МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Гжельский государственный университет»** (ГГУ)

Колледж ГГУ

Специальность 09.02.07 Информационные системы и программирования

**Реферат**

**По дисциплине «Компьютерные сети»**

**на тему «Сетевые протоколы, маска подсети»**

ВЫПОЛНИЛ:

Студент группы ИСП-О-17

Шашков И.С.

ПРОВЕРИЛА:

Прокуронова А.Ю.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

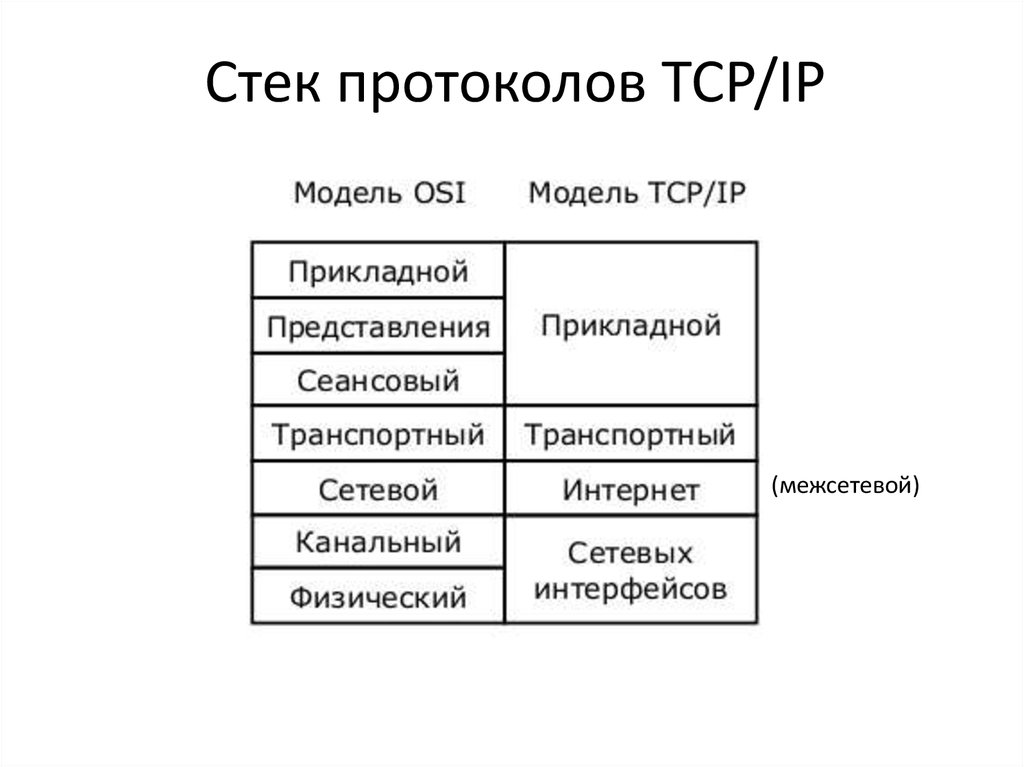
п. Электроизолятор

1. г.

# **Сетевые протоколы**

|  |  |
| --- | --- |
|  | **Сетевой протокол** - это набор правил, позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть компьютерами.Фактически разные протоколы зачастую описывают лишь разные стороны одного типа связи; взятые вместе, они образуют так называемый стек протоколов. Названия "протокол" и "стек протоколов" также указывают и на программное обеспечение, которым реализуется протокол. Уровни протоколов Наиболее распространённой системой классификации сетевых протоколов является так называемая модель OSI. В соответствии с ней протоколы делятся на 7 уровней по своему назначению - от физического (формирование и распознавание электрических или других сигналов) до прикладного (API для передачи информации приложениями): |

* **Прикладной уровень, Application layer** - Верхний (7-й) уровень модели, обеспечивает взаимодействие сети и пользователя. Уровень разрешает приложениям пользователя доступ к сетевым службам, таким как обработчик запросов к базам данных, доступ к файлам, пересылке электронной почты. Также отвечает за передачу служебной информации, предоставляет приложениям информацию об ошибках и формирует запросы к уровню представления. Пример: HTTP, POP3, SMTP.
* **Уровень представления, Presentation layer** - 6-й уровень отвечает за преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с уровня приложений, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. На уровне представления может осуществляться сжатие/распаковка или кодирование/декодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.
* **Сеансовый уровень, Session layer** - 5-й уровень модели отвечает за поддержание сеанса связи, что позволяет приложениям взаимодействовать между собой длительное время. Сеансовый уровень управляет созданием/завершением сеанса, обменом информацией, синхронизацией задач, определением права на передачу данных и поддержанием сеанса в периоды неактивности приложений. Синхронизация передачи обеспечивается помещением в поток данных контрольных точек, начиная с которых возобновляется процесс при нарушении взаимодействия.
* **Транспортный уровень, Transport layer** - 4-й уровень модели, предназначен для доставки данных без ошибок, потерь и дублирования в той последовательности, как они были переданы. При этом неважно, какие данные передаются, откуда и куда, то есть он предоставляет сам механизм передачи. Блоки данных он разделяет на фрагменты, размер которых зависит от протокола, короткие объединяет в один, а длинные разбивает. Протоколы этого уровня предназначены для взаимодействия типа точка-точка. Пример: TCP, UDP.
* **Сетевой уровень, Network layer** - 3-й уровень сетевой модели OSI, предназначен для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию, отслеживание неполадок и заторов в сети. На этом уровне работает такое сетевое устройство, как маршрутизатор.
* **Канальный уровень, Data Link layer** - этот уровень предназначен для обеспечения взаимодействия сетей на физическом уровне и контроля за ошибками, которые могут возникнуть. Данные, полученные с физического уровня, он упаковывает во фреймы, проверяет на целостность, если нужно исправляет ошибки и отправляет на сетевой уровень. Канальный уровень может взаимодействовать с одним или несколькими физическими уровнями, контролируя и управляя этим взаимодействием. Спецификация IEEE 802 разделяет этот уровень на 2 подуровня - MAC (Media Access Control) регулирует доступ к разделяемой физической среде, LLC (Logical Link Control) обеспечивает обслуживание сетевого уровня. На этом уровне работают коммутаторы, мосты. В программировании этот уровень представляет драйвер сетевой платы, в операционных системах имеется программный интерфейс взаимодействия канального и сетевого уровней между собой, это не новый уровень, а просто реализация модели для конкретной ОС. Примеры таких интерфейсов: ODI, NDIS.
* **Физический уровень, Physical layer** - самый нижний уровень модели, предназначен непосредственно для передачи потока данных. Осуществляет передачу электрических или оптических сигналов в кабель или в радиоэфир и соответственно их приём и преобразование в биты данных в соответствии с методами кодирования цифровых сигналов. Другими словами, осуществляет интерфейс между сетевым носителем и сетевым устройством. На этом уровне работают концентраторы (хабы), повторители (ретрансляторы) сигнала и медиаконверторы. Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.



В основном используются протокол TCP/IP

**Определение:**

Transmission Control Protocol/Internet Protocol, TCP/IP (Протокол управления передачей/Протокол Интернета)

Большинство операционных систем сетевых серверов и рабочих станций поддерживает TCP/IP, в том числе серверы NetWare, все системы Windows, UNIX, последние версии Mac OS, системы OpenMVS и z/OS компании IBM, а также OpenVMS компании DEC. Кроме того, производители сетевого оборудования создают собственное системное программное обеспечение для TCP/IP, включая средства повышения производительности устройств. Стек TCP/IP изначально применялся на UNIX-системах, а затем быстро распространился на многие другие типы сетей.

**Маска подсети** — [битовая маска](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BC%D0%B0%D1%81%D0%BA%D0%B0) для определения по [IP-адресу](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) адреса подсети и адреса узла (хоста, компьютера, устройства) этой подсети. В отличие от IP-адреса маска подсети не является частью [IP-пакета](https://ru.wikipedia.org/wiki/IP#Пакет).

Благодаря маске можно узнать, какая часть [IP-адреса](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) [узла сети](https://ru.wikipedia.org/wiki/%D0%A3%D0%B7%D0%B5%D0%BB_%D1%81%D0%B5%D1%82%D0%B8) относится к адресу сети, а какая — к адресу самого узла в этой сети.

Например, узел с IP-адресом 12.34.56.78 и маской подсети 255.255.255.0 находится в сети 12.34.56.0 с длиной префикса 24 бита. В случае адресации [IPv6](https://ru.wikipedia.org/wiki/IPv6) адрес 2001:0DB8:1:0:6C1F:A78A:3CB5:1ADD с длиной префикса 32 бита (/32) находится в сети 2001:0DB8::/32.

Другой вариант определения — это определение подсети IP-адресов. Например, с помощью маски подсети можно сказать, что один диапазон IP-адресов будет в одной подсети, а другой диапазон соответственно в другой подсети.

Чтобы получить адрес сети, зная IP-адрес и маску подсети, необходимо применить к ним операцию [поразрядной конъюнкции](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82%D0%BE%D0%B2%D1%8B%D0%B5_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8#И) (побитовое И). Например, в случае более сложной маски (битовые операции в IPv6 выглядят аналогично):

IP-адрес: 11000000 10101000 00000001 00000010 (192.168.1.2)

Маска подсети: 11111111 11111111 11111110 00000000 (255.255.254.0)

Адрес сети: 11000000 10101000 00000000 00000000 (192.168.0.0)

Под маской подсети понимают 32-разрядное число, составленное из единиц и нулей. Начинается маска из последовательности единиц, а завершается последовательностью нулей. Ее накладывают на [IP-адрес](https://tvoi-setevichok.ru/lokalnaya-set/kakie-ip-adresa-mozhno-ispolzovat-v-lokalnoy-seti.html). Ту часть адреса, на которую накладываются единицы, определяют адресом сети. На остальную часть накладываются нули — она отводится под адресацию хостов.

## Расчет маски подсети

Сетевой адрес составлен из двух частей — адреса сети и хоста. До появления масок специалисты применяли методы классового разделения сетей. Но число хостов в сети стало очень велико, а число выделяемых для них адресов сетей оказалось сильно ограниченным. Поэтому понадобилась дополнительная идея, которая была воплощена в маске. Она позволила в разных классах сетей выделить множество подсетей с разным количеством хостов.

Администратор сети, получив в распоряжение некий сетевой адрес, имеет возможность [разделить его на ряд подсетей](https://tvoi-setevichok.ru/korporativnaya-set/delenie-na-podseti-razdelenie-lokalnoy-seti-s-pomoshhyu-vlan.html) (а может использовать и без разделения). Зачем делить полученный адрес? В разных сетях нужно подключать различное число компьютеров — где-то надо подключить только 10 хостов, а где-то более 30.

Будет гораздо удобнее, если эти «количества» будут подключены в разных подсетях с общением через маршрутизатор.

Например, определим маску для сети класса С. Из соглашения известно, что под адрес сетей такого класса отводят первый, второй и третий байты 32-разрядного числа. Четвертый остается для распределения хостов. Тогда запись маски в точечно-двоичной нотации выглядит так:

11111111.11111111.11111111.00000000

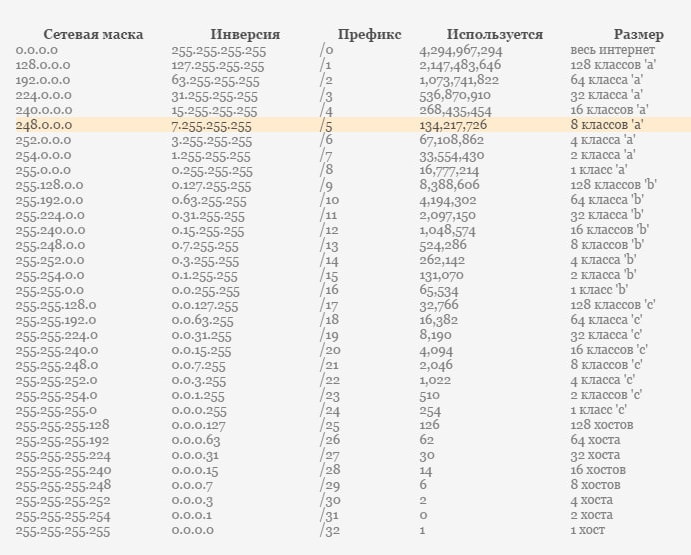
Как видим, первые 24 бита установлены, а последние 8 сброшены. Таким образом, маска в десятичном формате получит такой вид: 255.255.255.0. Идентичной записью станет следующая /24 — префиксная.

## https://cf.ppt-online.org/files1/slide/q/qw9NJgyE6mvhRfdzKL74b28VBlMcGQFAi3xtjY/slide-18.jpg

## Таблица масок подсетей

Маска позволяет выделить целое множество сетей класса С, как и сетевых адресов других типов. В предыдущем примере была показана маска для стандартной сети класса С. Однако если сбросить крайнюю единицу на ноль, тогда получим следующую запись 255.255.254.0 или /23. При такой маске можем получить 2 сети класса С, так как сброшенная единица может быть восстановлена. Запись с 17-ю единицами позволит адресовать сразу 128 сетей класса С.

С целью облегчения понимания бесклассовой адресации (CIDR) создаются целые таблицы соответствия префиксов, масок, количества подключаемых хостов и классов сетей. Сетевому администратору нет нужды рассчитывать маски, число сетей и хостов самостоятельно. Достаточно только заглянуть в список соответствия, чтобы ответить на вопрос какую маску выбрать при необходимости подключить конкретное число рабочих станций.



Так, если администратору надо подключить 30 рабочих станций, тогда маска сети должна завершаться 5-ю нулями. Действительно, для нумерации узлов достаточно 5 нулей, так как 2 в степени 5 равно 32.

При этом узел с пятью нулями отвечает за номер сети, а узел с 5-ю единицами является широковещательным. Соответственно три старшие бита должны заполняться единицами, как и три предшествующих байта, поэтому маска должна принять вид:

1111111.11111111.11111111.11100000 или 255.255.255.224.

Вместо вычислений администратор может воспользоваться данными из таблиц соответствий.

**Заключение**

В этом реферате я рассмотрел 2 протокола сети и метод расчета маски подсети.