

## 6. АДРЕСАЦИЯ В СЛОЖНО СТРУКТУРИРОВАННЫХ СЕТЯХ

Под сложно структурированной сетью будем понимать несколько логических сегментов сети, соединенных оборудованием с элементами коммутации и маршрутизации. Важнейшую роль при взаимодействии хостов в подобной сети играет определение *идентификатора подсети (Network ID)* и *идентификатора хоста (Host ID)*.

Для определения того, какая часть IP-адреса отвечает за ID подсети, а какая за ID хоста, применяются два способа: с помощью классов и с помощью масок.

### 6.1. Классы IP-адресов

Существует пять классов IP-адресов: *A, B, C, D* и *E* (рис. 6.1).



Рис. 6.1. Классы IP-адресов

За принадлежность к тому или иному классу отвечают первые биты IP-адреса. Деление сетей на классы описано в RFC 791 (документ описания протокола IP).

Целью такого деления являлось создание малого числа больших сетей (*класс A*), умеренного числа средних сетей (*класс B*) и большого числа малых сетей (*класс C*).

Если адрес начинается с 0, то сеть относят к *классу A* и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса A имеют номера в диапазоне от 1 до 126. Сетей класса A немного, зато количество узлов в них может достигать  $2^{24} - 2$ , то есть 16 777 214 узлов.

Если первые два бита адреса равны 10, то сеть относится к *классу В*. В сетях класса В под номер сети и под номер узла отводится по 16 бит, то есть по 2 байта. Таким образом, сеть класса В является сетью средних размеров с максимальным числом узлов  $2^{16} - 2$ , что составляет 65 534 узлов.

Если адрес начинается с последовательности 110, то это сеть *класса С*. В этом случае под номер сети отводится 24 бита, а под номер узла – 8 бит. Сети этого класса наиболее распространены, число узлов в них ограничено  $2^8 - 2$ , то есть 254 узлами.

Адрес, начинающийся с 1110, обозначает особый, *групповой адрес* (multicast). Пакет с таким адресом направляется всем узлам, которым присвоен данный адрес.

Адреса *класса Е* в настоящее время не используются (зарезервированы для будущих применений).

Характеристики адресов разных классов представлены в табл. 6.1.

Таблица 6.1

**Характеристики IP-адресов разных классов**

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Количество сетей	Максимальное число узлов в сети
<i>A</i>	0	1.0.0.0	126.0.0.0	126	$2^{24} - 2 = 16\,777\,214$
<i>B</i>	10	128.0.0.0	191.255.0.0	16384	$2^{16} - 2 = 65\,534$
<i>C</i>	110	192.0.0.0	223.255.255.0	2097152	$2^8 - 2 = 254$
<i>D</i>	1110	224.0.0.0	239.255.255.255	Групповой адрес	
<i>E</i>	11110	240.0.0.0	247.255.255.255	Зарезервирован	

Применение классов в целом характеризуется неэффективностью распределения IP-адресов. Например, если организации требуется тысяча IP-адресов, ей выделяется сеть *класса В*, при этом 64534 адреса не будут использоваться.

Существует два основных способа решения этой проблемы:

- более эффективная схема деления на подсети с использованием масок (RFC 950);
- применение протокола IP версии 6 (IPv6).

## 6.2. Использование масок

**Маска подсети** (subnetmask) – это число, которое используется в паре с IP-адресом; двоичная запись маски содержит единицы в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети.

Для стандартных классов сетей маски имеют следующие значения:

- *класс А* – 11111111.00000000.00000000.00000000 (255.0.0.0);
- *класс В* – 11111111.11111111.00000000.00000000 (255.255.0.0);
- *класс С* – 11111111.11111111.11111111.00000000 (255.255.255.0).

Маска подсети записывается либо в виде, аналогичном записи IP-адреса, например 255.255.255.0, либо совместно с IP-адресом с помощью указания числа единичных разрядов в записи маски, например 192.168.1.1/24, т. е. в маске содержится 24 единицы (255.255.255.0).

При использовании масок можно вообще отказаться от понятия классов.

*Пример 1.* Пусть задан IP-адрес 17.239.47.94, маска подсети 255.255.0.0 (другая форма записи: 17.239.47.94/16).

Требуется определить ID подсети и ID хоста в обеих схемах адресации.

1) *Адресация с использованием классов.* Двоичная запись IP-адреса имеет вид:

00010001.11101111.00101111.01011110.

Так как первый бит равен нулю, адрес относится к *классу А*. Следовательно, первый байт отвечает за ID подсети, остальные три байта – за ID хоста:

ID подсети: 17.0.0.0. ID хоста: 0.239.47.94.

2) *Адресация с использованием масок.* Запишем IP-адрес и маску подсети в двоичном виде:

IP-address: 17.239.47.94 = 00010001.11101111.00101111.01011110 ,

Subnetmask: 255.255.0.0 = 11111111.11111111.00000000.00000000 .

Вспомнив определение маски подсети, можно интерпретировать номер подсети как те биты, которые в маске равны 1, т.е. первые два байта. Оставшаяся часть IP-адреса будет номером узла в данной подсети.

ID подсети: 17.239.0.0. ID хоста: 0.0.47.94.

Номер подсети можно получить другим способом, применив к IP-адресу и маске операцию логического умножения или *конъюнкции (AND)*:

AND	00010001.	11101111.	00101111.	01011110
	11111111.	11111111.	00000000.	00000000
	00010001.	11101111.	00000000.	00000000
	17	239	0	0

В масках количество единиц в последовательности, определяющей границу номера сети, не обязательно должно быть кратным 8.

*Пример 2.* Задан IP-адрес 192.168.89.16, маска подсети – 255.255.192.0 (другая форма записи: 192.168.89.16/18).

Требуется определить ID подсети и ID хоста. Воспользуемся операцией AND:

IP-address: 192.168.89.16 =  $\text{AND}$  11000000.10101000.01011001.00010000

Subnetmask: 255.255.0.0 = 11111111.11111111.11000000.00000000.

subnetID:	11000000.	10101000.	01000000.	00000000
	192	168	64	0

Чтобы получить номер узла, нужно в битах, отвечающих за номер подсети, поставить нули:

HostID: 00000000.00000000.00011001.00010000 = 0.0.25.16.

*Ответ:* ID подсети = 192.168.64.0, ID хоста = 0.0.25.16.

*Для масок существует важное правило: разрывы в последовательности единиц или нулей недопустимы.*

Например, не существует маски подсети, имеющей следующий вид:

11111111.11110111.00000000.00001000 (255.247.0.8),

так как последовательности единиц и нулей не являются непрерывными.

С помощью масок администратор может структурировать свою сеть, не требуя от поставщика услуг дополнительных номеров сетей.

Допустим, организации выделена сеть класса *B*: 160.95.0.0 (рис. 6.2).

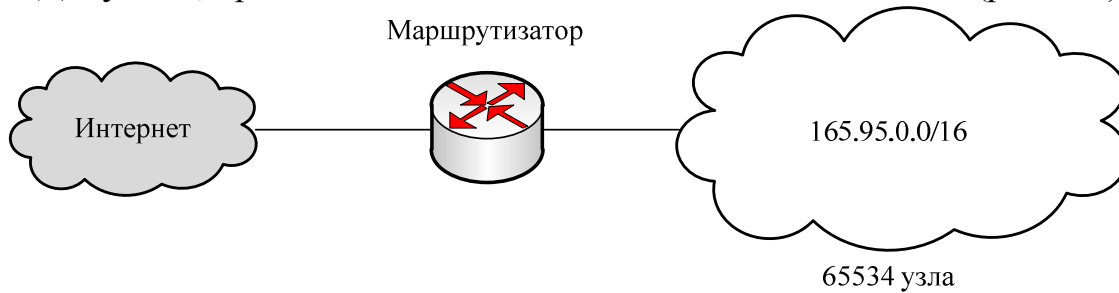


Рис. 6.2. Сеть

класса *B* до деления на подсети

В такой сети может находиться до 65534 узлов. Однако организации требуются 3 независимые сети с числом узлов в каждой не более 254. В этой ситуации можно применить деление на подсети с помощью масок. Например, при использовании маски 255.255.255.0 третий байт адреса будет определять номер внутренней подсети, а четвертый байт – номер узла (рис. 6.3).

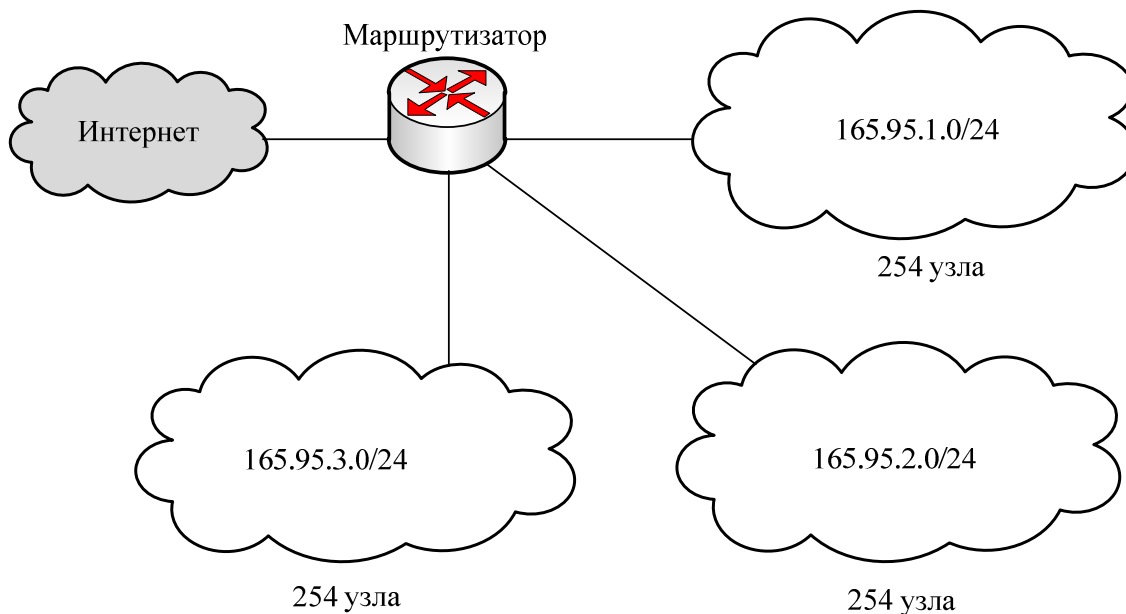


Рис. 6.3. Сеть

класса *B* после деления на подсети

Маршрутизаторы во внешней сети (Интернет) ничего «не знают» о делении сети 160.95.0.0 на подсети, все пакеты направляются на маршрутизатор организации, который переправляет их в требуемую внутреннюю подсеть.

### 7.1. Разработка программы определения ID подсети и ID хоста по заданному IP-адресу и маске подсети

Входными данными программы являются IP-адрес (переменная типа `unsigned long ip;`) и маска подсети (`unsigned long mask;`). Необходимо определить ID подсети (`unsigned long subnet;`) и ID хоста (`unsigned long host;`). Вспомнив определение маски, можно интерпретировать номер подсети как те биты, которые в маске равны 1. Тогда, ID подсети можно получить, применив к IP-адресу и маске операцию логического умножения, или конъюнкции (AND):

```
subnet = ip & mask;
```

Чтобы получить ID хоста, нужно в битах, отвечающих за номер подсети, поставить нули:

```
host = ip & ~mask;
```

Пусть задан IP-адрес 192.168.89.16, маска подсети 255.255.0.0.

192.168.89.16 = 11000000.10101000.01011001.00010000,

255.255.0.0 = 11111111.11111111.00000000.00000000.

Получим ID подсети: 192.168.0.0. ID хоста: 0.0.89.16 (рис. 6.4)

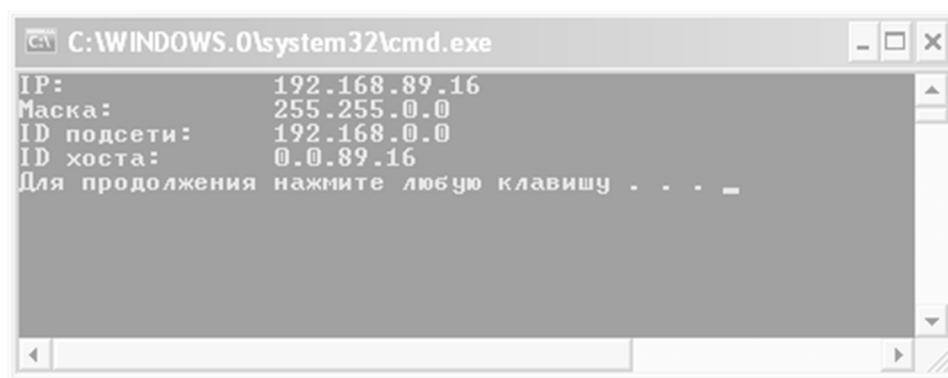


Рис. 6.4. Результат выполнения программы

В ходе выполнения задания необходимо проверить пределы допустимого диапазона октета: от 0 до 255, что и выполняет функция CheckAddress:

```
bool CheckAddress(char* ip_)
{
    int points=0,    // количество точек
    numbers=0;      // значение октета
    char* buff;      // буффер для одного октета
    buff = new char[3];

    for(int i=0; ip_[i]!='\0'; i++)
        // для строки IP-адреса
    {
        if(ip_[i]<='9' && ip_[i]>='0') // если цифра
        {
            if(numbers>3) return false;
            //если больше трех чисел в октете - ошибка
            buff[numbers++] = ip_[i];
            //скопировать в буффер
        }
        else if(ip_[i]=='.') // если точка
        {
            if(atoi(buff)>255) return false;
            // проверить диапазон октета
            if(numbers==0) return false;
            //если числа нет - ошибка
        }
    }
}
```

```

        numbers=0;
        points++;
    }
    else return false;
}

if(points!=3) return false;
// если количество точек в IP-адресе не 3 - ошибка
return true;
}

```

Затем необходимо выполнить преобразование строки IP-адреса (16 символов) в десятично-точечной нотации, содержащей 4 группы по 3 символа (октета), разделенных точками, в число `unsigned long`. Это можно выполнить, написав собственную функцию проверки и преобразования `CharToULong`:

```

unsigned long  CharToULong(char* ip_)
{
    unsigned long  out=0; //число для IP-адреса
    char *buff;
    buff =newchar[3];
        //буфер для хранения одного октета

    for(inti=0,j=0;ip_[i]!='\0';i++,j++)
    {
        if(ip_[i]!='.') //если не точка
            buff[j]=ip_[i]; // записать символ в буфер
        if(ip_[i]=='.'||ip_[i+1]=='\0')
        {
            // если следующий октет или последний

            out<<=8; //сдвинуть число на 8 бит
            if(atoi(buff)>255) return NULL;

            // елиоктетбольше 255 - ошибка
            out += (unsigned long)atoi(buff);

            //преобразовать и добавить к числу IP-адреса
            j=-1;
        };
    }

    returnout;
}

```

Аналогично для маски следует проверить пределы допустимого диапазона октета: от 0 до 255и выполнить преобразование в число. После чего, надо проверить корректность задания маски подсети. Выполнить вычисления ID подсети, IDхоста; преобразовать их к форматированной строке и вывести результат (рис. 6.4).

## Лабораторная работа № 7

*Цель:* изучение методов определения идентификатора сети и идентификатора хоста.

*Задание:* лабораторная работа заключается в разработке программы, выполняющей по введенному IP-адресу и маске определение Network ID и Host ID.

*Дополнительные задания:*

1. Проверка маски на непрерывность единиц.
2. Проверка правильности IP-адреса (четыре октета чисел от 0 до 255, разделенных точкой) и маски на разрешенные символы, на длину (не более 32 бит), количество октетов и т.д.

В табл. 6.2 представлены примеры корректных и некорректных IP-адресов и масок.

Таблица 6.2

### Примеры правильных и неправильных IP-адресов и масок

Правильные IP-адреса	172.16.192.1 192.168.0.1 10.0.0.1
Неправильные IP-адреса	256.0.0.1 1.1.1.1.1 1.0.1 1:1:1:1
Правильная маска	255.0.0.0 255.240.0.0 255.255.128.0 255.255.255.240
Неправильная маска	0.255.0.0 255.0.255.0 255.240.255.0 1.1.1.1 0.0.0.128