**6. АДРЕСАЦИЯ В СЛОЖНОСТРУКТУРИРОВАННЫХ СЕТЯХ**

Под сложноструктурированной сетью будем понимать несколько логических сегментов сети со своей адресацией, 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).

Номер узла

Номер сети

0

1 байт

3 байта

4 байта

Номер узла

Номер сети

1

2 байта

2 байта

0

Номер узла

Номер сети

1

3 байта

1 байт

1

0

Адрес группы Multicast

1

1

1

0

Зарезервирован

1

1

1

1

0

Класс A

Класс B

Класс C

Класс D

Класс E

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 (*класса А*), умеренного числа средних сетей (*класс* *В*) и большого числа малых сетей (*класс С*).

Если адрес начинается с 0, то сеть относят к *классу А* и номер сети занимает один байт, остальные 3 байта интерпретируются как номер узла в сети. Сети класса А имеют номера в диапазоне от 1 до 126. Сетей класса А немного, зато количество узлов в них может достигать 224 – 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 216 – 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 28 – 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 |
| *А* | 0 | 1.0.0.0 | 126.0.0.0 | 126 | 224 – 2 = 16777214 |
| *В* | 10 | 128.0.0.0 | 191.255.0.0 | 16384 | 216 – 2 = 65534 |
| *С* | 110 | 192.0.0.0 | 223.255.255.0 | 2097152 | 28 – 2 = 254 |
| *D* | 1110 | 224.0.0.0 | 239.255.255.255 | Групповой адрес | |
| *Е* | 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 64 534 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**(subnet mask) – 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 ,

Subnet mask: 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 = AND11000000.10101000.01011001.00010000  
Subnet mask: 255.255.0.0 =  11111111.11111111.11000000.00000000.  
subnet ID: 11000000.10101000.01000000.00000000

192 168 64 0

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

Host ID: 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 *В*: 160.95.0.0 (рис. 6.2).



Рис. 6.2. Сеть класса *В* до деления на подсети

В такой сети может находиться до 65 534 узлов. Однако организации требуется 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 *В* 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.

**6.3. Особые IP-адреса**

Некоторые IP-адреса являются особыми, они не должны применяться для идентификации обычных сетей.

1. Если первый октет ID сети начинается с 127, такой адрес считается адресом машины-источника пакета. В этом случае пакет не выходит в сеть, а возвращается на компьютер-отправитель. Такие адреса называются **loopback**(«**петля**», «замыкание на себя») и используются для проверки функционирования стека TCP/IP.
2. Если все биты IP-адреса равны нулю, адрес обозначает узел-отправитель и используется в некоторых сообщениях ICMP.
3. Если все биты адреса равны 1, адрес называется ***ограниченным широковещательным***(limited broadcast).Пакеты, направленные по такому адресу рассылаются всем узлам той подсети, в которой находится отправитель пакета. Также возможна другая форма записи ограниченного широковещательно адреса – все биты ID хоста равны 1, а все биты ID-сети равны 0.
4. Если все биты ID хоста равны 1 (при этом все биты ID-сети не равны 0, т.е. задается определенная сеть, а не сеть отправителя), адрес называется ***широковещательным***(broadcast);пакеты, имеющие широковещательный адрес, доставляются всем узлам подсети назначения.
5. Если все биты ID хоста равны 0, адрес считается **идентификатором подсети** (subnet ID).

Наличие особых IP-адресов объясняет, почему из диапазона доступных адресов исключаются два адреса – это случаи, когда все биты ID хоста равны 1 или 0. Например, в сети *класса* *С* не 256, а 254 узлов.

**6.4. 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;

Чтобы получить broadcast-адрес необходимо выполнить:

broadcast = ip & mask | ~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 Broadcast: 192.168.255.255 (рис. 6.4)

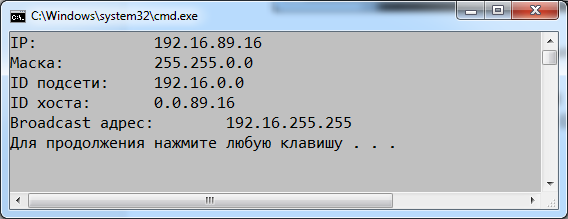


Рис. 6.4. Результат выполнения программы

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)

// проверить диапазон октета

return false;

if(numbers==0)

//если числа нет - ошибка

return false;

numbers=0;

points++;

delete[]buff;

buff = new char[3];

}

else return false;

}

if(points!=3)

// если количство точек в IP-адресе не 3 - ошибка

return false;

if(numbers==0||numbers>3)

return false;

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:

|  |
| --- |
| ul CharToLong(char\* ip\_)  {  ul out=0;//число для IP-адреса  char \*buff;  buff=new char[3];  //буфер для хранения одного октета |
| for(int i=0,j=0,k=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;  // еcли октет больше 255 – ошибка  out += (ul)atoi(buff);  //преобразовать и добавить  //к числу IP-адреса  k++;  j=-1;  delete[]buff;  buff=new char[3];  }  }  return out;  } |

Аналогично для маски следует проверить пределы допустимого диапазона октета: от 0 до 255 и выполнить преобразование в число. После чего, надо проверить корректность задания маски подсети.

|  |
| --- |
| ul ip,mask,host,subnet, broadcast;  char \*ip\_,\*mask\_;  bool flag=true;  ip\_=new char[16];  mask\_=new char[16];  do  {  if(!flag) cout<<"Неверно введён адрес!"<<endl;  cout<<"IP: ";  cin>>ip\_;  }while(!(flag=CheckAddress(ip\_))); |

|  |
| --- |
| ip=CharToLong(ip\_);  flag=true;  do  {  if(!flag) cout<<"Неправильная маска!"<<endl;  flag=true;  do  {  if(!flag) cout<<"Неверно введена маска!"<<endl;  cout<<"Маска: ";  cin>>mask\_;  }while(!(flag=CheckAddress(mask\_)));  mask=CharToLong(mask\_);  }while(!(flag=CheckMask(mask))); |

Выполнить вычисления ID подсети, ID хоста; преобразовать их к форматированной строке и вывести результат (рис. 6.4).

**6.5. Лабораторная работа №8**

**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 |

Продолжение таблицы 6.2

|  |  |
| --- | --- |
| 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 |

1. Расчет широковещательного (broadcast) адреса по известному IP-адресу и маске.