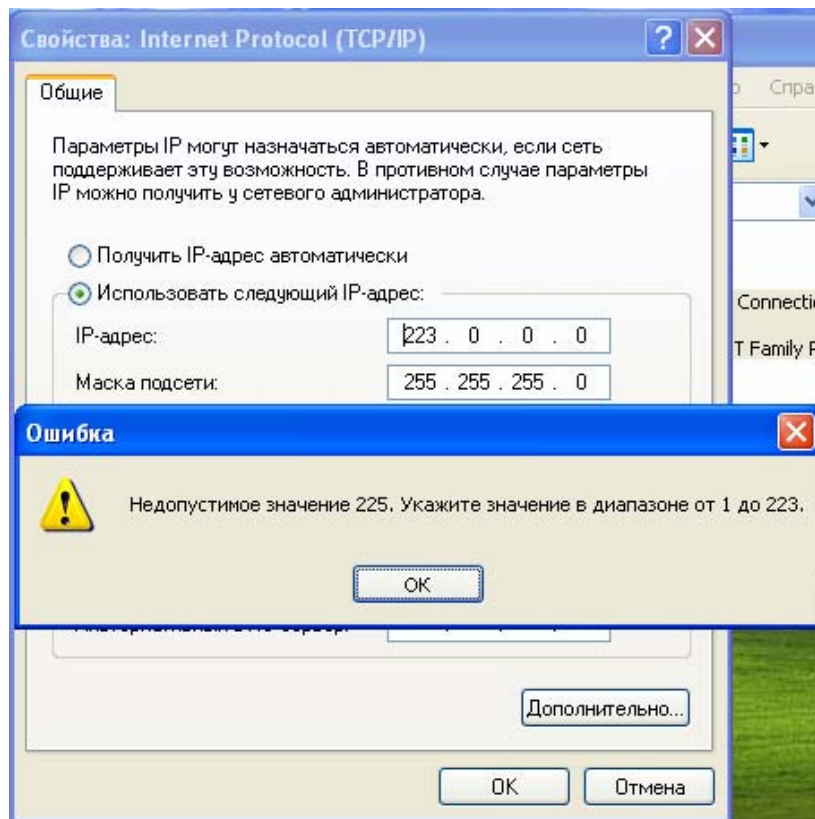
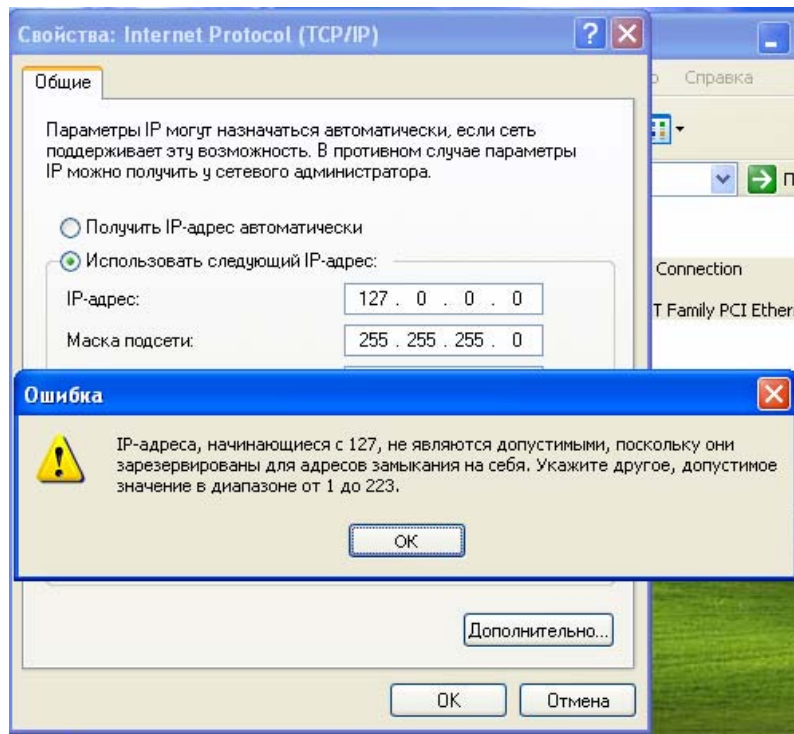
При конфигурировании IP протокола в современных ОС происходит автоматическая проверка правильности вносимых данных, т.е. если осуществляется попытка ввода недопустимых значений в соответствующие поля – система выдает ошибку с пояснением причин возникновения. К таким ситуациям относятся:

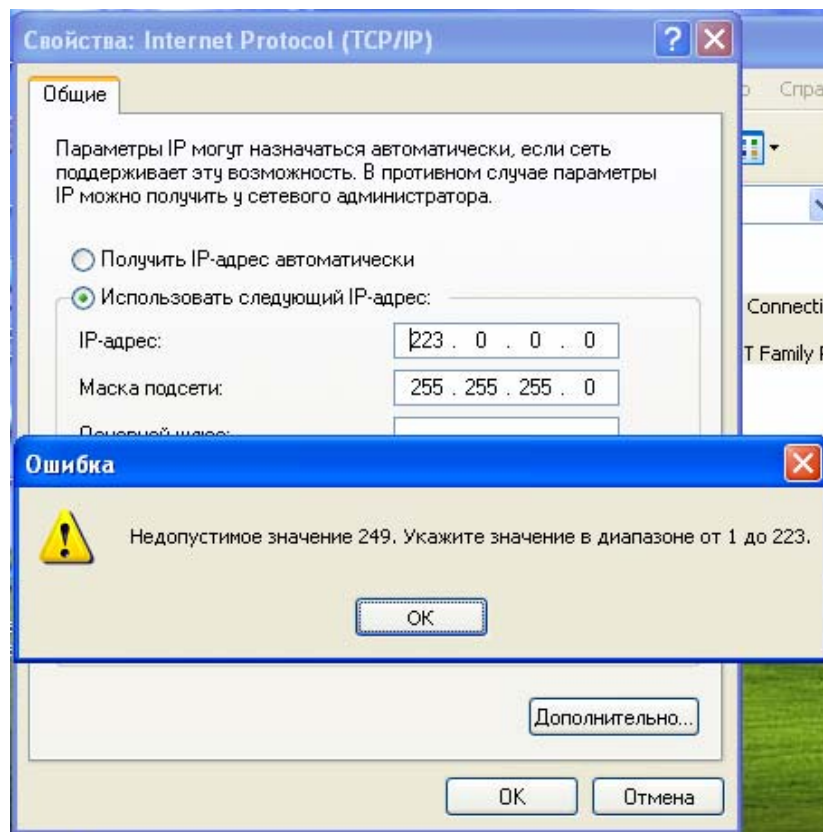
- попытка назначения адреса хоста несоответствующего введенной маске



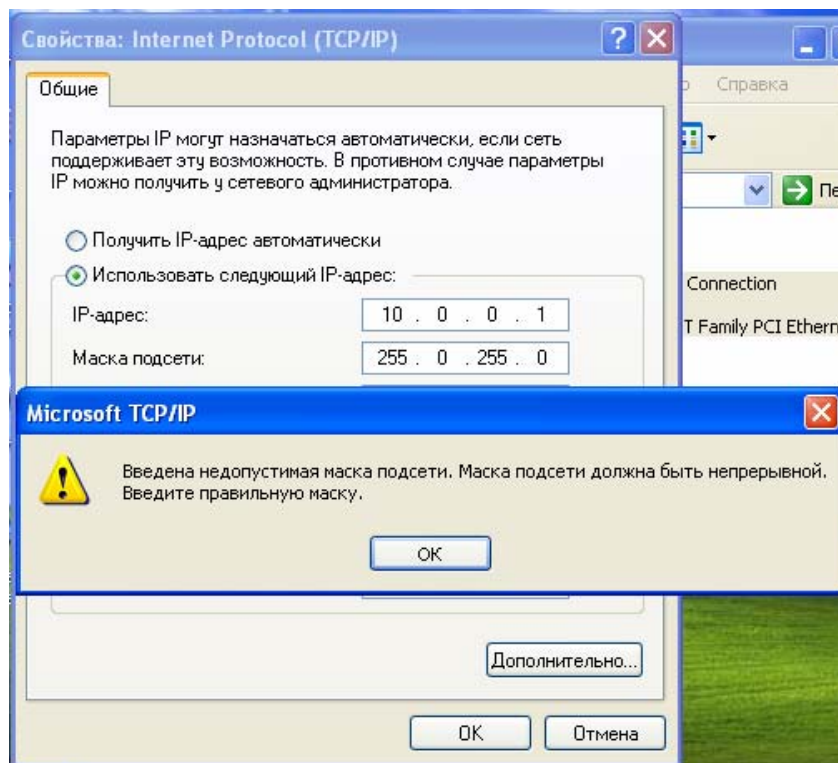
- назначение в качестве адреса хоста специальных адресов



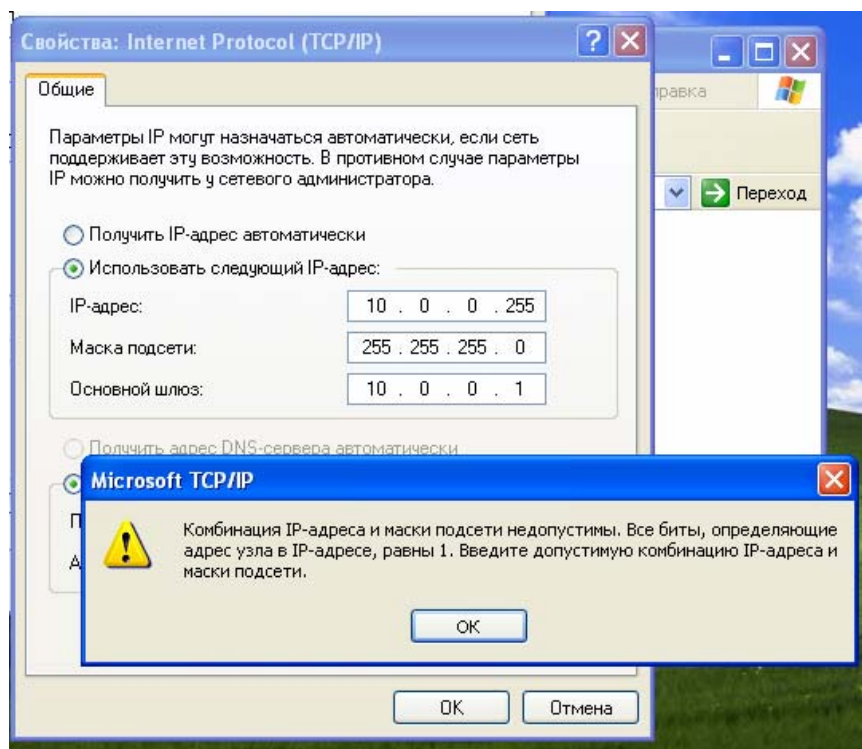




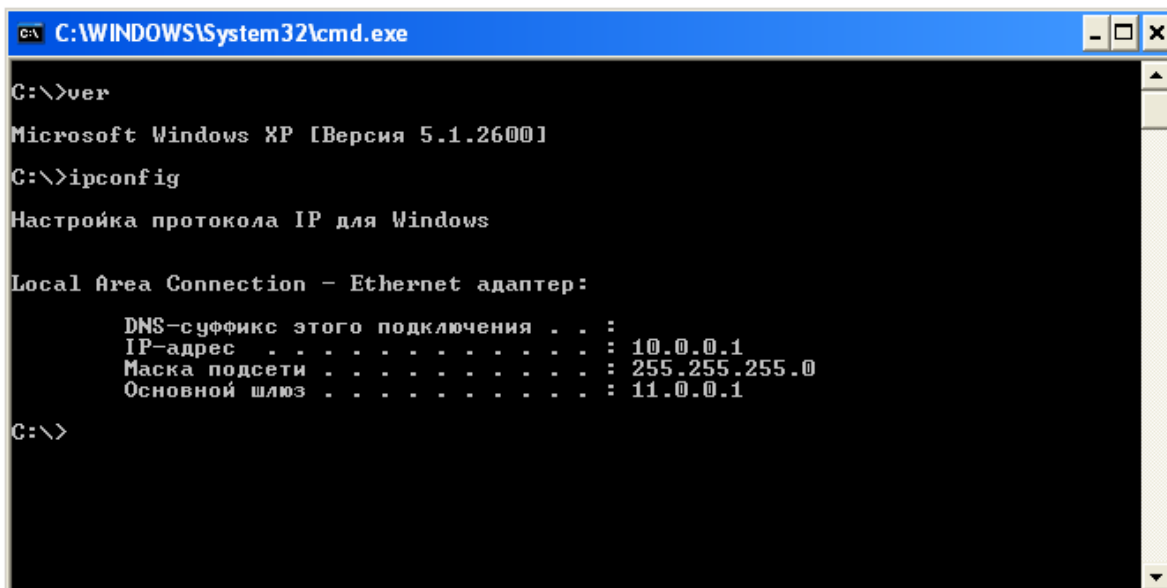
- назначение «дырявой» маски IP адреса



- назначение в качестве адреса хоста широковещательного адреса



При этом отдельные нюансы все-таки имеют место и в современных ОС. Например, возможность назначения адреса шлюза лежащего в сети отличной от сети интерфейса



Такая конфигурация хоста приведет к невозможности работы с другими сетями, хотя внутри сети взаимодействие будет осуществляться, как и при правильной конфигурации.

Следует обратить внимание, что подобный контроль не выполняется на старых ОС. И в этом случае необходимо особенно внимательно относиться к настройке протокола IP, т.к. система не контролирует введенные значения, а принимает их «как есть». Но, конечно же, при использовании неверных настроек – работа IP протокола в штатном режиме невозможна, т.е. неправильная конфигурация в этом случае станет причиной различного рода ошибок.

Задание для закрепления материала:

- Назначить сетевому интерфейсу поочередно IP адрес из сети класса А, В, С выбрав в качестве шлюза первый адрес в сети
- Сымитировать назначение «неправильных» адресов, объяснить причину возникновения ошибки.
- Просмотреть настройки IP протокола из консоли.