Организация подсетей и бесклассовая адресация

Реализация адресации подсетей с помощью масок

Если бит IP-адреса относится к префиксу подсети, то соответствующий ему бит маски полагается равным 1. Если же бит IP-адреса относится к идентификатору узла сети, то соответствующий бит маски полагается равным 0.

	0	8	16	24 31
Маска класса A	11111111	00000000	00000000	00000000
	255	0	0	0
Маска класса B	11111111	11111111	00000000	00000000
	255	255	0	0
Маска класса C	11111111	11111111	11111111	00000000
	255	255	255	0

Реализация адресации подсетей с помощью масок

	Сетевой префикс		Подсеть	Узел
IP-адрес	10010001	10010001	00010011	00010111
	145	145	19	23
Маска	11111111	11111111	11111111	00000000
	255	255	255	0
	Расширенный сетевой префикс			

Реализация адресации подсетей с помощью масок

	Сетевой префикс		Подсеть		Узел
IP-адрес	10010001	10010001	00010011	00	010111
	145	145	19	23	
Маска	11111111	11111111	11111111	11	000000
	255	255	255	192	
	Расширенный сетевой префикс				

В настоящее время чаще используют обозначение вида "/xx", где xx – количество бит в расширенном сетевом префиксе.

Таким образом, вместо записи IP-адреса 145.145.19.23 с маской 255.255.255.192, мы можем использовать следующую запись **145.145.19.23/26**, которая более компактная и понятная.

Реализация адресации подсетей с помощью масок

Маска	*Десятичная форма*	*Маска*	*Десятичная форма*
/1	128.0.0.0	/17	255.255.128.0
/2	192.0.0.0	/18	255.255.192.0
/3	224.0.0.0	/19	255.255.224.0
/4	240.0.0.0	/20	255.255.240.0
/5	248.0.0.0	/21	255.255.248.0.
/6	252.0.0.0	/22	255.255.252.0
/7	254.0.0.0	/23	255.255.254.0
/8	255.0.0.0	/24	255.255.255.0
/9	255.128.0.0	/25	255.255.255.128
/10	255.192.0.0	/26	255.255.255.192
/11	255.224.0.0	/27	255.255.255.224
/12	255.240.0.0	/28	255.255.255.240
/13	255.248.0.0	/29	255.255.255.248
/14	255.252.0.0	/30	255.255.255.252
/15	255.254.0.0	/31	255.255.255.254
/16	255.255.0.0	/32	255.255.255.255

Маска подсети переменной длины

Общая схема разбиения сети на подсети с масками переменной длины следующая: сеть делится на подсети максимально необходимого размера; затем некоторые подсети делятся на более мелкие; и так далее, до тех пор, пока это необходимо.

	Сетевой префикс		Подсеть			Узел
145.145.0.0/25	10010001	10010001	0000000	0	0	0000000
	145	145	0			0
	Расширенный сетевой префикс					

Чтобы предоставить свободу пользователем при выборе способа адресации подсетей, разработчики протокола TCP/IP предусмотрели возможность использования подсетей переменной длины. При этом узловая часть IP-адреса может разбиваться независимо для каждой сети. Однако после того, как система адресации определена, о ней должно стать известно всем компьютерам, подключенным к этим сетям.

Пример организации подсетей 1

Задача:
Организации выделен блок адресов 230.220.15.0/24. Разбить блок на 4 подсети, наибольшая из которых насчитывает 50 узлов. Учесть возможный рост в 10%.

	Сетевой префикс				Узел
	0	8	16	24 25	31
230.220.15.0/24	11100110	11011100	00001111	0 0	000000
	230	220	15	0	
	Расширенный сетевой префикс				

Рис. Организация 4-х подсетей. Для этого выделяется из узловой части два бита

Пример организации подсетей 1

	Сетевой префикс				Узел
	0	8	16	24 25	31
230.220.15.0/24	11100110	11011100	00001111	0 0	000000
	230	220	15	0	
230.220.15.0/26	11100110	11011100	00001111	0 0	000000
	230	220	15	0	
230.220.15.64/26	11100110	11011100	00001111	0 1	000000
	230	220	15	64	
230.220.15.128/26	11100110	11011100	00001111	1 0	000000
	230	220	15	128	
230.220.15.192/26	11100110	11011100	00001111	1 1	000000
	230	220	15	192	
	Расширенный сетевой префикс				

Рис. Номера 4-х подсетей

Пример организации подсетей 1

	Сетевой префикс				Узел
	0	8	16	24 25	31
230.220.15.63/26	11100110	11011100	00001111	0 0	111111
	230	220	15	63	
230.220.15.127/26	11100110	11011100	00001111	0 1	111111
	230	220	15	127	
230.220.15.191/26	11100110	11011100	00001111	1 0	111111
	230	220	15	191	
230.220.15.255/26	11100110	11011100	00001111	1 1	111111
	230	220	15	255	
	Расширенный сетевой префикс				

Рис. Широковещательные адреса для 4-х подсетей

Пример организации подсетей 1

Номер подсети	Наименьший адрес подсети	Наибольший адрес подсети
0	230.220.15.1	230.220.15.62
1	230.220.15.65	230.220.15.126
2	230.220.15.129	230.220.15.190
3	230.220.15.193	230.220.15.254

Рис. Адреса узлов в каждой из 4-х подсетей

Пример организации подсетей 2

Задача:
Компании выделен блок адресов 145.145.0.0/16. Нужно разбить адресное пространство на три части, выделить адреса для двух пар маршрутизаторов и оставить некоторый резерв.

	0	8	16	24	31
145.145.0.0/18	10010001	10010001	0 0	000000	00000000
	145	145	0		0
145.145.64.0/18	10010001	10010001	0 1	000000	00000000
	230	145	64		0
145.145.128.0/18	10010001	10010001	1 0	000000	00000000
	145	145	128		0
145.145.192.0/18	10010001	10010001	1 1	000000	00000000
	145	145	192		0
	Расширенный сетевой префикс				

Рис. Организация 4-х подсетей из адреса 145.145.0.0/16

Пример организации подсетей 2

	0	8	16	24	31
145.145.192.0/30	10010001	10010001	1 1 000000	000000	0 0
	145	145	192	0	
145.145.192.4/30	10010001	10010001	1 1 000000	000001	0 0
	230	145	192	4	

Рис. Организация в подсети 145.145.192.0/18 двух подсетей с маской /30

Пример организации подсетей 3

Задача:
Компания организовывает корпоративную сеть. Имеется четыре региональных офиса, связанные каналами с центральным офисом. К региональным офисам, в свою очередь, подключены областные филиалы данного региона. Решено использовать сеть 10.0.0.0/8 для корпоративной сети. Требуется составить схему IP-адресации компании, выбрав способ адресации наилучший с точки зрения маршрутизации.

Региональный офис	Подключено областных филиалов	Процент
офис A	10	36%
офис B	7	25%
офис C	3	11%
офис D	3	11%

Рис. Количество подключенных филиалов к каждому региональному офису

Пример организации подсетей 3

Региональный офис	Процент адресного пространства	Диапазон адресов	Блок выделенных адресов
офис A	25%	10.0.0.0 – 10.63.0.0	10.0.0.0/10
офис B	25%	10.64.0.0 – 10.127.0.0	10.64.0.0/10
офис C	12,5%	10.128.0.0 – 10.159.0.0	10.128.0.0/11
офис D	12,5%	10.160.0.0 – 10.191.0.0	10.160.0.0/11
резерв	25%	10.192.0.0 – 10.255.0.0	10.192.0.0/10

Рис. Диапазоны адресов каждого регионального офиса

Алгоритм маршрутизация при наличии подсетей

Все подсети, которым назначен одинаковый сетевой префикс, должны быть смежными. Во всех подсетях должны использоваться унифицированные маски, а все компьютеры должны поддерживать алгоритм маршрутизации подсетей.

1	Извлечь из дейтаграммы IP-адрес конечного получателя и выделить сетевой префикс netid
2	Если netid совпадает с сетевым префиксом одной из сетей, к которой непосредственно подключен маршрутизатор, выполнить прямую доставку дейтаграммы конечному получателю по соответствующей сети (которая включает определение физического адреса конечного получателя, вложение дейтаграммы в физический кадр и его отправка получателю)
3	Иначе, повторить шаг 2 для каждого элементы таблицы маршрутизации
4	Положить сетевой префикс равным поразрядному логическому И между IP-адресом и маской подсети, извлеченной из текущего элемента таблицы маршрутизации
5	Если netid равно значению полю адреса сети текущего элемента таблицы маршрутизации, переслать дейтаграмму по адресу ближайшей точки перехода, указанной в соответствующей колонке для текущего элемента таблицы маршрутизации
6	Повторить шаг 5 для следующего элемента таблицы маршрутизации
7	Если в результате нужный элемент в таблице маршрутизации не найден, то сгенерировать ошибку маршрутизации

Бесклассовая междоменная адресация CIDR

Идея использования методики *бесклассовой междоменной маршрутизации* (*Classless Inter Domain Routing*, или *CIDR*) заключается в том, что несколько непрерывных блоков адресов записываются в виде одного элемента, состоящего из пары значений: н*ачальный адрес сети* и *количество*. Поле *начальный адрес сети* подразумевает минимальное значение адреса в блоке, а параметр *количество* определяет общее число используемых блоков адресов.

Использование бесклассовой адресации позволяет провайдерам Интернет выделять своим клиентам непрерывный блок IP-адресов нужного размера. При этом сами IP-адреса рассматриваются как обычные целые числа, а количество адресов в блоке всегда кратно 2n.

Структуры данных и алгоритмы для бесклассового поиска

Для повышения эффективности поиска маршрутов следования дейтаграмм при использовании бесклассовой адресации в таблицах маршрутизации, необходимо применять структуры данных и алгоритмы отличные от тех, что используются для поиска классовых адресов. Для этих целей рекомендуется использовать методы, основанные на двоичных деревьях.

Протоколы начальной загрузки и автоконфигурации узлов

Протоколы конфигурации

Рассмотренный в предыдущих главах механизм определения IP-адреса при начальной загрузке с помощью протокола RARP имеет три недостатка:

• протокол RARP функционирует на низком уровне, использующая его прикладная программа должна взаимодействовать напрямую с сетевым оборудованием;
• формат протокола не предусматривает возможность передачи расширенных управляющих данных;
• протокол нельзя использовать в сетях, где аппаратные адреса назначаются динамически.

Чтобы компенсировать недостатки протокола RARP, был разработан новый *протокол начальной загрузки* (*BOOTstrap Protocol*, или *BOOTP*). Позже был создан *протокол динамической конфигурации узла сети* (*Dynamic Host Configuration Protocol*, или *DHCP*), который пришел на смену протоколу BOOTP.

Протокол начальной загрузки BOOTP

Прикладная программа может использовать ограниченное широковещание для отправки дейтаграммы в локальную сеть, но при этом сам узел, на котором работает прикладная программа, может не знать свой IP-адрес.

Для предотвращения возникновения дополнительного трафика, при использовании BOOTP, в перегруженной сети, значение ожидания ответа (тайм-аута) удваивается после каждой неудачной передачи. Для предотвращения одновременной передачи начальное значение тайм-аута должно выбираться случайно из определенного интервала.

Формат BOOTP-сообщения

0	8	16	24	31
Тип пакета	Тип оборудования	Длина физ. адреса	Число переходов	
Идентификатор транзакции				
Время после начала загрузки клиента		Не используется		
IP-адрес клиента				
Выделенный IP-адрес				
IP-адрес сервера				
IP-адрес маршрутизатора				
Физический адрес клиента (16 байт) …				
Имя сервера (64 байта) …				
Имя загрузочного файла (128 байт) …				
Поле, формат которого определяет производитель оборудования (64 байта) …				

Протокол динамической конфигурации узла DHCP

Этот протокол стал расширением протокола BOOTP и позволил решить две новые важные задачи:

- получение компьютером всей необходимой информации о конфигурации в одном сообщении;
- быстрое динамическое назначение компьютеру IP-адреса.

В протоколе DHCP предусмотрены три типа присвоения адресов:

- ручная конфигурация узлов сети; администратор может сам назначать определенные адреса заданным компьютерам;
- автоматическая конфигурация узлов сети; DHCP-сервер может присваивать постоянные адреса узлам сети при первом подключении;
- динамическая конфигурация узлов сети; сервер выделяет адреса компьютерам на определенный период времени.

Протокол динамической конфигурации узла DHCP

Протокол DHCP является автоматически конфигурируемым, так как позволяет получать узлу сети от сервера все необходимые для взаимодействия параметры без вмешательства администратора, но который контролирует этот процесс настройкой сервера.

Динамическое присвоение адресов является оптимальной схемой для клиентов, подключаемых к сети временно, или совместно использующих один и тот же набор IP-адресов и не нуждающихся в постоянных адресах.

Протокол динамической конфигурации узла DHCP

Основные задачи, которые решаются с помощью протокола DHCP следующие:

- DHCP представляет собой механизм, а не политику. DHCP должен управляться местными системными администраторами, путем задания желательных конфигурационных параметров.
- Клиенты не должны требовать ручной конфигурации. Каждый клиент должен быть способен прочесть локальные конфигурационные параметры.
- Сети не должны требовать ручной конфигурации для отдельных клиентов. В нормальных условиях, сетевой администратор не должен вводить какие-либо индивидуальные конфигурационные параметры клиента.
- DHCP не требует отдельного сервера для каждой субсети.
- Клиент DHCP должен быть готов получить несколько откликов на запрос конфигурационных параметров. Для повышения надежности и быстродействия можно использовать несколько DHCP-серверов, обслуживающих перекрывающиеся области сети.
- DHCP должен сосуществовать с ЭВМ, которые сконфигурированы вручную.
- DHCP должен быть совместим с логикой работы BOOTP-агента.
- DHCP должен обслуживать существующих клиентов BOOTP.

Протокол динамической конфигурации узла DHCP

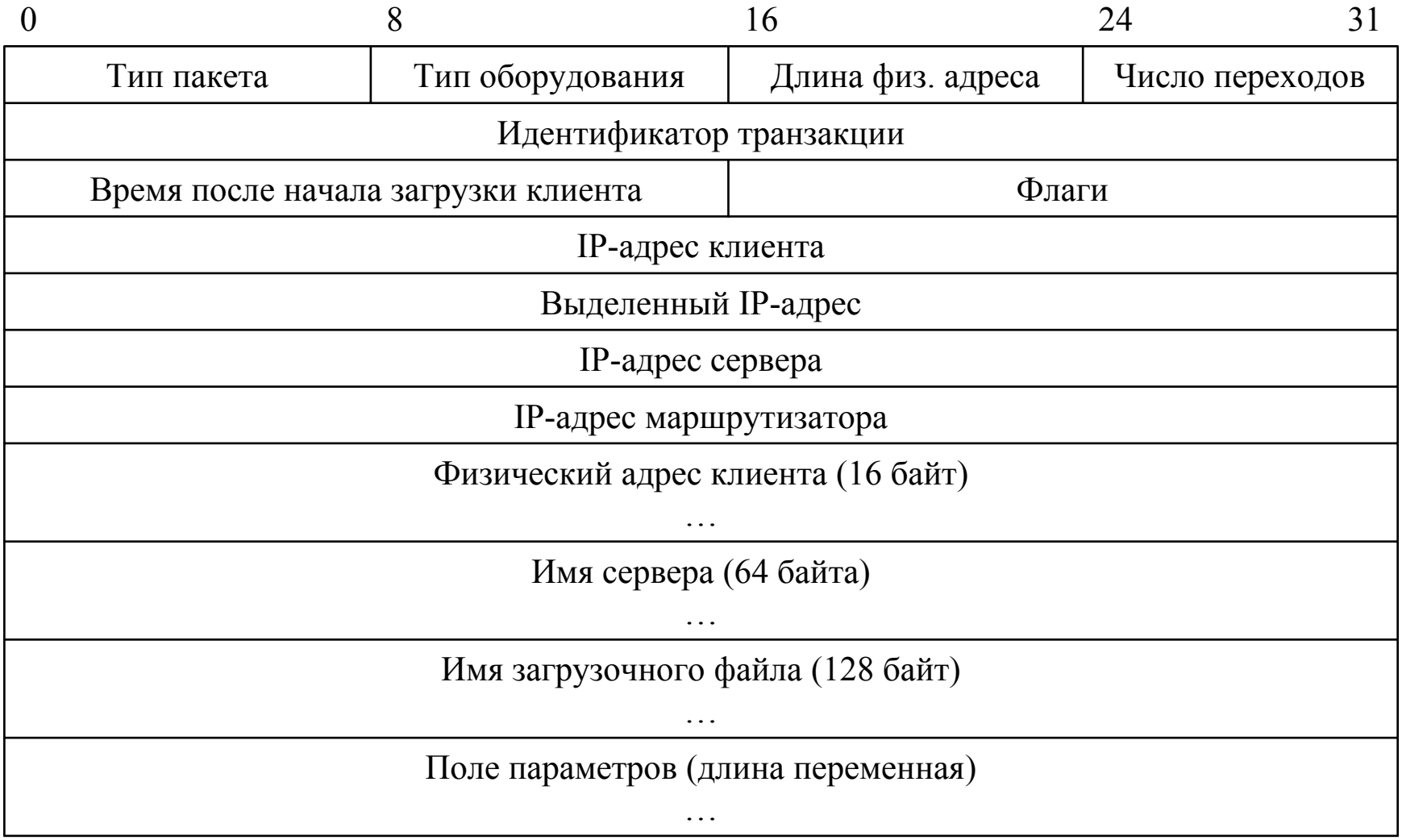
При взаимодействии клиент/сервер протокол DHCP должен:

• Гарантировать, что любой специфический сетевой адрес не будет использоваться более чем одним клиентом DHCP одновременно.
• Поддерживать DHCP конфигурацию клиента при стартовой перезагрузке DHCP-клиента. Клиенту DHCP должен, при каждом запросе по мере возможности, присваиваться один и тот же набор конфигурационных параметров.
• Поддерживать конфигурацию DHCP-клиента при перезагрузке сервера, и, по мере возможности, DHCP-клиенту должен присваиваться один и тот же набор конфигурационных параметров.
• Позволять автоматически присваивать конфигурационные параметры новым клиентам, чтобы избежать ручной конфигурации.
• Поддерживать фиксированное или постоянное присвоение конфигурационных параметров для заданного клиента.

Алгоритм динамического выделения адресов

Сообщение	Описание события
DHCPDISCOVER	Клиент широковещательно посылает сообщение, для обнаружения доступного сервера.
DHCPOFFER	Посылается сервером клиенту в ответ на сообщение DHCPDISCOVER и содержит предложение по конфигурационным параметрам.
DHCPREQUEST	Варианты сообщений клиента серверу: запрос параметров от одного сервера и неявный отказ от предложений других серверов; подтверждение корректности ранее присвоенного адреса; запрос расширения времени жизни конкретного сетевого адреса.
DHCPACK	Посылается сервером клиенту и содержит конфигурационные параметры, включая присвоенный сетевой адрес.
DHCPNAK	Посылается сервером клиенту, сообщая о том, что сетевой адрес не корректен, или время использования адреса клиентом истекло.
DHCPDECLINE	Клиент и сервер обнаружили, что сетевой адрес уже используется.
DHCPRELEASE	Посылается клиентом серверу с целью отказа от сетевого адреса и аннулирует оставшееся время действия адреса.
DHCPINFORM	Посылается клиентом серверу с просьбой о локальных конфигурационных параметрах, при этом клиент уже имеет полученный извне сетевой адрес.

Формат сообщения протокола DHCP



Формат сообщения протокола DHCP

Рис. Формат поля типа DHCP-сообщения, который используется для определения параметров в DHCP-сообщении

Код типа	Тип DHCP-сообщения
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNACK
7	DHCPRELEASE

Рис. Возможные значения типа DHCP-сообщения