Министерство образования и науки Российской Федерации

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

**Кафедра теоретических основ радиотехники (ТОР)**

**В.Г. Козлов, Е.С. Семигук,**
**С.И. Богомолов**

# ПРОГРАММНЫЕ СРЕДСТВА СИСТЕМ СВЯЗИ

**Учебное пособие**

**2008**

Рецензент: заведующий кафедрой ТОР ТУСУРа,
д-р техн. наук, профессор А.В. Пуговкин

Корректор: Осипова Е.А.

**Козлов В.Г., Семигук Е.С., Богомолов С.И.**
Программные средства систем связи: Учебное пособие. — Томск: Томский межвузовский центр дистанционного образования, 2008. — 162 с.

# ОГЛАВЛЕНИЕ

# ПРЕДИСЛОВИЕ

Жизнь в современном обществе сложно представить без средств связи. С каждым годом растут объемы передаваемой информации, развивается сеть Internet, расширяются предоставляемые услуги по использованию средств связи, появляются новые технологии. Для передачи информации используются различные сети.

Одновременно с развитием средств передачи информации шло и развитие протоколов передачи данных. В данном пособии основное внимание уделено стеку протоколов TCP/IP.

TCP/IP — это множество коммуникационных протоколов, которые определяют, как компьютеры различных типов могут общаться между собой. Термин TCP/IP обычно обозначает все, что связано с протоколами TCP и IP. Он охватывает целое семейство протоколов, прикладные программы и даже саму сеть. В состав семейства входят протоколы UDP, ARP, ICMP, FTP и многие другие. TCP/IP — это технология межсетевого взаимодействия и стандарт стека протоколов, на котором основана работа глобальной сети Internet.

Простое объяснение работы протокола TCP/IP примерно звучит так: Internet используются два основных протокола: межсетевой протокол IP и протокол управления передачей — TCP.

Протокол IP — это протокол, описывающий формат пакета данных, передаваемого по сети. Протокол TCP предназначен для контроля передачи, контроля целостности передаваемой информации. Протокол UDP обеспечивает передачу информации, то есть в целом повторяет функции TCP, но не обеспечивает той надежности доставки данных, которую гарантирует использование протокола TCP.

Протокол ARP применяется для связи сетевого уровня и физического, а точнее для сопоставления IP-адреса устройства с его MAC-адресом.

Протокол ICMP отрабатывает ошибки, возникающие в сети и сообщает о проблемах с передачей данных отправителю.

Основное предназначение протокола FTP — это пересылать (копировать, передавать) файлы. Глобальная сеть заключает в себе огромные информационные ресурсы и этот протокол делает доступным большую часть программного фонда Internet.

# 1 ОБЩАЯ МОДЕЛЬ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ

Информационные технологии к настоящему времени прошли четыре основных этапа:

1. Внедрение и эксплуатация больших ЭВМ.
2. Внедрение ЭВМ малого и среднего класса (мини ЭВМ).
3. Персональные компьютеры.
4. Революционное развитие сетевых технологий.

На ранней стадии 4-го этапа пользователи были вполне удовлетворены возможностями базовых сетевых технологий Ethernet, Token Ring, X25.

Дадим некоторые определения:

**Базовая сетевая технология** — согласованный набор протоколов и реализующих их программно-аппаратных средств, достаточный для построения вычислительной сети.

**Протокол** — формат описания передаваемых сообщений и правила, по которым происходит обмен информацией между двумя или несколькими системами.

**Сеть** — это:

1) соединение группы узлов (компьютеров или других устройств);

2) группа точек, узлов или станций, соединенных коммуникационными каналами и набор оборудования, обеспечивающего соединение станций и передачу между ними информации.

Проблемы возникли при появлении новых технологий (Fast Ethernet, FDDI и др.) — степень неоднородности при объединении локальных сетей существенно повысилась по следующим причинам:

– отсутствие стандарта или его неточная реализация;
– попытки использования фирменных стандартов;
– попытки самостоятельного улучшения существующих стандартов.

Для упрощения соединений между большим числом сетей разного типа важно иметь единую, стандартную и независимую от поставщика среду.

Определение. **Совместимое оборудование** (СО) — оборудование, работающее по единым правилам.

Применение СО облегчает задачу внедрения в единую систему разнообразного оборудования, а также позволяет администрации выбирать подходящее оборудование произвольного производителя.

Все эти причины привели к необходимости разработки базовой модели сетевого взаимодействия. В 1984 году с целью упрощения и стандартизации взаимодействия устройств в сетях Международная организация по стандартизации (International Organization for Standardization — ISO) предложила семиуровневую эталонную коммуникационную модель Взаимодействие Открытых Систем (ВОС) (Open System Interconnection — OSI).

Модель ВОС как единый комплекс стандартов является основной для взаимной совместимости оборудования и программ различных поставщиков. Модель ВОС разбивает большую задачу движения информации в сети на 7 относительно автономных задач, более мелких и обеспечивающиеся соответствующими уровнями.

Каждый уровень выполняет свою часть функций, необходимых для установления соединения с парным ему уровнем. В то же время он выполняет определенную обработку данных, реализуя набор услуг для вышележащего уровня.

Следует заметить, что в реальной жизни некоторые реализации пропускают один или несколько уровней. Два самых низких уровня модели ВОС реализуются программно-аппаратными средствами. Верхние пять — в основном программными.

1. Физический уровень — передача битов по физическим каналам (кабель, сетевая карта).

2. Канальный уровень — обеспечивает передачу кадров между любыми двумя узлами сети (участвуют аппаратные адреса). На этом уровне определяют доступность среды передачи, разбивается поток информации (поступающей из физического уровня) на кадры. Обеспечивается корректность передачи путем вычислением контрольной суммы кадра.

3. Сетевой уровень — обеспечивает доставку данных между любыми двумя узлами сети с произвольной топологией. Управление сетью состоит в выборе маршрута передаче сообщения по линиям, соединяющим эти узлы. На этом уровне формируются пакеты.

4. Транспортный уровень — обеспечивает передачу данных между любыми узлами сети с определенной надежностью (то есть если какой-то фрагмент пакета не дошел до узла назначения, то повторная передача недошедших фрагментов будет обеспечена средствами этого уровня).

5. Сеансовый — предоставляет средства управления диалогом. Например, аутентификация и проверка полномочий. На этом уровне по запросам в сети создаются порты для приема и передачи сообщений и организуются соединения — логические каналы.

6. Уровень представления — определяет преобразование внешних данных в необходимый для передачи вид (перекодировка символов из одного алфавита в другой).

7. Прикладной уровень — определяет набор различных сетевых сервисов на уровне конечного пользователя.

Инкапсуляция данных это процесс, в котором информация «обертывается» заголовками протокола, или, другими словами, помещается в поле данных другого протокола.

В модели OSI каждый уровень производит инкапсуляцию данных вышестоящего уровня по мере того, как данные перемещаются вниз по стеку протоколов.

На передающей стороне инкапсуляция данных происходит следующим образом:

1. Пользовательские данные преобразуются в Данные. (Уровень Приложения).

2. Данные преобразуются в Сегменты. (Транспортный Уровень).

3. Сегменты преобразуются в Пакеты или Дейтаграммы. (Сетевой Уровень).

4. Пакеты (Дейтаграммы) преобразуются в Кадры (Фреймы). (Канальный Уровень).

5. Кадры (Фреймы) преобразуются в Биты. (Физический Уровень).

На рисунке 1.1 показан пример переноса данных с помощью модели ВОС.

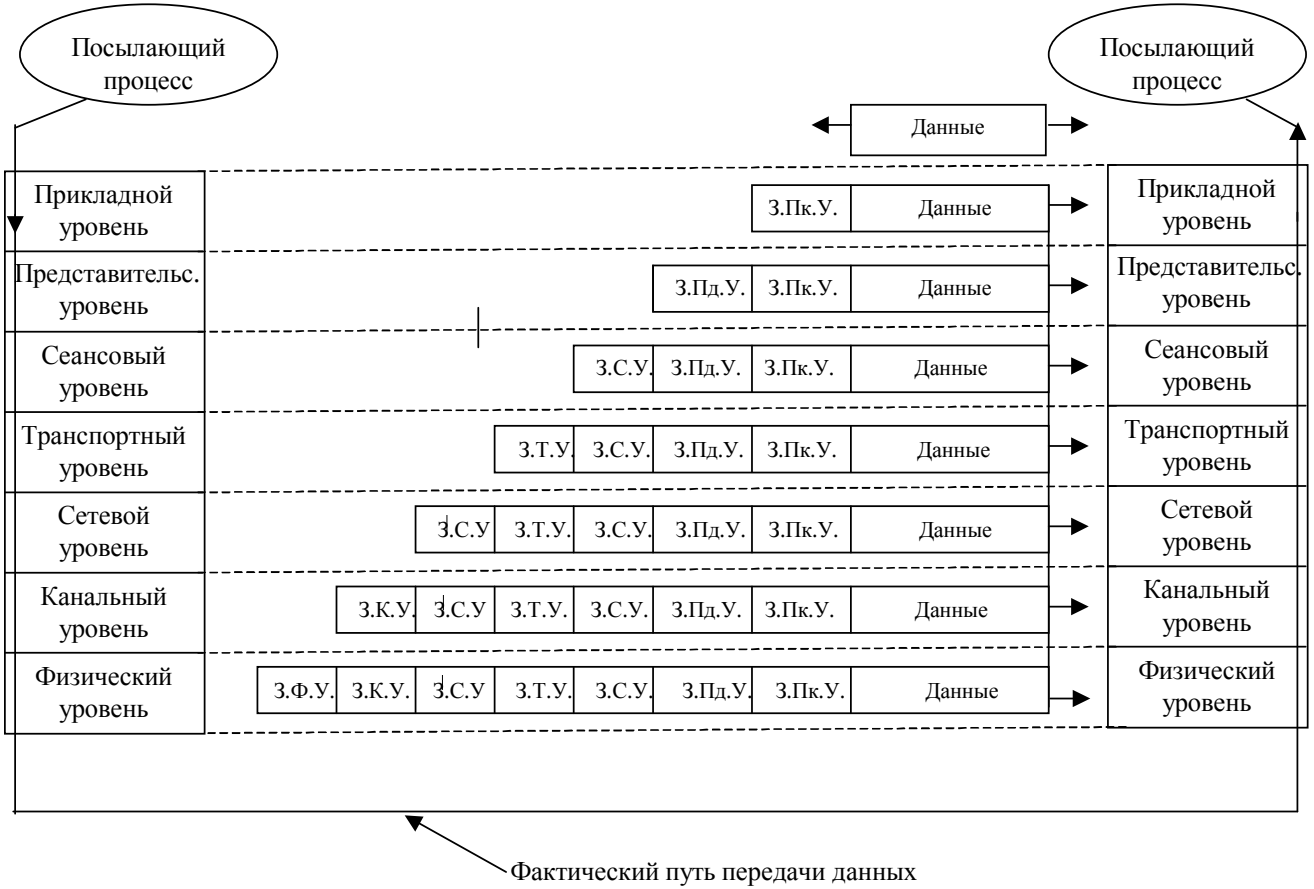



Рисунок 1.1 — Модель взаимодействия открытых систем ВОС (OSI)

Модель ВОС описывает путь информации через сетевую среду от прикладной программы одного компьютера до другой, находящейся (выполняющейся) на другом компьютере. Информация проходит сначала через все уровни вниз на первой системе, пока не достигнет физического уровня и не будет передана в физическую среду. По мере прохождения уровней информация преобразуется путем присоединения служебной информации — заголовков (на рисунке 1.1 это З.Ф.У. — заголовок физического уровня, З.К.У. — заголовок канального уровня и др.), которая интерпретируется только тем уровнем, для которого она предназначена, в соответствии с его протоколом. После получения информации она проходит через все уровни системы наверх, к прикладному, по пути преображаясь в первоначальный вид. Заголовок для нижнего уровня является просто данными.

Создается впечатление, что каждый уровень общается с равным ему уровнем на удаленной системе. Таким образом, взаимодействие между удаленными системами представляется в виде нескольких логических каналов, соответствующих уровням модели, передача данных в каждом из которых определяется протоколом своего уровня. Сообщение, проходящее через уровни, делится на заголовок и информативную часть (концепция вложенных заголовков).

Модель ВОС не является реализацией сети (хотя существуют протоколы, полностью удовлетворяющие уровням). Она определяет функции каждого уровня и дает представление о движении данных в сети. Определим стек протоколов как набор стандартизованных коммуникационных протоколов, реализующий принцип уровневого деления модели OSI/ISO.

Широко известны, например, стеки IPX/SPX, TCP/IP, ISO.

Перед рассмотрением реального списка протоколов рассмотрим принцип взаимодействия «клиент — сервер», как наиболее часто встречающийся тип сетевого взаимодействия в настоящее время. «Клиент — сервер» — общий способ описания услуг и модель пользовательских процессов (программ) для потребления этих услуг.

Смысл этой модели заключается в том, что один из взаимодействующих через сетевую среду процессов является сервером, то есть представляющим вполне определенные услуги (например, доступ к файлам данного компьютера), а другой, как правило, прикладная программа — «клиент» (Web-браузер и тому подобное).

Программы-серверы можно разделить на два класса:

1. Итеративные (iterative).

2. Конкурентные (concurrent).

Модель работы итеративного сервера:

1. Ожидание запроса от клиента.

2. Обработка запроса от клиента (в то время другие клиенты не обслуживаются).

3. Передача результата обработки запроса запросившему клиенту.

4. Возврат в состояние 1.

Модель работы конкурентного сервера:

1. Ожидание запроса от клиента.

2. Запуск нового экземпляра сервера для обработки запроса от данного клиента. Новый экземпляр может быть запущен как новый процесс, нить, в зависимости от возможностей данной ОС. Новый сервер полностью обрабатывает запрос клиента. По завершению обработки он завершается.

3. Возврат в состояние 1.

Каждый клиент при этом работает со «своим» сервером. В это время принимаются запросы от других клиентов. ОС должна быть многозадачной. Заметим, что клиент не сообщает с каким типом сервера он работает. Как правило UDP-серверы — итеративные, TCP — конкурентные. Ниже на рисунках 1.2 и 1.3 приведены алгоритмы работы итеративного и конкурентного серверов.

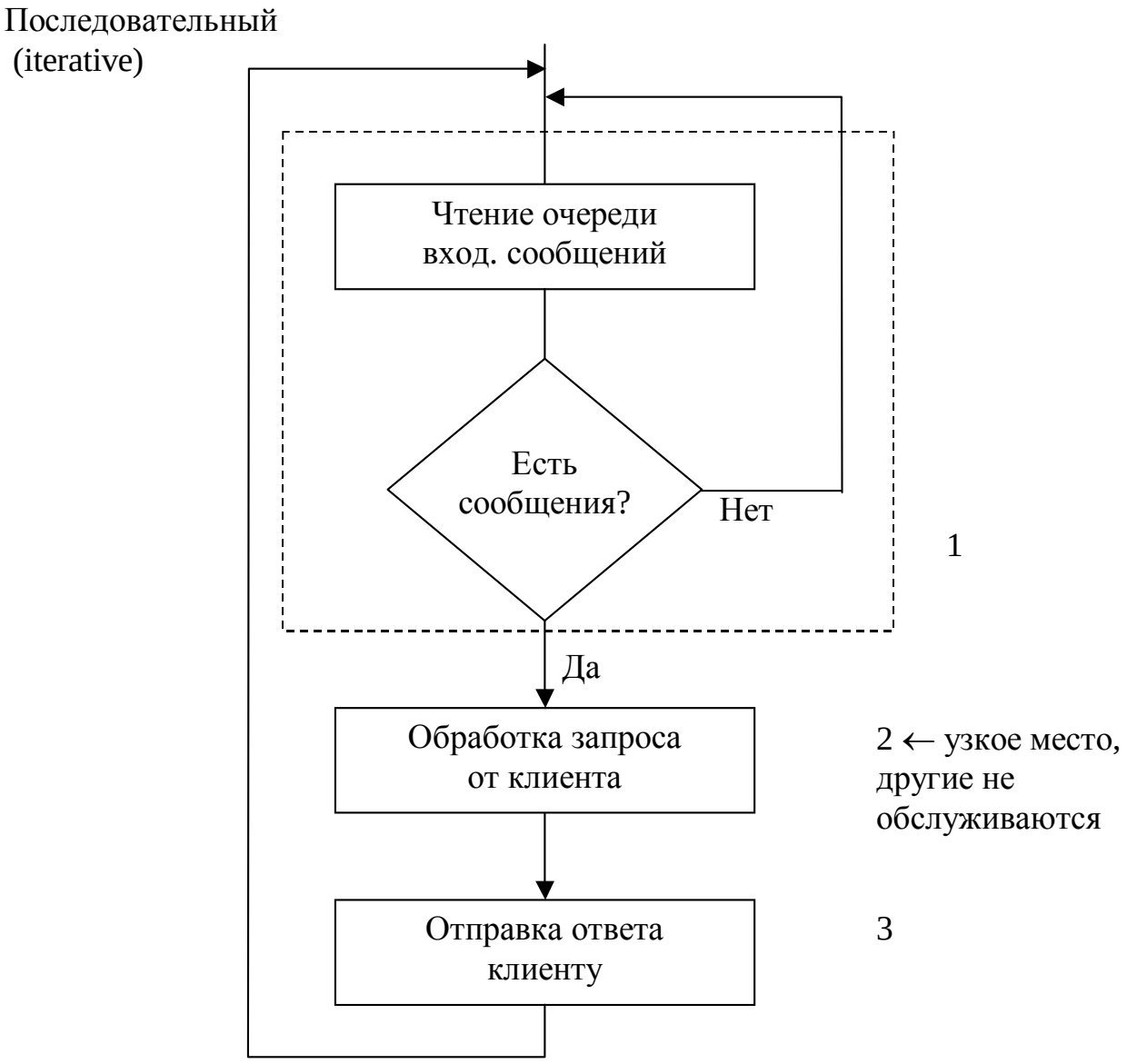



Рисунок 1.2 — Алгоритм работы итеративного сервера

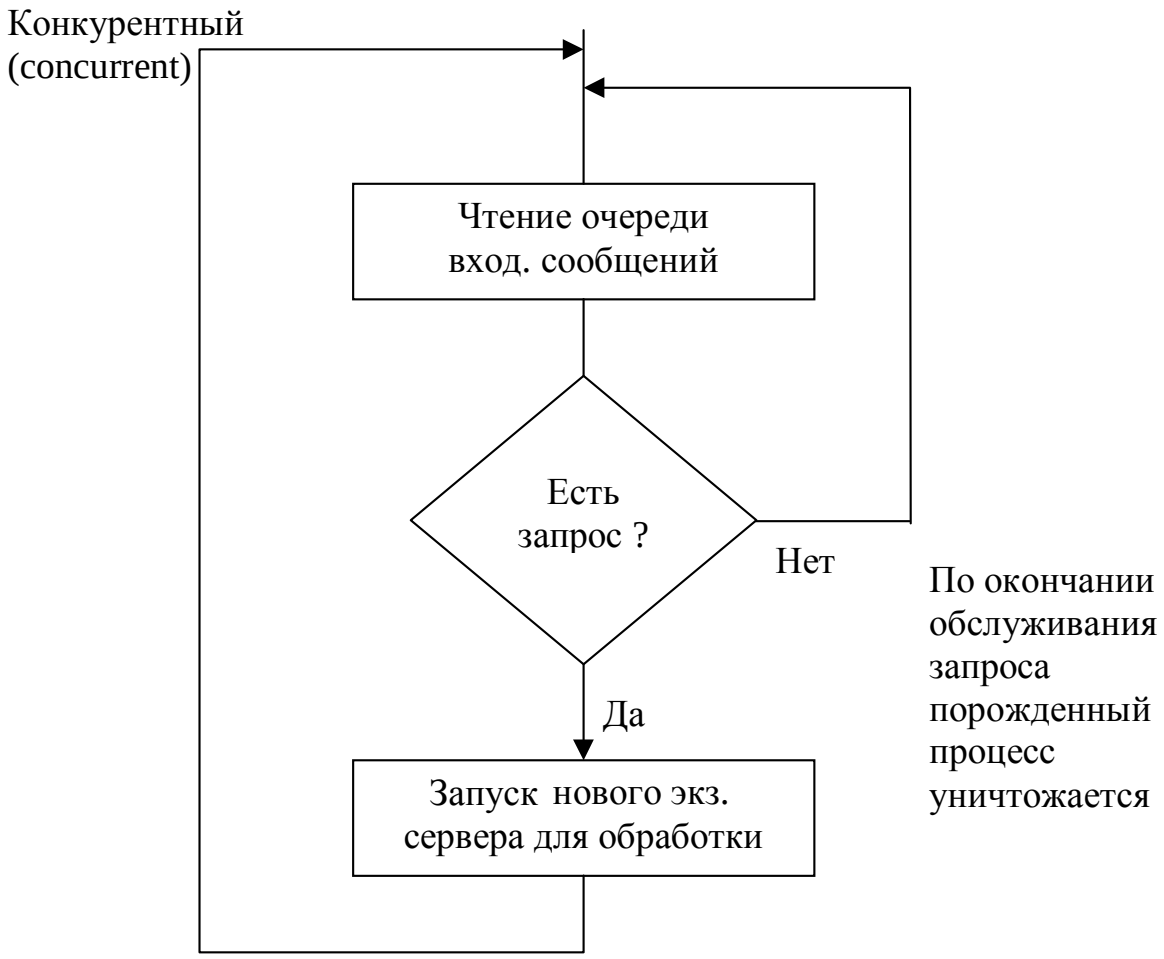



Рисунок 1.3 — Алгоритм работы конкурентного сервера

# 2 СЕМЕЙСТВО ПРОТОКОЛОВ TCP/IP

**Краткая история**

В июле 1968 года DARPA (Defense Advanced Research Projects Agency — оборонное управление перспективного планирования научно-исследовательских работ) Министерства Обороны США начало финансирование проекта по созданию экспериментальной компьютерной сети коммутации пакетов (packet switching network). Сеть была названа ARPANET и связывала компьютеры исследовательских лабораторий. Над проектом работала фирма «Bolt, Beranek and Newman». В декабре 1969 года произошел запуск сети из 4-х узлов: Стэндфордский исследовательский институт, Калифорнийский университет в Санта-Барбаре, Калифорнийский университет в Лос-Анджелесе и университет Юты. Успех сети привел к ее быстрому росту. По мере развития сети были разработаны необходимые коммуникационные протоколы. Некоторые заметные этапы создания Internet.

1973 г. — разработчики объединили отдельные сети в проект «Internetting Problem».

1981 г. — NSF (National Science Foundation — национальный научный фонд США) одобрил создание NSFnet.

1983 г. — 400 соединенных компьютеров, стандартизованы TCP, IP. DARPA начала финансировать Калифорнийский университет в Беркли по поддержке TCP/IP в UNIX.

1984 г. — ARPANET делится на MILNET (для военных целей) и ARPANET (для мирных целей).

1986 г. — NSF создал опорную сеть 56 кб/с (6 суперкомпьютерных центров США) — NFSnet создана.

1990 г. (июнь) — ARPANET прекратила существование, NSFnet стала опорой Internet.

1995 г. — 4.852.000 компьютеров (из них 3.000.000 в США) в сети.

1996 г. — 13.000.000 компьютеров.

1997 г. — 16.146.000 компьютеров.

## 2.1 Основные достоинства TCP/IP

1. Это семейство основано на свободно доступных открытых форматах, разработанных независимо от конкретного оборудования или ОС. Именно поэтому TCP/IP является наиболее рас-

пространенным средством объединения разнородного оборудования и программного обеспечения.

2. Применение TCP/IP не зависит от конкретного оборудования физического уровня. Это позволяет использовать TCP/IP в физических сетях разного типа (Ethernet, Token Ring, X25 и др).

3. TCP/IP имеет гибкую систему адресации, позволяющую любому устройству в сети однозначно адресовать другое устройство сети. Одна и та же система адресации может использоваться как в локальных, так и в территориальных распределенных сетях, включая Internet.

4. В семейство TCP/IP входят стандартизованные протоколы высшего уровня для поддержки прикладных сетевых услуг: передача файлов, электронная почта, удаленный доступ и др.

К настоящему моменту семейство TCP/IP в целом сформировано и является основой современного Интернета.

Пользователей и разработчиков программного обеспечения в большей степени интересуют протоколы и сервисы высшего уровня. Этот уровень активно развивается, появляются новые версии существующих протоколов, разрабатываются совершенно новые протоколы.

## 2.2 Архитектура TCP/IP

Для ясности изложения приведем ниже соответствие уровней стека протоколов BOC (OSI) и стека TCP/IP (рис. 2.1).

<table>
<tr><td>Уровни модели BOC(OSI)</td><td colspan="6"></td><td>Уровни стека TCP/IP</td></tr>
<tr><td>7</td><td rowspan="2">WWW, WAIS</td><td rowspan="2">SNMP</td><td rowspan="2">FTP</td><td rowspan="2">Telnet</td><td rowspan="2">SMTP</td><td rowspan="2">TFTP</td><td rowspan="2">1</td></tr>
<tr><td>6</td></tr>
<tr><td>5</td><td colspan="5" rowspan="2">TCP</td><td rowspan="2">UDP</td><td rowspan="2">2</td></tr>
<tr><td>4</td></tr>
<tr><td>3</td><td colspan="2">IP</td><td>ISMP</td><td>RIP</td><td colspan="2">OSPF</td><td>3</td></tr>
<tr><td>2</td><td colspan="6" rowspan="2">Не регламентируется<br>Ethernet, Token Ring, FDDI, X.25, SPIP,PPP</td><td rowspan="2">4</td></tr>
<tr><td>1</td></tr>
</table>

Рисунок 2.1 — Соответствие уровней стека протоколов BOC (OSI) и TCP/IP

Архитектура TCP/IP основана на представлении, что коммутационная инфраструктура включает 3 объекта:

– процессы;

– хосты (машины, входящие в сеть, предназначенные для выполнения программ пользователя);

– сеть.

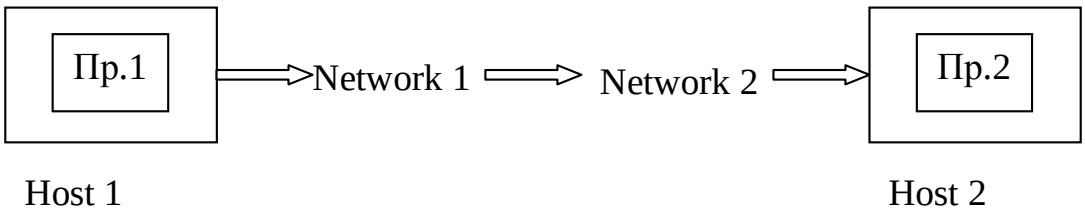



Рисунок 2.2 — Передача сообщения через сеть

Процессы — основные коммуникационные объекты, поскольку, в конечном счете, именно между ними осуществляется обмен информацией. Чтобы доставить данные процессу, их сначала нужно передать нужному хосту, а затем уже целевому процессу. Более того, эти 2 фазы могут выполняться независимо. Таким образом, от коммуникационной инфраструктуры требуется обеспечение иерархии и доставки данных между хостами, а хосты, в свою очередь, обязаны доставить данные конкретному процессу.

Архитектуру TCP/IP составляют 4 уровня.

| | |
|---|---|
| 1 | Уровень приложений (Application/process layer) |
| 2 | Транспортный уровень (Host-to-host layer) |
| 3 | Уровень Internet (Internet layer) |
| 4 | Уровень сетевого интерфейса (Network interface layer) |

Ознакомимся подробнее с каждым из уровней.

4. Уровень сетевого интерфейса — передача данных между коммуникационными узлами, подключенными к одному и тому же сетевому сегменту (Ethernet, PPP). Формально протоколы этого уровня не являются частью стека TCP/IP, т. к. не определены ни стандартами Минобороны США, ни стандартами Internet. Вместо этого используются существующие сетевые протоколы и определяются методы передачи TCP/IP трафика с помощью кон-

кретной коммуникационной технологии (например, RFC894 — Standard for Transmission of Datagrams over Ethernet Network).

3. Уровень Internet или уровень межсетевого взаимодействия — основа архитектуры стека протоколов TCP/IP, который реализует передачу пакетов в режиме без установления соединения, то есть дейтограммным способом. Передача пакетов идет по маршруту, который в данный момент наиболее рационален.

Основным для этого уровня является протокол IP (является дейтаграммным), к этому уровню относятся все протоколы, связанные с составлением и модификацией таблиц маршрутизации. Например, RIP (Routing Internet Protocol), OSPF (Open Shortest Path First) — протоколы сбора маршрутной информации. Также протокол межсетевых управляющих сообщений ICMP (Internet Control Message Protocol) — предназначен для обмена информацией об ошибках между маршрутизаторами сети и узлом-источником пакетов. С помощью специальных пакетов ICMP сообщает о невозможности доставки пакета, о превышении времени жизни, об изменении маршрута пересылки и др.

2. Транспортный уровень — решает задачу обеспечения надежной информационной связи между 2-мя конечными узлами сети.

На этом уровне функционирует протокол управления передачей TCP/IP (Transmission Control Protocol) и протокол дейтаграмм пользователя UDP (User Data gram Protocol). Протокол TCP обеспечивает надежную передачу сообщений благодаря образованию логических соединений. Он разделяет поток пакет на части — сегменты и передает их нижнему уровню. После того, как эти сегменты будут доставлены в пункт назначения, протокол TCP снова соберет их в непрерывный поток данных — переданное сообщение.

1. Прикладной уровень объединяет все службы, предоставляемые системой пользовательским приложением. Прикладной уровень реализуется программными системами, построенными в архитектуре «клиент-сервер». Эти протоколы «занимаются» деталями конкретного приложения и «не интересуются» способами передачи данных по сети. Этот уровень постоянно расширяется, к старым сетевым службам типа Telnet, FTP, TFTP, DNS, SNMP при-

соединяются сравнительно новые службы, например, HTTP — протокол передачи гипертекстовой информации.

Уровни адресации:

1. Физический — MAC адреса.
2. IP-адреса — адреса на сетевом уровне.
3. Номера портов — идентифицируют конкретные процессы.

Работа модулей TCP/IP напоминает сборочный конвейер: каждый модуль выполняет определенную для него задачу, полагаясь на качество работы, выполненной на предыдущем этапе.

PDU — Protocol Data Unit состоит из данных, переданных верхним модулем и заголовка со служебной информацией, распознаваемым модулем того же уровня.

## 2.3 Стандартизация

Международный союз телекоммуникаций (ITU — International Organization for Union) был создан в 1865 г. Задачей этого союза была стандартизация международных средств связи, что в те дни означало телеграф. *Уже тогда было ясно, что если половина стран будет использовать азбуку Морзе, а другая половина какой-нибудь другой код, то возникнут проблемы.* С появлением международной телефонной связи международный союз телекоммуникаций (МСТ) занялся также разработкой стандартов телефонии.

МСТ (ITU) состоит из 3-х главных секторов:

1. Сектор радиосвязи (ITU-R).
2. Сектор стандартизации телекоммуникаций (ITU-T).
3. Сектор развития (ITU-D).

*Следует заметить, что рекомендации ITU-T технически являются лишь предложениями, которые правительства любой страны могут принять, либо проигнорировать. На практике никто не может любой стране помешать принять телефонный стандарт, отличный от всего остального мира, однако тем самым эта страна отрежет себя от всего остального мира.*

Международные стандарты разрабатываются также Международной организацией по стандартизации (ISO — International Organization for Standardization), добровольной организацией,

созданной в 1946 г. В нее входят национальные организации по стандартизации 89 стран.

В области телекоммуникационных стандартов ISO и ITU-T часто сотрудничают (ISO является членом ITU-T), чтобы не допустить появления двух несовместимых международных стандартов.

В области стандартов Интернета работает Совет по архитектуре Интернета (IAB — Internet Architecture Board). Стандарты IAB оформляются в виде технических отчетов, называемых RFC (Requests for Comments). RFC доступны в Интернете для всех желающих. На сегодня существует около 2000 этих документов.

Стандарты TCP/IP всегда публикуются в виде RFC, но не все RFC определяют стандарты.

Содержание RFC делится на 2 части:

1. Состояние стандартизации:
– стандарт на уровне утверждения;
– предназначается к рассмотрению;
– экспериментальный протокол;
– протокол устарел и не используется.
2. Статус протокола:
– требуется для внедрения;
– рекомендуется для внедрения;
– может внедряться производителями по выбору;
– не рекомендуется для внедрения.

# 3 СЕТЕВОЙ УРОВЕНЬ СТЕКА TCP/IP

## 3.1 Структура пакета протокола IP

Протокол IP (Internetwork Protocol) обеспечивает доставку дейтаграммы от источника к получателю через систему связанных между собой сетей.

Сам протокол не исправляет ошибки, а только сообщает об ошибках в исходящие хост-компьютеры. Контроль за правильностью переданных данных осуществляют более высокие уровни стека. Функции: адресация, фрагментация/дефрагментация, маршрутизация данных.

Дейтаграмма состоит из заголовка и данных протоколов верхних уровней.


Рисунок 3.1

Каждая дейтаграмма обрабатывается как самостоятельный объект, не зависящий от других дейтаграмм.

Поля IP заголовка представлены на рисунке 3.2.

- **Номер версии** (Version = 4 (4 bit для протокола IP V4)) указывает номер версии.
- **Длина заголовка** (Header length, 4 bit) длина заголовка в 32 битных словах меняется в зависимости от числа параметров.
- Поле **Тип сервиса** (TOS, 8bit) задает вид критерия выбора маршрута. С помощью установки бит данного поля в качестве критерия может быть задана либо длина маршрута, либо надежность, либо сбалансированность трафика.
- Поле **Общая длина (**Total length in bytes, 16 bit (65535 байт max)) содержит измеряемую в байтах суммарную длину дейтаграммы, включая длину IP-заголовка и данных.

| Версия (4) **Version** | Длина (4) **Header length** | Тип сервиса (8) **TOS** | Общая длина (16) **Total length in bytes** | |
|---|---|---|---|---|
| Идентификатор (16) **Identification** | | | Флаги (3) **Flags** | Смещение фрагмента (13) **Fragment offset** |
| Время жизни (8) **TTL** | | Протокол (8) **Protocol** | Контрольная сумма заголовка (16) **Header Checksum** | |
| Адрес IP-источника (32) **Source Address** | | | | |
| Адрес IP-назначения (32) **Destination Address** | | | | |
| Опции IP (необязательно) (32) **Options** | | | | Выравнивание |
| Данные (32) | | | | |

Рисунок 3.2 — Поля IP заголовка

- Поле **Идентификатор** (Identification, 16 bit) представляет собой уникальный номер, характеризующий конкретную дейтаграмму. Используется для распознавания продублированных пакетов, а также при распознавании пакетов, являющихся результатом фрагментации исходного пакета.
- Поле **Флаг** (Flags 3 bit) содержит 3 бита: первый бит этого поля имеет значение 0, второй определяет, разрешена или нет фрагментация блока данных.
- Поле **Смещение фрагмента (**Fragment offset 13 bit) указывает положение данного фрагмента в дейтаграмме. Используется при сборке/разборке фрагментов пакетов при передачах их между сетями с различными величинами MTU (длин кадров).
- Поле **Время жизни** (TTL, 8 bit) дейтограммы (верхнее значение периода времени нахождения дейтаграммы в сети max =255 секунд, 0 — уничтожается, 1hop — 1секунда).
- Поле **Протокол** (Protocol 8 bit) содержит указание, какой протокол следует за IP. Каждый протокол, относящийся к TCP/IP, идентифицируется фиксированным номером.

1 — ICMP (протокол сообщений управления Интернет).

6 — TCP (протокол управления передачей).

17 — UDP (протокол дейтограммы пользователя).

- Поле **Контрольная сумма заголовка** (Header Checksum, 16 bit) служит для проверки правильности информации заголовка дейтаграммы, подтверждает отличие от нуля поля «времени жизни».
- Поля **Адрес IP-источника** (Source Address, 32 bit) и **Адрес IP-назначения** (Destination Address, 32 bit) используются маршрутизаторами и шлюзами для маршрутизации блока данных в сети.
- Поле **Опции IP** (Options,32 bit) чаще всего отсутствует (безопасность, RR, SR и другие). Поле выравнивание дополняется нулями до 32-битной границы.

## 3.2 Типы адресов в сетях стека TCP/IP

Каждый компьютер в сети TCP/IP имеет адреса трех уровней:

- Символьный идентификатор-имя, например, 5ЕК.У1.1ВМ.COM. Этот адрес назначается администратором и состоит из 3-х частей: имя машины, имя организации, имя домена. Этот адрес используется на прикладном уровне. *Между доменным именем и IP-адресом узла нет никакого алгоритмического соответствия, поэтому необходимо использовать какие-то дополнительные таблицы или службы, чтобы узел сети однозначно определялся как по доменному имени, так и по IP-адресу. В сетях TCP/IP используется специальная распределенная служба Domain Name System (DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также DNS-именами.*
- IP-адрес, например, 109.26.17.100. Эти адреса состоят из 4 байт и представляют собой основной тип адресов, на основании которых сетевой уровень передает пакеты между сетями. IP-адрес назначается администратором во время конфигурирования компьютеров. Сетевой адрес может быть выбран произвольно, либо назначен по рекомендации специального подразделения Internet (NIC, а у нас Relcom), если сеть должна работать как составная часть Internet. IP-адрес состоит из двух частей: номера сети и номера узла. Обычно поставщики услуг Internet получают диапазоны адресов у подразделений NIC, а затем распределяют их между своими абонентами. *Маршрутизатор по определению входит сразу в несколько сетей. Поэтому каждый порт маршрути-*

*затора имеет собственный 1P-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.*

• Локальный адрес или адрес сетевого адаптера, например, 11-АО-17-ЗО-ВС-01. Этот адрес назначается производителем оборудования и является уникальным. Он имеет формат 6 байтов: 3 байта — номер фирмы производителя, 3 байта — порядковый номер сетевого адаптера данной фирмы. Адрес сетевого адаптера используется на канальном уровне и называется MAC-адресом. *Однако протокол IP может работать и над протоколами более высокого уровня, например, над протоколом IPX или X.25. В этом случае локальными адресами для протокола IP соответственно будут адреса IPX и X.25. Следует учесть, что компьютер в локальной сети может иметь несколько локальных адресов даже при одном сетевом адаптере. Некоторые сетевые устройства не имеют локальных адресов. Например, к таким устройствам относятся глобальные порты маршрутизаторов, предназначенные для соединений типа «точка-точка».*

## 3.3 Структура и типы IP-адресов

На рисунке 3.3 показана *структура IP-адреса*. Он состоит из 4 байтов и записывается в виде четырех чисел, содержащихся в каждом байте, разделенных точками, например:

128.10.2.30 — десятичная форма представления адреса,

10000000 00001010 00000010 00011 — двоичная форма.

Адрес состоит из двух логических частей — номера сети и номера узла в сети. Какая часть адреса относится к номеру сети, а какая к номеру узла, определяется значениями первых битов адреса:

• 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. (Номер 0 не используется, а номер 127 зарезервирован для специальных целей.) В сетях количество узлов должно быть больше $2^{16}$ и не превышать $2^{24}$.


Рисунок 3.3 — Структура IP-адреса

- Если первые два байта адреса равны 10, то сеть относятся к *классу B* и являются сетью средних размеров с числом узлов $2^8 – 2^{16}$. В сетях класса B под адрес сети и под адрес узла отводится по 16 битов.
- Если адрес начинается с последовательности 110, то это сеть *класса C,* в которой не может быть больше, чем $2^8$ узлов. Под адрес сети отводится 24 бита, а под адрес узла — 8 битов.
- Еще имеются два типа IP-адресов — адрес класса D, который начинается с последовательности 1110 и обозначает особый адрес — multicasting и адрес класса E, который начинается с кода 11110, он зарезервирован для будущих применений.

*До этого мы считали, что каждому узлу сети соответствует один IP-адрес. Однако как быть с маршрутизатором, который связывает несколько сетей? Легко представить и такую ситуацию, когда компьютер входит в несколько сетей. В этом случае сетевой узел должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.*

В протоколе IP существует несколько соглашений об особой интерпретации IP-адресов:

- если IP-адрес состоит только из двоичных нулей,

| 0000……..0000 |
|---|

то он обозначает тот узел, который сгенерировал этот пакет;

- если в поле номера сети стоят 0,

| 0000……0 | Номер узла |
|---|---|

то интерпретируется только номер узла, т.к. предполагается, что этот узел принадлежит той же самой сети, что и узел, который отправил пакет;

- если все двоичные разряды IP-адреса равны 1,

| 1111……..11 |
|---|

то этот пакет посылается всем узлам, но только относящимся к данной сети (ограниченное широковещательное сообщение — limited broadcast);

- если сплошные 1 стоят только в поле адреса узла,

| Номер сети | 1111…….11 |
|---|---|

то это означает широковещательное сообщение — broadcast, которое рассылается всем узлам сети с заданным номером. Например, пакет с адресом 192.190.21.155 доставляется всем узлам сети 192.190.21.0;

- еще один адрес 127.0.0.1 зарезервирован для обозначения обратной связи — Loopback, которая используется для тестирования работы программного обеспечения узла без реальной отправки пакета по сети.

*Когда программа посылает данные по IP-адресу 127.0.0.1, то образуется как бы «петля». Данные не передаются по сети, а возвращаются модулям верхнего уровня как только что принятые. Поэтому в IP-сети запрещается присваивать машинам IP-адреса, начинающиеся со 127. Можно отнести адрес 127.0.0.0 ко внутренней сети модуля маршрутизации узла, а адрес 127.0.0.1 — к адресу этого модуля на внутренней сети. На самом деле любой адрес сети 127.0.0.0 служит для обозначения своего модуля маршрутизации, а не только 127.0.0.1, например 127.0.0.3.*

Уже упоминавшаяся форма IP-адреса — multicast — означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле multicast . Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может вхо-

дить в несколько групп. Такие сообщения в отличие от широковещательных называются мультивещательными. Групповые адреса (multicast) имеют специальную структуру, без деления на поля «номер сети» и «номер узла».

*При адресации необходимо учитывать те ограничения, которые вносятся особым назначением некоторых IP-адресов. Так, ни номер сети, ни номер узла не может состоять только из одних двоичных единиц или только из одних двоичных нулей. Отсюда следует, что максимальное количество узлов, приведенное в таблице для сетей каждого класса, на практике должно быть уменьшено на 2. Например, в сетях класса С под номер узла отводится 8 бит, которые позволяют задавать 256 номеров: от 0 до 255. Однако на практике максимальное число узлов в сети класса С не может превышать 254, так как адреса 0 и 255 имеют специальное назначение. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса А состоит из одних двоичных единиц.*

В таблице 3.1 приведены диапазоны адресов для всех классов сетей.

Таблица 3.1

| Класс | Формат | Первые биты | Наибольший № сети | Max число сетей | Max число узлов |
|---|---|---|---|---|---|
| A | с.у.у.у | 0 | 126.0.0.0. | 127 | $2^{24}$=16.777.214 |
| B | с.с.у.у | 10 | 191.255.0.0 | 16.384 | $2^{16}$=65.534 |
| C | с.с.с.у | 110 | 223.225.225.0 | 2.097.152 | $2^{8}$=254 |
| D | | 1110 | 239.225.225.225 | Multicast | |
| E | | 11110 | 247.255.255.255 | Зарезервирован | |

Номер сети назначается специальным подразделением Internet — NIC.

Большие сети получают адреса класса А, средние — класса В, а маленькие — класса С.

## 3.4 Подсети

Если в организации много компьютеров или они далеко стоят друг от друга, имеет смысл разбить большую сеть на несколько более мелких и соединить их через маршрутизаторы. К достоинствам такого подхода относятся:

- Сокращенный сетевой трафик. Все ощутят снижение трафика любого типа. При этом сети остаются однотипными. Без маршрутизаторов трафик грозит почти полностью затормозить работу сети. А при их использовании большая часть трафика остается в локальной сети; через маршрутизатор будут передаваться лишь пакеты, предназначенные потребителям из других сетей.
- Оптимизация производительности сети. Это премия за сокращение сетевого трафика.
- Упрощенное управление. В группе небольших сетей, связанных друг с другом, гораздо легче выявить и решить возникающие проблемы, чем в одной большой сети.
- Упрощенный охват больших географических пространств. Связи глобальных сетей намного медленнее и более дорогостоящие по сравнению со связями локальных сетей. Объединение многочисленных мелких сетей делает всю систему более эффективной.

### *3.4.1 Выделение подсетей*

Выделение подсетей реализуется за счет присвоения некоторого адреса подсети каждой машине заданной физической сети. Например, на рис. 3.4 каждая машина подсети 4 имеет адрес подсети, равный 4.

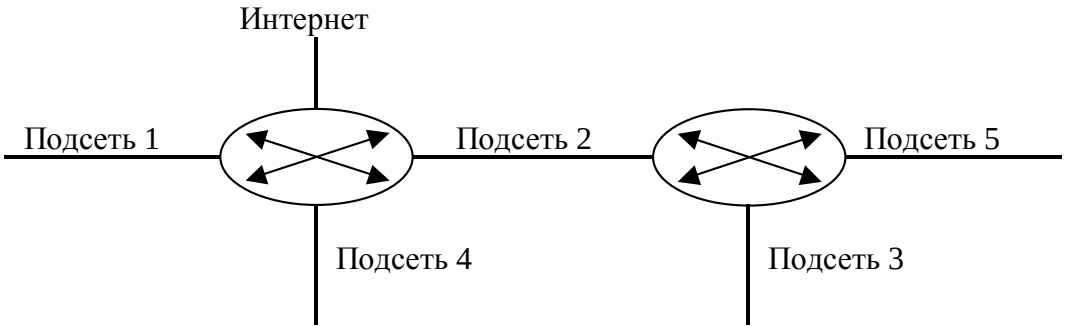



Рисунок 3.4 — Использование подсетей

```
Адрес сети 182.16.0.0
Все хосты сети используют
адрес сети 182.16
```

Теперь рассмотрим, как адрес подсети размещается в IP-адресе. Часть IP-адреса, содержащую адрес сети, изменять нельзя, поскольку все машины сети используют его совместно. Например, все машины сети Аста имеют адрес сети, равный 182.16. Поэтому изменять можно только вторую часть IP-адреса. В схеме адресации с подсетями часть адреса хоста рассматривается как адрес подсети. На рисунке 3.5 показано как в IP-адрес может быть встроен адрес подсети.

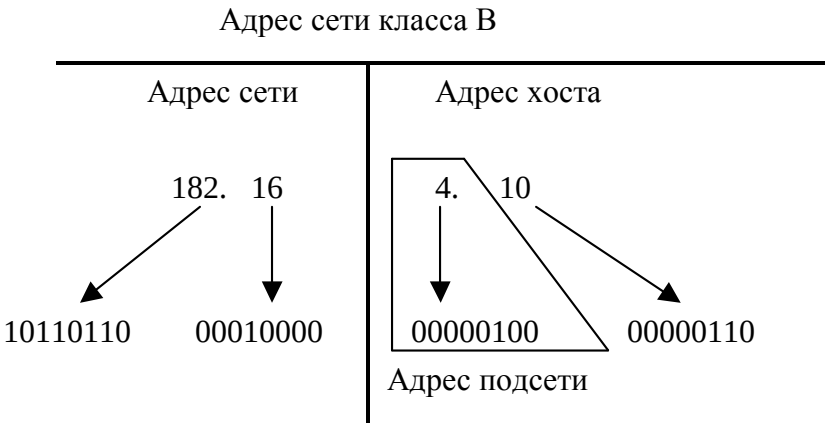



Рисунок 3.5 — Адрес подсети может быть встроен в IP-адрес только в части, относящейся к адресу хоста

Ниже на рис. 3.6 показано, как используются адреса сети и подсети. Этот подход применим к любой подсети, выделенной в составе сети.

Поскольку рассматриваемая сеть относится к классу B, в первых двух байтах IP-адреса содержится адрес сети, совместно используемый всеми машинами сети, независимо от того, к каким подсетям они принадлежат. Адрес каждой машины подсети должен иметь в третьем байте число 0000 0100. Четвертый байт (адрес хоста) содержит уникальное число, идентифицирующее конкретный хост. Таким образом, в состав IP-адреса входят три поля: адрес сети, адрес подсети и адрес хоста.

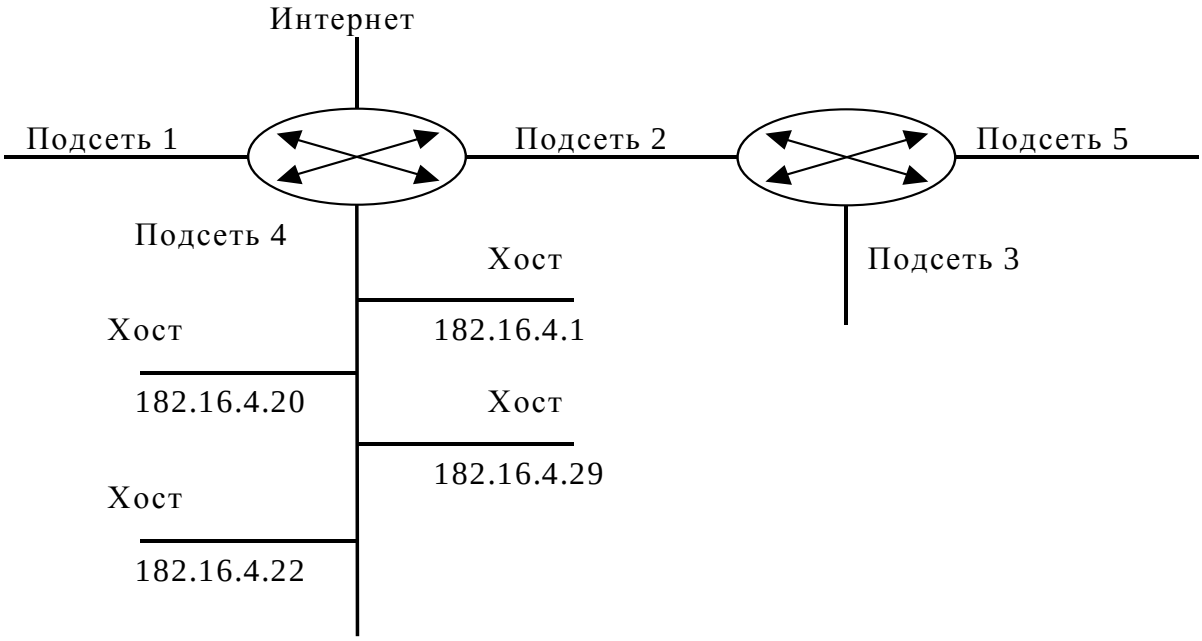



Рисунок 3.6 — Адрес сети и адрес подсети

### *3.4.2 Маски подсетей*

При применении схемы адресации с подсетями каждая машина сети должна знать, какая часть адреса хоста занята адресом подсети. Для этого на каждой машине создается маска подсети.

**Кодирование разрядов маски подсети**

Администратор сети создает 32-битовую маску подсети, состоящую из 0 и 1. Единицы в маске подсети помечают позиции, относящиеся к адресам сети и подсети. Нули заносятся в позиции, отведенные под адрес хоста. Эта идея проиллюстрирована на рисунке 3.7.

Кодирование разрядов маски подсети

Единицами помечено положение адреса сети и подсети
Нулями помечено положения адреса хоста

Маска подсети для компании Асма

| 11111111.11111111. | 11111111. | 00000000 |
|---|---|---|
| Позиция адреса подсети | Позиция маски подсети | Позиция адреса хоста |

Рисунок 3.7 — Маска подсети

В примере первые два байта маски подсети заполнены единицами, поскольку адрес сети относится к классу B (формат Сеть.Сеть.Узел.Узел). Третий байт, обычно относящийся к адресу хоста, теперь представляет собой адрес подсети. Поэтому все его биты заполнены единицами. И только в четвертом байте хранится уникальный адрес хоста.

Маска подсети может быть также представлена в десятичном формате. Двоичная комбинация 1111 1111 соответствует десятичному числу 255, Таким образом, маску подсети нашего примера можно отобразить двумя способами (см. рисунок 3.8).

Двоичное представление:
11111111.11111111.11111111.00000000
Десятичное представление:
255.255.255.0

Рисунок 3.8 — Представления маски подсети

Совсем необязательно, чтобы в состав сети входили подсети, т.е. маска подсети может и не использоваться. В подобных случаях говорят, что существует маска подсети, принятая по умолчанию. Маски, принятые по умолчанию для различных классов сетей (см. таблицу 3.2), изменять нельзя.

Таблица 3.2 — Маски подсетей, принятые по умолчанию

| Класс сети | Формат адреса | Маска подсети |
|---|---|---|
| A | Сеть.Узел.Узел.Узел | 255.0.0.0 |
| B | Сеть.Сеть.Узел.Узел | 255.255.0.0 |
| C | Сеть.Сеть.Узел.Узел | 255.255.255.0. |

После того как администратор сети создаст маску подсети для каждой машины, программное обеспечение протокола IP будет просматривать IP-адреса, используя маску подсети для определения адреса подсети. Соответствующая операция представлена на рисунке 3.9.

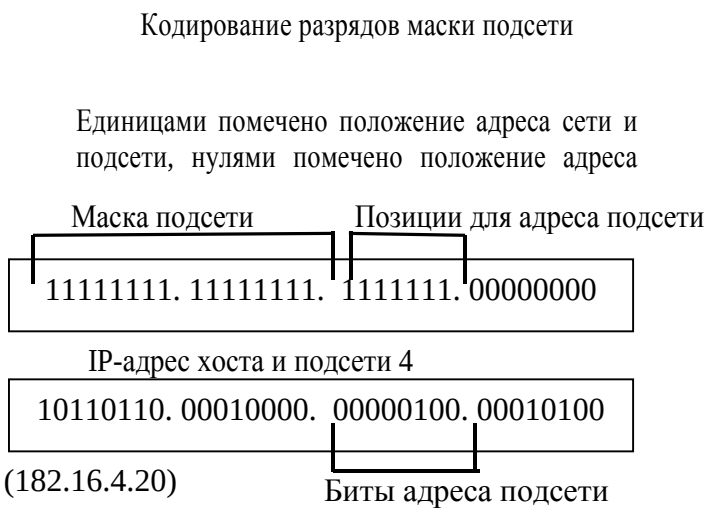



Рисунок 3.9 — Просмотр IP-адреса с помощью маски подсети

В этом примере по маске подсети программно определяется, что третий байт IP-адреса больше не относится к адресу хоста, а занят адресом подсети, равным 0000 0100.

Если для размещения адреса подсети использовать третий байт IP-адреса класса B, то задание и определение адреса подсети становится легким делом. Например, если в сети Аста необходимо определить подсеть 12, в третий байт адресов всех машин этой подсети следует вставить 0000 1100.

При использовании всего третьего байта IP-адреса класса B для адреса подсети можно создать довольно много подсетей. Действительно, так как байт состоит из восьми битов, можно определить $2^8 = 256$ различных адресов. Однако поскольку адрес не может состоять только из единиц или нулей, число возможных подсетей уменьшается до 254. Таким образом, в сети Аста можно выделить до 254 подсетей, каждая из которых будет содержать до 254 хостов.

Приведем формулы для подсчета максимального числа подсетей и хостов в подсети:

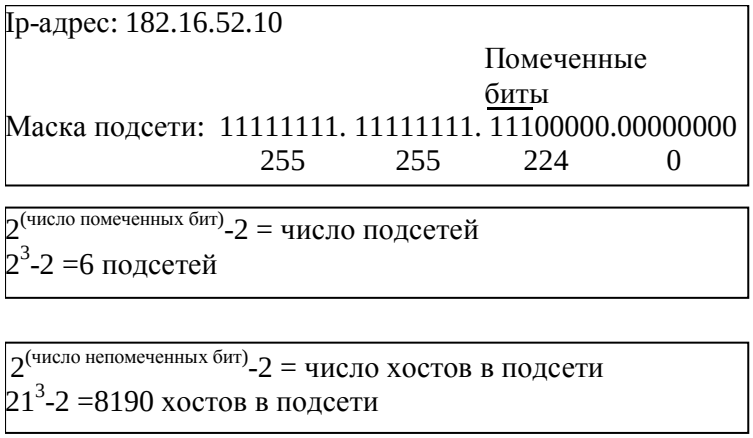



Рисунок 3.10 — Формулы для подсчета подсетей и хостов

*Помеченными* считаются биты с единичным значением, а *непомеченными* — нулевые биты. На рисунке 3.10 приведен пример применения этих формул.

Таким образом, если число помеченных битов равно 3, согласно приведенным формулам можно создать до шести подсетей. Для адресации хостов остается 5 бит в третьем байте и 8 бит в четвертом. При 13-битовых адресах к подсети можно подключить до $2^{13} – 2 = 8190$ хостов**.**

*При возврате к применению полного байта адреса узла в качестве адреса подсети (маска 255.255.255.0) возможное количество узлов в каждой подсети сократится. Без выделения подсетей адрес класса B может иметь 65 534 различных значений, которые разрешается использовать в качестве адреса узлов.*

*Если адрес подсети занимает полный байт адреса узла, остается только один байт для адресов хостов, т.е. в подсети можно разместить до 254 хостов. Когда в какой-нибудь подсети потребуется увеличить это число, возникнут затруднения. Для их разрешения вам придется либо укоротить поле адреса подсети в маске и тем самым увеличить длину адреса хоста, либо добавить вторичный IP-адрес к интерфейсу маршрутизатора, что также позволит увеличить число хостов в подсети. Однако побочным эффектом такого решения будет уменьшение количества образуемых подсетей.*

Ниже приведен пример использования более коротких адресов подсетей. Если компании необходимо образовать 14 подсетей, то нет смысла отводить под адрес подсети весь третий байт IP-адреса. Для создания 14-ти различных адресов подсетей достаточно занять только 4 бита адреса хоста ($2^4 – 2 = 14$). При этом в остав-

шихся 12-ти битах адреса хоста можно разместить $2^{12} - 2 = 4094$ различных адресов. Таким образом, в сети выделяется 14 подсетей, к каждой из которых можно подключить до 4094 хостов.

**Использование четырех битовых адресов подсетей**
**Аста**

Адрес сети: **132.8** (класс B; сеть.сеть.хост.хост)

Пример IP-адреса: **10000100.0000 1000.0001 0010. 0011 1100**

Десятичное представление: 132 . 8 . 18 . 60

Кодирование разрядов маски подсети

Единицами помечено положение адреса сети и подсети
Нулями помечено положение адреса хоста
*Маска подсети:*
*Двоичное представление:* **1111 1111. 1111 1111.1111 0000. 0000 0000**
*Десятичное представление:* 255 . 255 . 240 . 0
(Десятичное число 240 равно двоичному числу 1111 0000)

Позиции, отведенные под адрес подсети

*Маска подсети:* 1111 **1111. 1111 1111. 1111 0000. 0000 0000**
*IP-адрес машины:* **1000 0100. 0000 1000. 0001 0010. 0011 1100**
(Десятичное 132.8.16.60) I I

Позиции, отведенные под адрес подсети

**Преобразование адреса подсети из двоичного в десятичное представление**

*Позиции маски подсети:* **1 1 1 1 0 0 0 0**
*Значения:* **128 64 32 16 8 4 2 1**
*Третий байт IP-адреса*: **0 0 0 1 0 0 1 0**
*Десятичный эквивалент:* **0 + 16 = 16**
*Адрес подсети в IP-адресе:* **16**

Итак, IP-адрес сети: 132.8.0.0.
IP-адрес подсети: 0.0.16.0.
IP-адрес хоста: 0.0.2.60.

# 4 ПРИНЦИПЫ МАРШРУТИЗАЦИИ

Важнейшей задачей сетевого уровня является маршрутизация — передача пакетов между двумя конечными узлами в составной сети.

Рассмотрим принципы маршрутизации на примере составной сети, изображенной на рисунке 4.1.

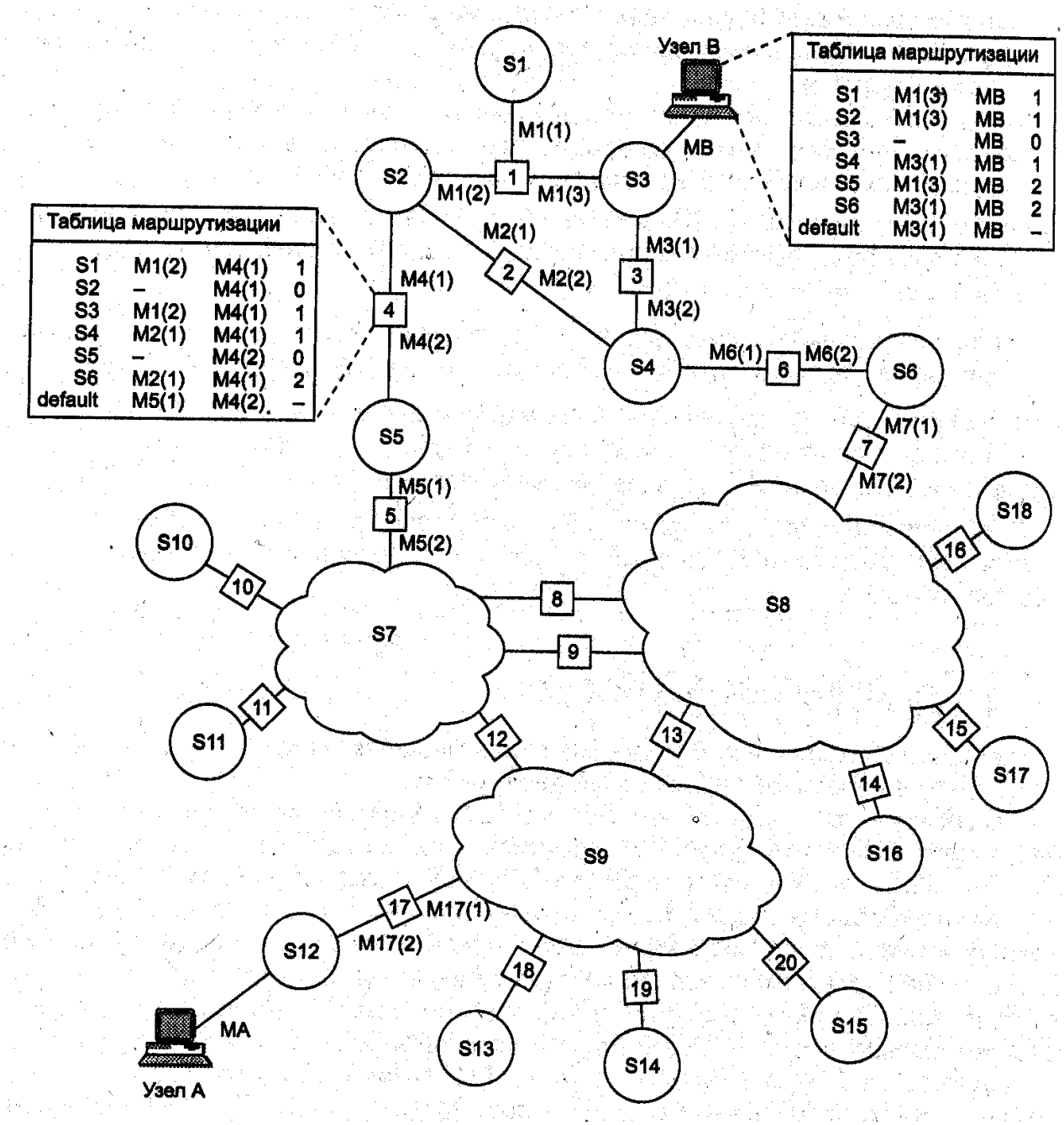



Рисунок 4.1 — Пример составной сети

В этой сети 20 маршрутизаторов объединяют 18 сетей в общую сеть; S1, S2, ..., S18 — это номера сетей. Маршрутизаторы имеют по нескольку портов (по крайней мере, по два), к которым присоединяются сети. Каждый порт маршрутизатора можно рассматривать как отдельный узел сети: он имеет собственный сетевой адрес и собственный локальный адрес в той подсети, которая к нему подключена. Например, маршрутизатор под номером 1 имеет три порта, к которым подключены сети S1, S2, S3. На рисунке сетевые адреса этих портов обозначены как M1(1), M1(2) и M1(3). Порт M1(1) имеет локальный адрес в сети с номером S1, порт M1(2) — в сети S2, а порт M1(3) — в сети S3. Таким образом, маршрутизатор можно рассматривать как совокупность нескольких узлов, каждый из которых входит в свою сеть. Как единое устройство маршрутизатор не имеет ни отдельного сетевого адреса, ни какого-либо локального адреса.

Маршрут — это последовательность маршрутизаторов, которые должен пройти пакет от отправителя до пункта назначения.

В сложных составных сетях почти всегда существует несколько альтернатив-маршрутов для передачи пакетов между двумя конечными узлами. Так, пакет, отправленный из узла A в узел B, может пройти через маршрутизаторы 17, 12, 5, 4 и 1 или маршрутизаторы 17,13,7, 6 и 3. Нетрудно найти еще несколько маршрутов между узлами A и B.

Задачу выбора маршрута из нескольких возможных решает маршрутизатор, а также конечные узлы. Маршрут выбирается исходя из текущей конфигурации сети, а также указанного критерия выбора маршрута. Обычно в качестве критерия выступает задержка прохождения маршрута отдельным пакетом или средняя пропускная способность маршрута. *Часто также используется весьма простой критерий, учитывающий только количество пройденных в маршруте промежуточных маршрутизаторов (хопов).*

Чтобы по адресу сети назначения можно было бы выбрать рациональный маршрут дальнейшего следования пакета, каждый маршрутизатор анализирует таблицу маршрутизации.

Для примера рассмотрим как будет выглядеть таблица маршрутизации для схемы на рис. 4.1, например, в маршрутизаторе 4 (табл. 4.1).

Таблица 4.1 — Таблица маршрутизации маршрутизатора 4

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| S1 | M1(2) | M4(1) | 1 |
| S2 | – | M4(1) | 0 (подсоединена) |
| S3 | M1(2) | M4(1) | 1 |
| S4 | M2(1) | M4(1) | 1 |
| S5 | – | M4(2) | 0 (подсоединена) |
| S6 | M2(1) | M4(1) | 2 |
| Default | M5(1) | M4(2) | – |

Таблица 4.1 значительно упрощена по сравнению с реальными таблицами, например, отсутствуют столбцы с масками, признаками состояния маршрута, временем, в течение которого действительны записи данной таблицы (их применение будет рассмотрено позже). Кроме того, как уже было сказано, здесь указаны адреса сетей условного формата, не соответствующие какому-либо определенному сетевому протоколу. Тем не менее эта таблица содержит основные поля, имеющиеся в реальных таблицах при использовании конкретных сетевых протоколов, таких как IP, IPX или X.25.

В первом столбце таблицы перечисляются номера сетей, входящих в интерсеть. В каждой строке таблицы следом за номером сети указывается сетевой адрес следующего маршрутизатора (более точно — сетевой адрес соответствующего порта следующего маршрутизатора), на который надо направить пакет, чтобы тот передвигался по направлению к сети с данным номером по рациональному маршруту.

Когда на маршрутизатор поступает новый пакет, номер сети назначения, извлеченный из поступившего кадра, последовательно сравнивается с номерами сетей из каждой строки таблицы. Строка с совпавшим номером сети указывает, на какой ближайший маршрутизатор следует направить пакет. Например, если на какой-либо порт маршрутизатора 4 поступает пакет, адресованный в сеть S6, то из таблицы маршрутизации следует, что адрес следующего маршрутизатора — M2(1), то есть очередным этапом

движения данного пакета будет движение к порту 1 маршрутизатора 2.

Число записей в таблице маршрутизации уменьшают за счет использования специальной записи — «маршрутизатор по умолчанию» (default).

*Если принять во внимание топологию составной сети, то в таблицах маршрутизаторов, находящихся на периферии составной сети, достаточно записать номера сетей, непосредственно подсоединенных к данному маршрутизатору или расположенных поблизости, на тупиковых маршрутах. Обо всех же остальных сетях можно сделать в таблице единственную запись, указывающую на маршрутизатор, через который пролегает путь ко всем этим сетям. Такой маршрутизатор называется маршрутизатором по умолчанию, а вместо номера сети в соответствующей строке помещается особая запись, например, default.*

В нашем примере таким маршрутизатором по умолчанию для сети S5 является маршрутизатор 5, точнее его порт M5(1). Это означает, что путь из сети S5 почти ко всем сетям большой составной сети пролегает через этот порт маршрутизатора.

Часто порт по умолчанию связан с магистралью сети, так что пакеты, попав на магистраль, попадут, в конце концов, в маршрутизатор, знающий их сеть назначения.

Перед тем как передать пакет следующему маршрутизатору, текущий маршрутизатор должен определить, на какой из нескольких собственных портов он должен поместить данный пакет. Для этого служит третий столбец таблицы маршрутизации. Еще раз подчеркнем, что каждый порт идентифицируется собственным сетевым адресом.

Некоторые реализации сетевых протоколов допускают наличие в таблице маршрутизации сразу нескольких строк, соответствующих одному и тому же адресу сети назначения. В этом случае при выборе маршрута принимается во внимание столбец «Расстояние до сети назначения». Расстояние может измеряться хопами, временем прохождения пакета по линиям связи, какой-либо характеристикой надежности линий связи на данном маршруте.

В табл. 4.1 расстояние между сетями измерялось хопами. Расстояние для сетей, непосредственно подключенных к портам маршрутизатора, здесь принимается равным 0.

Наличие нескольких маршрутов к одному узлу делают возможным передачу трафика к этому узлу параллельно по нескольким каналам связи, это повышает пропускную способность и надежность сети.

Задачу маршрутизации решают также конечные узлы — компьютеры. Средства сетевого уровня должны определить, направляется ли пакет в другую сеть или адресован какому-нибудь узлу данной сети. Если номер сети назначения совпадает с номером данной сети, то для данного пакета не требуется решать задачу маршрутизации. Если же номера сетей отправления и назначения не совпадают, то маршрутизация нужна. Таблицы маршрутизации конечных узлов полностью аналогичны таблицам маршрутизации, хранящимся на маршрутизаторах.

Таблица маршрутизации для конечного узла B схемы 4.1 могла бы выглядеть следующим образом (табл. 4.2). Здесь MB — сетевой адрес порта компьютера B. На основании этой таблицы конечный узел B выбирает, на какой из двух имеющихся в локальной сети S3 маршрутизаторов следует посылать тот или иной пакет.

Таблица 4.2 — Таблица маршрутизации конечного узла B

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| S1 | M1(3) | MB | 1 |
| S2 | M1(3) | MB | 1 |
| S3 | – | MB | 0 |
| S4 | M3(1) | MB | 1 |
| S5 | M1(3) | MB | 2 |
| S6 | M3(1) | MB | 2 |
| Default | M3(1) | MB | – |

Конечные узлы в еще большей степени, чем маршрутизаторы, пользуются приемом маршрутизации по умолчанию. Хотя они также в общем случае имеют в своем распоряжении таблицу

маршрутизации, ее объем обычно незначителен, что объясняется периферийным расположением всех конечных узлов. Конечный узел часто вообще работает без таблицы маршрутизации, имея только сведения об адресе маршрутизатора по умолчанию. При наличии одного маршрутизатора в локальной сети этот вариант — единственно возможный для всех конечных узлов.

Ниже помещена таблица маршрутизации другого конечного узла составной сети — узла А (табл. 4.3). Компактный вид таблицы маршрутизации отражает тот факт, что все пакеты, направляемые из узла А, либо не выходят за пределы сети S12, либо непременно проходят через порт 1 маршрутизатора 17. Этот маршрутизатор и определен в таблице маршрутизации в качестве маршрутизатора по умолчанию.

Таблица 4.3 — Таблица маршрутизации конечного узла А

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| S12 | – | MA | 0 |
| default | M17(1) | MA | – |

Для конечных узлов таблицы маршрутизации часто создаются вручную администраторами и хранятся в виде постоянных файлов на дисках.

## 4.1 Протоколы маршрутизации

Задача маршрутизации решается на основе анализа таблиц маршрутизации, размещенных во всех маршрутизаторах и конечных узлах сети.

Алгоритмы маршрутизации, в которых каждый маршрутизатор ответственен за выбор только одного шага маршрута, а окончательный маршрут складывается в результате работы всех маршрутизаторов, называются *одношаговыми (одноликовыми).*

При *маршрутизации от источника* узел-источник задает в отправляемом в сеть пакете полный маршрут его следования через все промежуточные маршрутизаторы. При этом нет необходимости

строить и анализировать таблицы маршрутизации. Это ускоряет прохождение пакета по сети, разгружает маршрутизаторы, но при этом большая нагрузка ложится на конечные узлы.

Эта схема в вычислительных сетях применяется сегодня гораздо реже, чем схема распределенной одношаговой маршрутизации. Однако в новой версии протокола IP наряду с классической одношаговой маршрутизацией будет разрешена и маршрутизация от источника.

Одношаговые алгоритмы в зависимости от способа формирования таблиц маршрутизации делятся на три класса:

– алгоритмы статической маршрутизации;
– алгоритмы простой маршрутизации;
– алгоритмы динамической маршрутизации.

В алгоритмах статической маршрутизации администратор сети сам решает, на какие маршрутизаторы надо передавать пакеты с теми или иными адресами, и вручную заносит соответствующие записи в таблицу маршрутизации. Таблица создается в процессе загрузки и в дальнейшем используется без изменений до тех пор, пока ее содержимое не будет отредактировано вручную. Такие исправления могут понадобиться, например, если в сети отказывает какой-либо маршрутизатор и его функции возлагаются на другой маршрутизатор. Различают одномаршрутные таблицы, в которых для каждого адресата задан один путь, и многомаршрутные таблицы, определяющие несколько альтернативных путей для каждого адресата. В многомаршрутных таблицах должно быть задано правило выбора одного из маршрутов. Чаще всего один путь является основным, а остальные — резервными. Понятно, что алгоритм фиксированной маршрутизации с его ручным способом формирования таблиц маршрутизации приемлем только в небольших сетях с простой топологией. Однако этот алгоритм может быть эффективно использован и для работы на магистралях крупных сетей, так как сама магистраль может иметь простую структуру с очевидными наилучшими путями следования пакетов в подсети, присоединенные к магистрали.

В алгоритмах простой маршрутизации таблица маршрутизации либо вовсе не используется, либо строится без участия протоколов маршрутизации. Выделяют три типа простой маршрутизации:

– случайная маршрутизация, когда прибывший пакет посылается в первом попавшем случайном направлении;

– лавинная маршрутизация, когда пакет широковещательно посылается по всем возможным направлениям, кроме исходного (аналогично обработке мостами кадров с неизвестным адресом);

– маршрутизация по предыдущему опыту, когда выбор маршрута осуществляется по таблице, но таблица строится по принципу моста путем анализа адресных полей пакетов, появляющихся на входных портах.

Самыми распространенными являются алгоритмы динамической маршрутизации. Эти алгоритмы обеспечивают автоматическое обновление таблиц маршрутизации после изменения конфигурации сети. Протоколы, построенные на основе динамических алгоритмов, позволяют всем маршрутизаторам собирать информацию о топологии связей в сети, оперативно отрабатывая все изменения конфигурации связей. В таблицах маршрутизации при динамической маршрутизации обычно имеется информация об интервале времени, в течение которого данный маршрут будет оставаться действительным. Это время называют *временем жизни маршрута (Time To Live, TTL).*

Динамические алгоритмы обычно имеют распределенный характер, который выражается в том, что в сети отсутствуют какие-либо выделенные маршрутизаторы, которые собирали бы и обобщали топологическую информацию: эта работа распределена между всеми маршрутизаторами.

Динамические протоколы обмена маршрутной информацией делятся на две группы:

– дистанционно-векторные алгоритмы;

– алгоритмы состояния связей.

В алгоритмах **дистанционно-векторного типа** каждый маршрутизатор периодически и широковещательно рассылает по сети вектор, компонентами которого являются расстояния от данного маршрутизатора до всех известных ему сетей. Под расстоянием обычно понимается число хопов (возможно учитывать и время прохождения пакетов по сети между соседними маршрутизаторами). При получении вектора от соседа маршрутизатор наращивает расстояния до указанных в векторе сетей на расстояние до данного соседа. Получив вектор от соседнего маршрутиза-

тора, каждый маршрутизатор добавляет к нему информацию об известных ему других сетях, а затем снова рассылает новое значение вектора по сети. В конце концов, каждый маршрутизатор узнает информацию обо всех имеющихся в интерсети сетях и о расстоянии до них через соседние маршрутизаторы.

Дистанционно-векторные алгоритмы хорошо работают только в небольших сетях. В больших сетях они засоряют линии связи интенсивным широковещательным трафиком, к тому же изменения конфигурации могут отрабатываться по этому алгоритму не всегда корректно, так как маршрутизаторы не имеют точного представления о топологии связей в сети, а располагают только обобщенной информацией — вектором дистанций, к тому же полученной через посредников. Работа маршрутизатора в соответствии с дистанционно-векторным протоколом напоминает работу моста, так как точной топологической картины сети такой маршрутизатор не имеет.

Наиболее распространенным протоколом, основанным на дистанционно-векторном алгоритме, является протокол RIP.

***Алгоритмы состояния связей*** обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети. На начальном этапе каждый маршрутизатор передает информацию только о своих непосредственных связях с помощью пакетов состояний (LSP) или пакетов-приветствий (Hello). Затем каждый маршрутизатор компилирует все LSP, поступившие из сети, и образует базу данных о топологии сети. После этого вычисляются маршруты ко всем сетям: для каждой сети с маршрутизацией по состоянию вычисляются кратчайший и оптимальный пути. Затем каждый маршрутизатор формирует древовидную структуру, в которой он сам является корнем.

Протоколами, основанными на алгоритме состояния связей, являются протоколы OSPF стека TCP/IP, IS-IS стека OSI.

Фирмой Cisco были разработаны протоколы IGRP и EIGRP.

# 5 ПРОТОКОЛЫ ARP И RARP (RFC 826)

ARP (Address Resolution Protocol) — протокол определения адреса.

RARP (Reverse Address Resolution Protocol) — протокол обратного определения адреса.

Когда протокол IP должен отправить датаграмму, он уже имеет информацию от протоколов верхних уровней об IP-адресе получателя. Однако протокол IP должен в свою очередь указать протоколу уровня доступа к сети, например, Ethernet или Token Ring, аппаратный адрес получателя. Если этот адрес неизвестен, для его получения IP применяет протокол ARP. ARP производит широковещательный опрос сети, требуя, чтобы компьютер с определенным IP-адресом сообщил свой аппаратный адрес, например, адрес сетевой платы Ethenet компьютера-получателя, и по нему выясняет местоположение получателя. Этот аппаратный адрес называют MAC-адресом или физическим адресом.

**32 bit IP-address**

**RARP ↑ ↓ ARP**

**48 bit Ethernet-address**

ARP используется только в момент отправки пакета в сеть. Основным инструментом является ARP таблица.

Алгоритм ARP:

1. Broadcast Ethernet.

Каждый сетевой адаптер принимает широковещательные передачи. ARP-запрос можно интерпретировать так: «Если ваш IP-адрес совпадает с указанным, то сообщите мне ваш Ethernet-адрес». Пакет ARP-запроса выглядит примерно так:

| | |
|---|---|
| *IP-адрес отправителя* | *223.1.2.1* |
| *Ethernet-адрес отправителя* | *08:00:39:00:2F:C3* |
| *Искомый IP-адрес* | *223.1.2.2* |
| *Искомый Ethernet-адрес* | *<пусто>* |

Рисунок 5.1 — Пример ARP-запроса

2. Исходная IP-дейтаграмма ставится в очередь.

3. Возвращенный ARP-ответ помещается в ARP-таблицу.

Каждый модуль ARP проверяет поле искомого IP-адреса в полученном ARP-пакете и, если адрес совпадает с его собственным IP-адресом, посылает ответ прямо по Ethernet-адресу отправителя запроса. ARP-ответ можно интерпретировать так: «Да, это мой IP-адрес, ему соответствует такой-то Ethernet-адрес». Пакет с ARP-ответом выглядит примерно так:

| | |
|---|---|
| *IP-адрес отправителя* | *223.1.2.2* |
| *Ethernet-адрес отправителя* | *08:00:28:00:38:A9* |
| *Искомый IP-адрес* | *223.1.2.1* |
| *Искомый Ethernet-адрес* | *08:00:39:00:2F:C3* |

Рисунок 5.2 — Пример ARP-ответа

Этот ответ получает машина, сделавшая ARP-запрос. Драйвер этой машины проверяет поле типа в Ethernet-кадре и передает ARP-пакет модулю ARP. Модуль ARP анализирует ARP-пакет и добавляет запись в свою ARP-таблицу.

Обновленная таблица выглядит следующим образом:

| *IP-адрес* | *Ethernet-адрес* |
|---|---|
| *223.1.2.1* | *08:00:39:00:2F:C3* |
| *223.1.2.2* | *08:00:28:00:38:A9* |
| *223.1.2.3* | *08:00:5A:21:A7:22* |
| *223.1.2.4* | *08:00:10:99:AC:54* |

Рисунок 5.3 — ARP-таблица после обработки ответа

4. Выполняется преобразование IP→Ethernet для поставленного в очередь пакета.

5. Ethernet-кадр паредается в сеть.

Если на шаге 3 нет ARP ответа, IP-дейтаграмма уничтожается.

Для сокращения в сети широковещательного траффика, порожденного протоколом ARP, каждый конечный хост имеет общую ARP-таблицу для всех своих сетевых интерфейсов. Записи в кэше могут быть статическими (создаются пользователем) и ди-

намическими, причем динамические записи периодически обновляются.

RARP используется гораздо реже и, в основном, применяется для сетевой загрузки бездисковых компьютеров.

## 5.1 Транспортный уровень стека TCP/IP. Протокол UDP (User Datagram Protocol), RFC768

UDP (User Datagram Protocol) — протокол передачи пользовательских датаграмм.

Идентификатор протокола в заголовке IP для UDP — 17.

Протокол UDP не создает виртуальных каналов и называется протоколом без установления соединения. UDP обеспечивает негарантированную доставку информации и при этом использует значительно меньше ресурсов.

В некоторых ситуациях гораздо выгоднее применять протокол UDP, а не TCP. Затраты на установление соединения, поддержку и закрытие соединения для каждого мелкого сообщения могли бы значительно понизить производительность сети. Кроме того, UDP имеет преимущество перед TCP в том случае, если надежность передачи обеспечивается на уровне приложений/процессов. Однако решение об используемом протоколе принимается разработчиком приложения, а не пользователем, который хотел бы ускорить передачу данных.

Протокол UDP получает блоки информации с более высоких уровней и разбивает их на сегменты. Каждому сегменту назначается порядковый номер для последующей сборки в исходный блок у получателя. Однако UDP не заботится о последовательности поступления пакетов к получателю. Он всего лишь нумерует и отправляет сегменты, сразу же забывая о них. UDP не проверяет, доставлены ли сегменты, и даже не допускает подтверждений, поэтому считается *ненадежным* протоколом.

Адреса, которые используются протоколами транспортного уровня, называются номерами портов. Номер порта состоит из 16 бит и стандартизован (RFC 1700). Порт можно определить как абстрактную точку назначения конкретной прикладной программы, находящейся на конкретном компьютере. Механизм портов

позволяет хосту одновременно поддерживать несколько сеансов связи.

**Определение.** Совокупность IP-адреса и номера порта называется сокетом (socket).

Назначение портов приложениям на хостах происходит независимо друг от друга. Модули транспортного уровня могут сами назначить порт или приложение укажет номер порта, с которым оно будет работать. Порт может задаваться любым числом в диапазоне 0÷65536 $(2^{16})^4$. Отметим, что на портах 0÷1023 работают привилегированные процессы. Например, сервер SNMP всегда ожидает поступлений сообщений в порт 161. Если клиент SNMP желает получить услугу, он посылает запрос в UDP-порт 161 на машину, где работает сервер. В каждом узле может быть только один сервер SNMP, так как существует только один UDP-порт 161. Данный номер порта является общеизвестным, то есть фиксированным номером, официально выделенным для услуг SNMP. Общеизвестные номера определяются стандартами Internet.

Метод инкапсуляции часто применяется, когда двум сетям, использующим один и тот же сетевой протокол, нужно связаться через транзитную сеть, которая работает с другими сетевыми протоколами.

Инкапсуляция UDP:

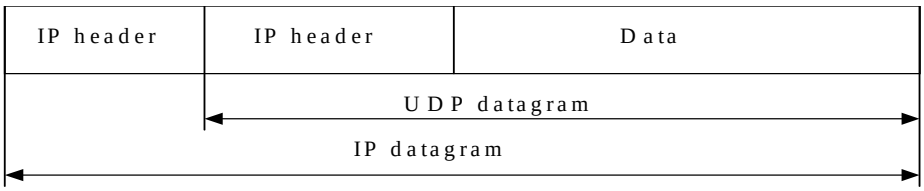



Рисунок 5.4 — Инкапсуляция UDP дейтаграммы

Поля заголовка UDP-дейтаграммы

| Порт источника<br>Source Port<br>(16 бит) | Порт получателя<br>Destination Pont<br>(16 бит) | Длина<br>Length<br>(16 бит) | Контрольная<br>сумма<br>Checksum | Данные<br>Data |
|---|---|---|---|---|

Рисунок 5.5 — Формат сегмента UDP

Заголовок UDP занимает всего 8 байт. Таким образом, накладные расходы при использовании UDP минимальны. Разработчик приложения может выбрать протокол TCP для обеспечения надежности или протокол UDP для ускорения передачи.

Source Port (16 bit) — порт отправителя. Может содержать номер порта, с которого был отправлен пакет, когда это имеет значение (отправитель ждет ответа). Если нет, поле заполняется нулями.

Destination Pont (16 bit) — порт получателя — порт хоста назначения, на который будет доставлена данная дейтограмма.

Length (16 bit) — длина (в байтах) дейтаграммы, включая заголовок и данные. Минимальное значение — 8, максимальное = = 65535 – 20 – 8 = 65507.

Checksum (16 bit) — контрольная сумма.

Если поле Checksum=0, значит, контрольная сумма не вычислялась. Контрольная сумма, как правило, не вычисляется при работе на высоконадежных линиях связи. Так как протокол IP не вычисляет контрольную сумму поля данных в IP-дейтограммах, только контрольная сумма UDP может указать целостность пришедших данных.

Сервисы, использующие UDP: DNS, TFTP, NTP, BOOTP.

При вычислении контрольной суммы поле Checksum полагается = 0. Если рассчитанная контрольная сумма = 0, то она передается как поле, целиком состоящее из единиц (из RFC 768).

Замечание по терминологии:

Пакет — модуль данных, передаваемых между уровнем IP и уровнем сетевого интерфейса. Пакет может быть полной дейтаграммой или ее фрагментом.

### *5.1.1 Фрагментация IP-пакетов*

В большинстве типов сетей значение MTU, то есть максимальный размер поля данных, в которое должен инкапсулировать свой пакет протокол IP, значительно отличается. Сети Ethernet имеют MTU = 1500 байт, сети FDDI = 4096 байт, а сети X – 25 = 128 байт.

Процедуры фрагментации рассчитаны на то, что пакет мог быть разбит на любое количество частей, которые впоследствии могли бы быть вновь собраны.

Для процедуры фрагментации в IP-пакетах поле **Идентификация** (которое используется для того, чтобы не перепутать фрагменты различных пакетов) устанавливается значение уникальное для данной пары отправитель-получатель, а также время, в течение которого пакет может быть активным в сети. Поле **Смещение фрагмента** (используется при сборке/разборке фрагментов пакетов при передачах их между сетями с различными величинами MTU) указывает положение данного фрагмента в исходном пакете. **Флаг** «more fragments» показывает появление последнего фрагмента.

Эти поля дают достаточное количество информации для сборки пакетов.

Пример фрагментации IP дейтаграммы:

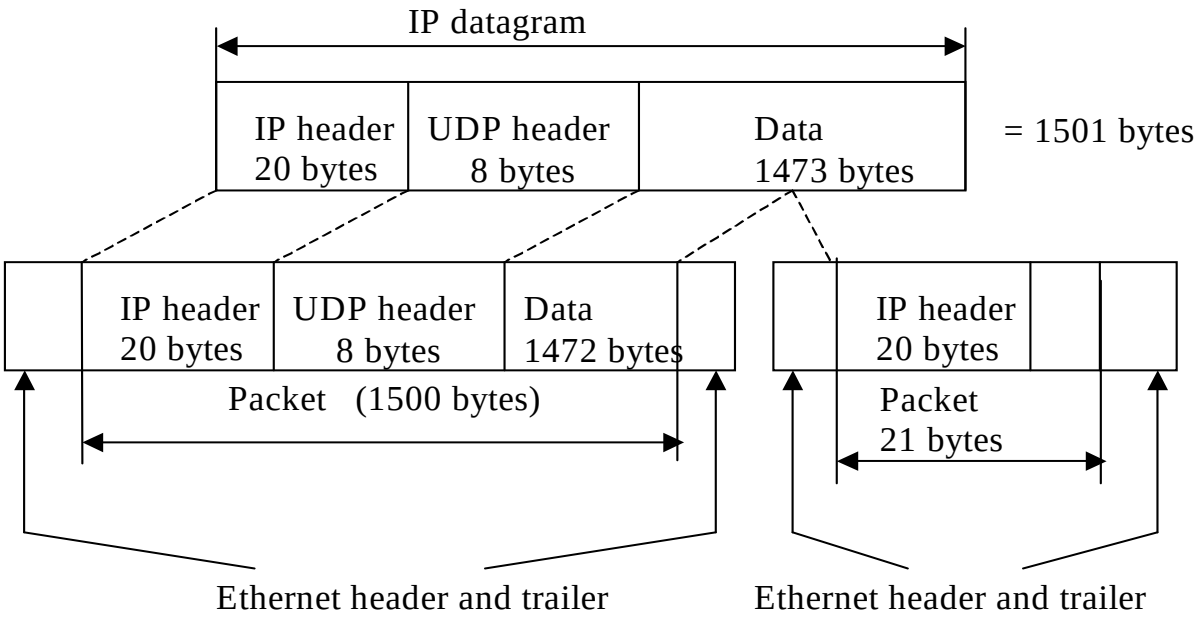



Рисунок 5.6

# 6 ПРОТОКОЛ ICMP (RFC 792)

За работой Интернета следят маршрутизаторы. Когда случается что-нибудь неожиданное, о происшествии сообщается по протоколу ICMP (Internet Control Message Protocol). Идентификатор этого протокола в заголовке IP – 1.

Этот протокол часто рассматривается как часть уровня IP. Он передает сообщения об ошибках и других ситуациях, требующих внимания. Причем обратим внимание на то, что ICMP только сообщает об ошибках, но не исправляет их. ICMP сообщения содержат временные метки, определяющие когда они были посланы. Сообщения ICMP обычно возникают либо на уровне IP, либо на транспортном (UDP или TCP) уровне.

Протоколом ICMP определено около десяти типов сообщений.

Таблица 6.1 — Основные типы ICMP-сообщений

| **Тип сообщения** | | Код | **Описание** |
|---|---|---|---|
| 3 | Адресат недоступен | 0-15 | Пакет не может быть доставлен |
| 11 | Время истекло | 0,1 | Время жизни пакета упало до нуля |
| 12 | Проблема с параметром | 0,1 | Неверное поле заголовка |
| 4 | Гашение источника | 0 | Сдерживающий пакет |
| 5 | Переадресовать | 0-3 | Научить маршрутизатор географии |
| 15 | Запрос отклика | 0 | Спросить машину, жива ли она |
| 16 | Отклик | 0 | Да, я жива |
| 13 | Запрос временного штампа | 0 | То же, что и Запрос отклика, но с временным штампом |

Сообщение АДРЕСАТ НЕДОСТУПЕН используется, когда подсеть или маршрутизатор не могут обнаружить пункт назначения или когда пакет с битом DF (не фрагментировать) не может быть доставлен, так как путь преграждает сеть с маленьким размером пакетов.

Сообщение ВРЕМЯ ИСТЕКЛО посылается, когда пакет игнорируется, так как его счетчик уменьшился до нуля. Это событие является признаком того, что пакет двигается по замкнутым

путям, что имеется большая перегрузка или установлено слишком низкое значение таймера.

Сообщение ПРОБЛЕМА С ПАРАМЕТРОМ указывает, что обнаружено неверное значение поле заголовка, что является признаком наличия ошибки в программном обеспечении отправившего этот пакет хоста или промежуточного маршрутизатора.

Сообщение ГАШЕНИЕ ИСТОЧНИКА ранее использовалось для усмирения хостов, которые отправляли слишком много пакетов. Хост, получивший такое сообщение, должен был снизить обороты. В настоящее время подобное сообщение редко используется, так как при возникновении перегрузки подобные пакеты только подливают масла в огонь. Теперь борьба с перегрузкой в Интернете осуществляется в основном на транспортном уровне.

Сообщение ПЕРЕАДРЕСОВАТЬ посылается хосту, отправившему пакет, когда маршрутизатор замечает, что пакет адресован неверно.

Сообщение ОТКЛИК и ЗАПРОС ОТКЛИКА посылается, чтобы определить, достижим и жив ли конкретный адресат.

На рис. 6.1 показана структура заголовка ICMP-пакета. Данному заголовку ICMP предшествует обычный IP-заголовок без поля опций (Options) и выравнивания (Padding), а поля TOS=0, Protocol=1.

| Тип (8битов) | Код (8 битов) | Контрольная сумма UDP (16 битов) |
|---|---|---|
| Содержимое зависит от типа и кода | | |

Рис. 6.1 — ICMP сообщение

Для всех сообщений ICMP первые 4 бита одинаковые. Остальная часть зависит от типа сообщения.

Поле type — 18 различных значений типа ICMP сообщений.

Поле code — детализирует сообщение.

Поле checksum охватывает все ICMP сообщения. Рассчитывается также, как и для IP заголовка. Является обязательным.

## 6.1 Утилита Ping

Название происходит от:

- Packet Internetwork Groper — пакетный межсетевой щуп.

Программа Ping предназначена для проверки доступности удаленного хоста. Программа посылает ICMP эхо запрос (Echo request=0/0) на хост и ожидает возврата ICMP эхо отклика (Echo reply=8/0).

Обычно, если Вы не можете послать Ping на хост, то не сможете получить доступ к этому хосту, используя Telnet или FTP. Помимо этого, с помощью Ping можно оценить время возврата пакета от хоста, что позволяет оценить удаленность хоста.

Параметры командной строки программы ping позволяют, например, задать:

– время ожидания ответа;
– интервал отправки запросов;
– количество пакетов (по умолчанию 4);
– размер пакетов;
– и другое.

При выполнении Ping для адреса локальной заглушки (127.0.0.1) проверяется установка и заглушка протокола TCP/IP.

В простейшем случае аргументом выступает IP-адрес или имя хоста.

```
$ ping  www.freebsd.org
PING  freebsd.org (204.216.27.21):56 data bytes
64 bytes from 204. 216.27.21: icmp_seq=0 ttl=235 time=731.099 ms
                                    •
                                    •
                                    •
64 bytes from 204. 216.27.21: icmp_seq=7 ttl=235 time=673.492 ms
9 packets transmitted, 7 packets received, 22 % loss.
Hound-trip  min/avg/max=377.568/559.207/855/513.
```

В выводе ping отчетливо видны потери пакетов и приход ответов в неупорядоченной последовательности, что говорит о том, что пакеты идут разными маршрутами.

Первый ответ занимает обычно больше времени, чем остальные, т.к. имеют место:

– запрос к DNS по имени (до отправки первого запроса);

– ARP-запрос, если нет в кэше (во время отправки первого запроса).

Если в командной строке указать — R, то в заголовок IP будет включена опция «Record Router» (запись маршрута) и можно будет увидеть путь пакетов «туда» и «обратно» (правда, это количество ограничено 9 шлюзами).

По умолчанию период посылки echo request — 1с. Если через 20 с не пришел ответ, то возвращается « host is down».

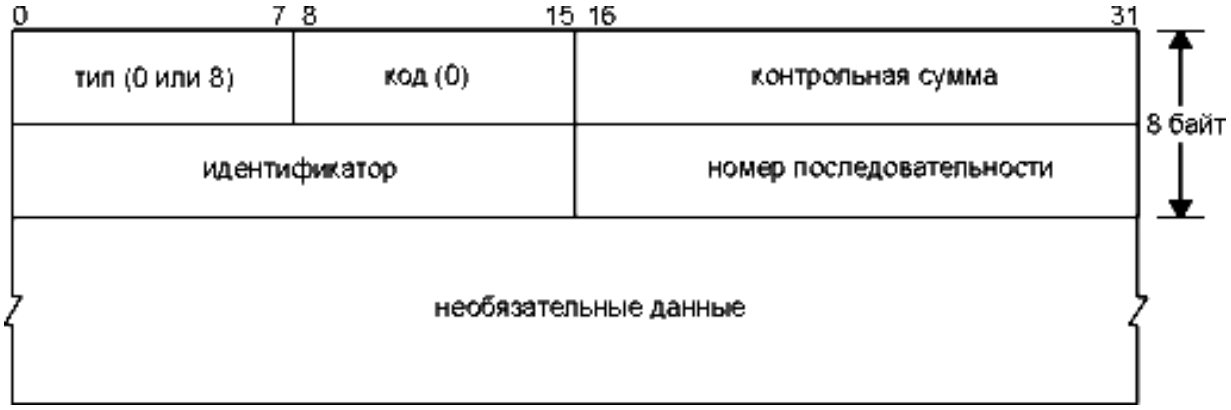



Рисунок 6.2 — Формат ICMP сообщения для эхо запроса и эхо отклика

Ping выводит принятый TTL. Хост обычно устанавливает его в max=255, но не всегда.

Вариант с TTL:

1. Не изменять.
2. Устанавливать TTL=255 (255 — N routers).
3. Другая величина.

Большинство реализаций TCP/IP поддерживают Ping сервер непосредственно в ядре — сервер не является пользовательским процессом. Так же, как в случае других ICMP запросов, в отклике сервера должны содержаться поля идентификатора (identifier) и номера последовательности (sequence number). Кроме того, любые дополнительные данные, посланные клиентам, должны быть отражены эхом.

В поле identifier — выставлен id процесса, отправляющего ICMP запрос. Это позволяет программе ping идентифицировать

вернувшийся ответ, если на одном и том же хосте в одно и то же время запущено несколько программ ping.

Номер последовательности начинается с 0 и увеличивается на единицу каждый раз когда посылается следующий эхо запрос. Ping печатает номер последовательности каждого возвращенного пакета, позволяя нам увидеть, потерялся ли пакет, поменялась ли последовательность движения пакетов и был ли пакет продублирован. Так как IP является ненадежным сервисом доставки датаграмм, любое из трех вышеперечисленных условий может появиться при работе программы ping.

Полезные ключи командной строки:

– c (число) — число пакетов переданных/принятых;
– I (wait) — ждать wait секунд между посылками;
– N — вывод в числовом виде (IP-адрес), указывает на то, что не надо преобразовывать адреса в имена при помощи DNS. Это несколько ускоряет работу;
– R — устанавливает опцию «Record router»;
– S (size) — размер данных в пакете ICMP (по умолчанию 56);
– V (verbose output) — выводить не только Echo reply.

## 6.2 Утилита traceroute

Программа Traceroute, написанная Van Jacobson, — отладочное средство, которое позволяет нам посмотреть маршрут, по которому двигаются IP датаграммы от одного хоста к другому. Хотя маршруты в общем случае могут быть разными, в большинстве случаев они будут совпадать.

Почему не используется опция записи маршрута RR (Если в командной строке указать — R, то в заголовок IP будет включена опция «Record Router» и можно будет увидеть путь пакетов «туда» и «обратно»):

1) не все маршрутизаторы исторически поддерживали эту опцию;

2) количество записей о маршрутизаторах в поле options ограничено его размером (40 байт), соответственно, максимум 9 адресов.

Traceroute использует в своей работе поле TTL. Когда шлюз принимает дейтаграмму с ttl=0 или ttl=1, он уничтожает ее и по-

сылает хосту, который ее отправил ICMP сообщение «время истекло» (time exceeded). Принцип работы Traceroute заключается в том, что IP дата — грамма, содержащая это ICMP сообщение, имеет в качестве адреса источника IP-адрес маршрутизатора.

Рассмотрим работу программы на примере рис. 6.3. На хост назначения отправляется IP датаграмма с TTL, установленным в единицу. Первый маршрутизатор, который должен обработать датаграмму, уничтожает ее (так как TTL равно 1) и отправляет ICMP сообщение об истечении времени (time exceeded). Таким образом, определяется первый маршрутизатор в маршруте. Затем Traceroute отправляет датаграмму с TTL равным 2, что позволяет получить IP-адрес второго маршрутизатора. Это продолжается до тех пор, пока датаграмма не достигнет хоста назначения.

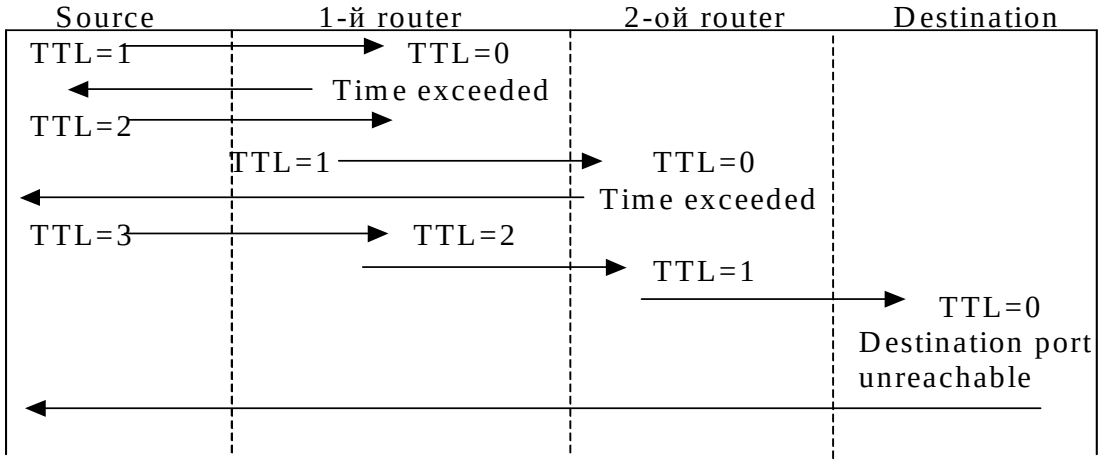



Рисунок 6.3

В UDP датаграммах, которые посылает Traceroute, устанавливается несуществующий номер UDP порта (больше чем 30000), что делает невозможным обработку этой датаграммы каким-либо приложением. Поэтому когда прибывает подобная датаграмма, UDP модуль хоста назначения генерирует ICMP сообщение «порт недоступен» (port unreachable). Все что необходимо в этом случае Traceroute, это определить тип принятого ICMP сообщения — либо об истечении времени (time exceeded), либо о недоступности порта (port unreachable) — именно таким образом мы узнаем, доставлена ли датаграмма в пункт назначения.

Флаги программы traceroute:

– g — установка Loose Source Routing (max 8 gateways);

– m — установка максимального TTL (по умолчанию 30);

– n — выводить IP адреса, а не имена;

– p — базовый порт (33434) base+n ports – 1;

– v (verbose) — вывод всех ICMP пакетов (не только time exceeded и port unreachable);

– w — задает время ожидания в секундах (5).

Рассмотрим пример работы программы Traceroute. Пройдем по маршруту от svr4 к slip через маршрутизатор bsdi.

```
svr4 % traceroute slip
traceroute to slip (140.252.13.65), 30 hops max, 40 byte packets
1 bsdi (140.252.13.35)  20 ms  10 ms  10 ms
2 slip (140.252.13.65)  120 ms  120 ms  120 ms
```

Первая строка без номера содержит имя и IP-адрес пункта назначения и указывает на то, что величина TTL не может быть больше 30. Размер датаграммы установлен в 40 байт, из которых 20 байт отводится на IP заголовок, 8 байт на UDP заголовок и 12 байт на пользовательские данные. (В 12 байтах пользовательских данных содержится номер последовательности, который увеличивается на единицу при отправке каждой следующей датаграммы, копия исходящего TTL и время, когда датаграмма была отправлена.)

Следующие две строки вывода начинаются с TTL, после чего следует имя хоста или маршрутизатора и их IP-адреса. Для каждого значения TTL отправляется 3 датаграммы. Для каждого возвращенного ICMP сообщения рассчитывается и печатается время возврата (round-trip). Если ответ не получен в течение пяти секунд на любую из трех датаграмм, печатается звездочка, после чего отправляется следующая датаграмма. В нашем примере первые три датаграммы имели TTL, установленный в единицу, а ICMP сообщения вернулись через 20, 10 и 10 миллисекунд. Следующие три датаграммы были отправлены с TTL равным 2, а ICMP сообщения вернулись с задержкой 120 миллисекунд. Так как TTL со значением 2 достигло конечного пункта назначения, программа прекратила свою работу.

Сравнение флагов командной строки утилит ping и traceroute (tracert). Для операционных систем FreeBSD и Windows 98.

| | Win | Unix | Описание |
|---|---|---|---|
| PING | -t | По умолч. | Непрерывная посылка зондирующих пакетов |
| | -a | -n | Не разрешать адрес в имена |
| | -n N | -c N | Число отправляемых пакетов |
| | -l N | -s n | Число байт данных в пакете (в Unix – root only, >1) |
| | -f | | Установить флаг DF (не фрагментировать) |
| | -i N | | Задать значение поля TTL=N в IP-заголовке |
| | -v N | | Задать явно поле TOS=N |
| | -r N | -R | Установить опцию Record Route (N — число фиксированных точек) |
| | -s | | Установить опцию Timestamp |
| | -j список | | Установить опцию Loose Source Routing |
| | -k список | | Установить опцию Strict Source routing |
| | -w N | | Интервал ожидания ответа |
| TRACEROUTE | -d | -n | Не разрешать адрес в имена |
| | -h N | -m N | Ограничить max значение TTL числом N (число хостов 30) |
| | -j список | -g список | Установить LSRR (до 8 gateways) |
| | -w N(ms) | -w N(ms) | Установить время ожидания (5c) |
| | | -p port | Установить базовый порт (33434 –default) base + n host – 1 |
| | | -v | Verbose — подробный вывод |
| | | -t N | Установить поле TOS в N |

# 7 СИСТЕМА ДОМЕННЫХ ИМЕН DOMAIN NAME SYSTEM (DNS)

Основное назначение DNS — предоставление возможности для каждой машины в сети узнать по символьному (доменному) имени адрес другой машины, а по IP-адресу — имя.

Например, команда ftp://192.45.66.17 будет устанавливать сеанс связи с нужным ftp-сервером, а команда http://203.23.106.33 откроет страницу на Web-сайте. Однако пользователи обычно предпочитают работать с символьными именами комьютеров. Следовательно, в сетях TCP/IP должны существовать символьные имена хостов и механизм для установления соответствия между символьными именами и IP-адресами.

На раннем этапе развития Internet (1970 г.) имена всех хостов содержались в одном файле HOSTS.TXT, который велся централизованно в Стэндфордском Исследовательском Институте (SRI) и распространялся с хоста SRI — NIC.

С ростом числа хостов в ARPANET возникли следующие проблемы:

– значительный рост трафика и загрузка процессора, связанные с распространением файла hosts.txt.;

– коллизии имен — никто не мог запретить кому-либо добавить в файл hosts.txt. на локальной машине уже существующее имя;

– согласованность — все труднее было поддерживать ее в пределах растущей сети. Хосты меняли адреса, добавлялись новые и т.д.

Для разрешения всех этих проблем и была разработана доменная служба имен DNS (RFC 1034 и 1035).

Суть системы DNS заключается в иерархической схеме имен, основанной на доменах и распределенной базе данных, реализующих эту схему. Данные каждого домена доступны из всей сети с помощью механизма «клиент-сервер». Программы, называемые серверами имен (name servers, NS), обеспечивают серверную часть механизма «клиент-сервер». Они содержат информацию о некотором сегменте DNS и делают ее доступной для клиентов, которые, в свою очередь, называются резолверами (resolvers) или «разрешителями». Чаще всего резолвер — это на-

бор библиотечных подпрограмм, которые умеют создавать запросы, отправлять их на серверы имен и принимать ответы.

В качестве транспорта DNS обычно использует протокол UDP. Сервер имен, как правило, работает на порту 53.

## 7.1 Пространство имен DNS

Вся база данных DNS представляет собой инвертированное дерево с корневым узлом наверху. Дерево имен начинается с корня, обозначаемого здесь точкой (.). Затем следует старшая символьная часть имени, вторая по старшинству символьная часть имени и т.д. Младшая часть имени соответствует конечному узлу сети. Составные части имени отделяются друг от друга точкой. Пример (рис. 7.1): rk.tusur.ru. — последняя точка указывает на то, что имя является «абсолютным» или FQDN (Fully Quilted Domain Name — полное доменное имя). Если точки нет, то требуется дополнение имени до FQDN. Алгоритм такого дополнения зависит от используемого резолвера.

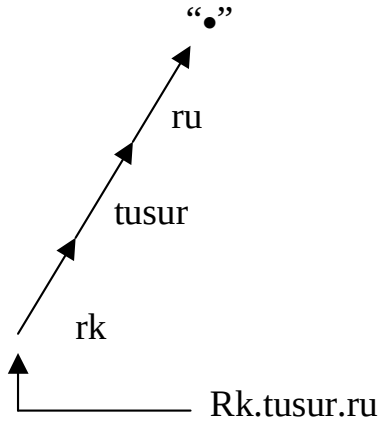



Рисунок 7.1

Каждый узел дерева имеет текстовую метку. Каждый узел также является корнем нового дерева. Каждое из этих поддеревьев представляет собой части всей базы данных — домены в DNS. Каждый домен может в дальнейшем быть поделен на дополнительные разделы — называемые поддоменами *(subdomain)*. Каждый домен имеет уникальное имя. Доменное имя в домене идентифицирует его позицию в базе данных. В DNS доменное имя — это последовательность меток, начиная от метки возле корня это-

го домена до корня всего дерева, разделенных символом «точка» («.»).

В Internet корневой домен управляется центром InterNIC. Домены верхнего уровня назначаются для каждой страны (двухбуквенный домен типа ru, su, us, uk, tv и т.д.), а также на организационной основе.

com. — коммерческие организации (microsoft.com).

edu. — образовательные учреждения (berkeley.edu).

gov. — правительственные организации (whitehouse.gov).

mil. — военные организации США (army.mil).

net. — организации, обеспечивающие работу сети (ripn.net).

org. — некоммерческие организации (freebsd.org).

int. — международные организации (nato.net).

Одной из главных задач внедрения DNS являлась децентрализация администрирования. Каждый домен администрируется отдельной организацией. Каждая организация, владеющая доменом, может разбить его на поддомены и передать ответственность за них другим организациям (например, филиалам или клиентам, так InterNIC делегировал свои полномочия распределения имен в России организации РосНИИРОС, которая отвечает за делегирование имен поддоменов в домене RU.) Иллюстрируется на рисунках 7.2 и 7.3.

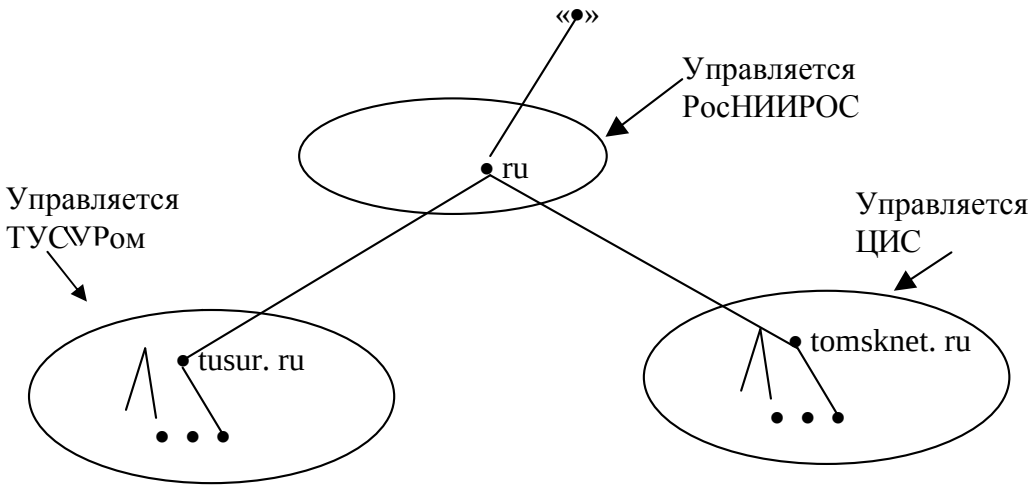



Рисунок 7.2

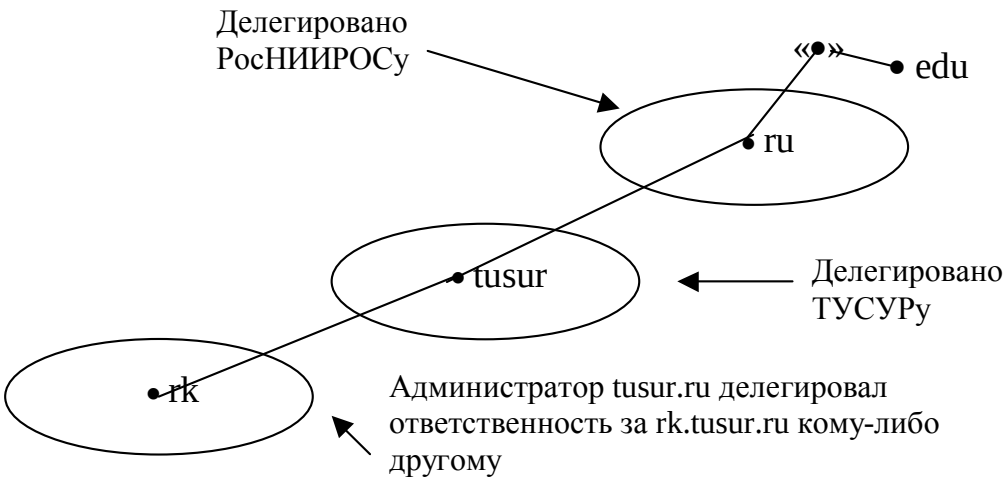



Рисунок 7.3

Некоторые общие правила:

1. Длина метки — до 63 символов.

2. Иерархия доменов — до 127 уровней.

3. Общая длина доменного имени — 255 символов. Все нюансы можно найти в RFC 1032-1035 и в тех RFC, которые выходят как обновления указанных документов.

4. Прописные и строчные буквы в доменных именах не различаются, хотя и выглядят так, как изначально определены. Например, имя Rk.TuSur.Ru эквивалентно имени rk.tusur.ru с точки зрения DNS.

## 7.2 Серверы имен и зоны

Сервер имен (NameServer, NS) — это программа, хранящая и выдающая по запросу информацию о части пространства доменных имен. Обычно NS имеет полную информацию о некоторой части пространства доменных имен, называемую **зоной**, которую сервер загружает из дискового файла или с другого NS. Такой NS называется **авторитетным** за такую зону. NS одновременно может быть авторитетен за несколько зон.

Различие между доменом и зоной очень важное, но, в то же время, весьма тонкое. Зона, в общем, содержит те же доменные имена, что и одноименный домен, за исключением доменных имен в делегированных поддоменах.

Если поддомен домена не делегирован вовне, зона будет содержать наряду с именами зоны также имена такого поддомена (см. рисунок 7.4). Поддомены rectorat.tusur.ru содержатся в той же зоне, что и tusur.ru. Зоны rk.tusur.ru и fet.tusur.ru администрируются отдельно (см. рисунок 7.5).

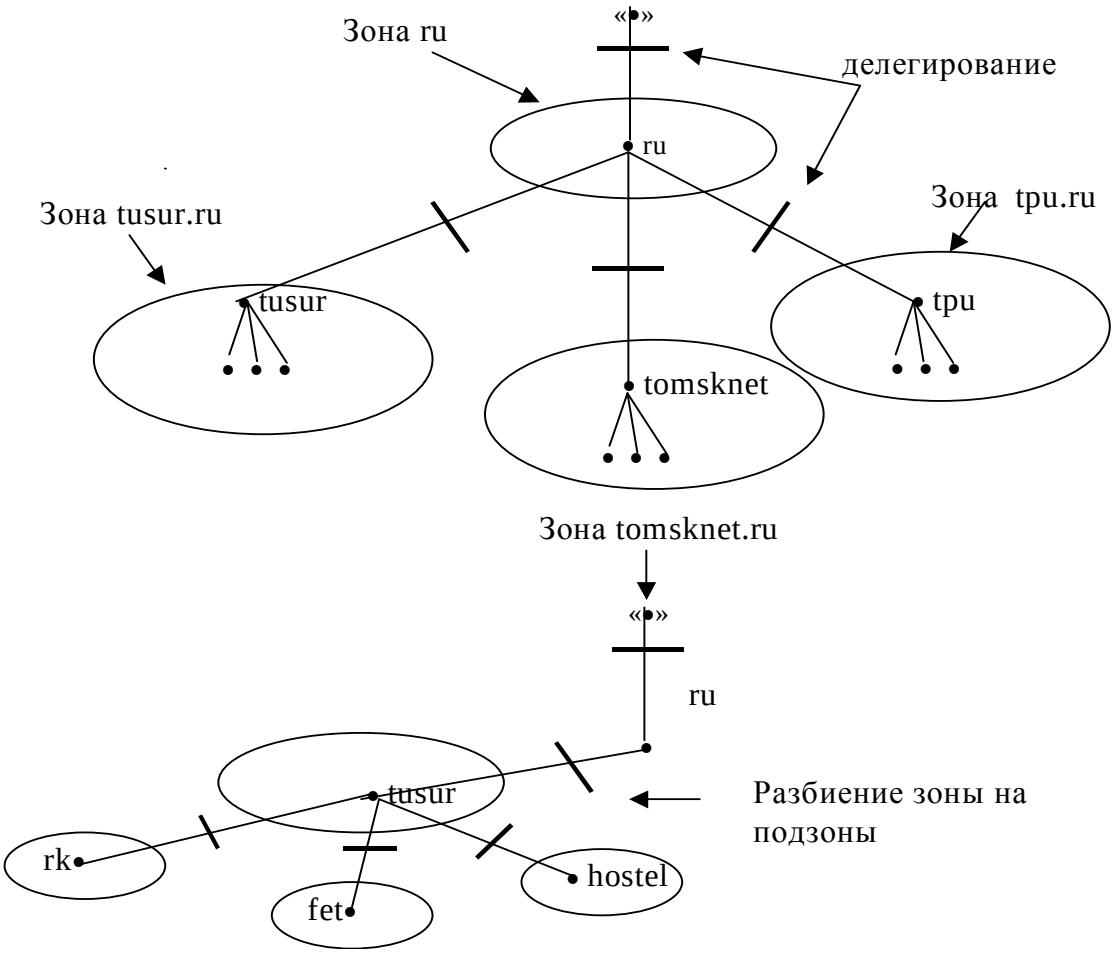



Рисунок 7.4

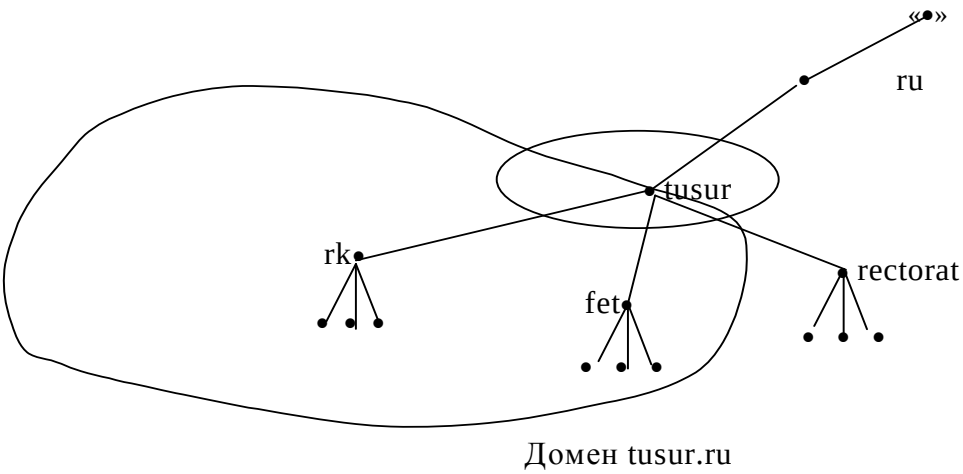



Рисунок 7.5

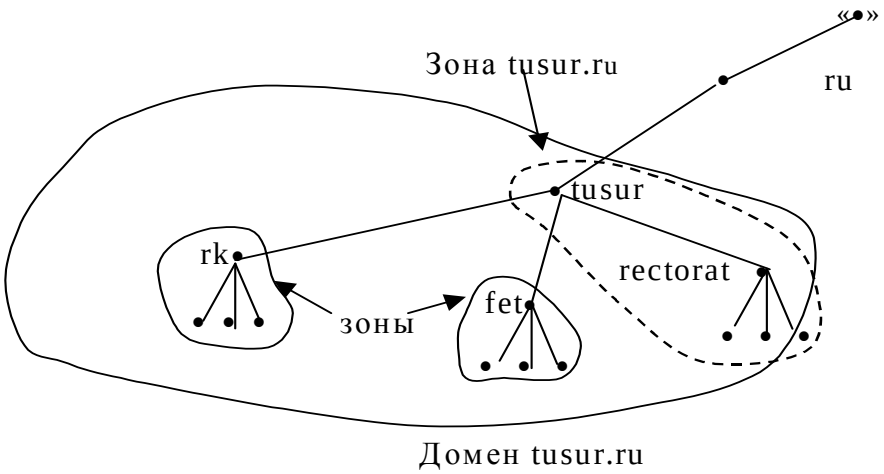



Рисунок 7.6

Домен может содержать информацию больше, чем требуется от данного NS. Поэтому NS загружает не весь домен, а только свою зону. Поскольку зона ограничена делегированием, она никогда не включает в себя неделегированных данных. Представьте, что корневой сервер имен загружает не корневую зону, а корневой домен: он должен был бы загрузить все пространство имен! В простых случаях (небольшой домен) весь домен может содержаться в одной зоне. Отсюда следует, что когда вы делеги-

руете часть домена, ваши данные вместо полной информации о поддомене, включают указатель на сервер имен, который является авторитетным за интересующий поддомен. Если у вашего NS спрашивают об этом поддомене, он может ответить правильным списком серверов имен, авторитетных за испрашиваемые данные.

## 7.3 Типы серверов имен (NS)

Стандарт DNS определяет 2 типа NS:

1. Primary master (PM) — читает данные зоны с диска.

2. Secondary master (SM) — берет данные о зоне с другого сервера имен: или с primary master, или с другого secondary master.

Первичный и вторичный сервера должны быть независимы и избыточны таким образом, чтобы система DNS не вышла из строя при отказе одного из серверов.

Процесс передачи информации от первичного сервера (PM) вторичному (SM) называется передачей зоны (zone transfer). Когда в зоне появляется новый хост, администратор добавляет соответствующую информацию (минимум, имя и IP-адрес) в дисковый файл на PM. После чего SM уведомляется о необходимости повторно считать свои конфигурационные файлы. SM регулярно опрашивают PM (обычно каждые 3 часа), и если PM содержат новую информацию, вторичный получает ее с использованием передачи зоны.

## 7.4 Резолверы (Resolvers)

Резолверы — это клиенты, которые обращаются к серверам имен.

Программы, работающие на хосте и нуждающиеся в разрешении имен, включают в себя библиотечные программы резолвера.

Резолвер выполняет следующие действия:

– запрашивает сервер имен, который указан в его конфигурационном файле (/etc/resolv.conf — в Unix);

– интерпретирует ответ (это может быть либо запись о ресурсах, либо ошибка);

– возвращает информацию запрашиваемой программе.

Резолвер в описываемом случае — это не отдельный процесс. Все что умеет такой резолвер — спросить, подождать ответа, возможно, переспросить по тайм-ауту или ошибке. Всю работу по поиску данных в DNS выполняет тот сервер имен, который спрашивает резолвер. Существуют более умные резолверы, умеющие, например, кэшировать информацию.

## 7.5 Процесс разрешения имен (Resolution)

Процесс поиска информации в DNS называется разрешением имени или просто разрешением. Для того, чтобы этот процесс был возможен, NS, по сути, должен лишь спросить корневой сервер о направлении поиска, а тот, а свою очередь, направит его в нужном направлении.

Корневые (root) NS знают, где находятся NS, авторитетные за домены верхнего уровня *(хотя, фактически, большинство NS авторитетны за gTLDs)*. Будучи спрошенным о любом доменном имени, корневой NS генерирует отсылку — дает-IP адрес NS, авторитетного за тот домен верхнего уровня, в котором находится искомое доменное имя. В свою очередь, NS верхнего уровня выдает список NS, авторитетных за домен второго уровня, в котором находится искомое имя.

Корневые NS очень важны для разрешения имен, поэтому чтобы снять с них лишнюю нагрузку в DNS предусмотрен специальный механизм — кэширование. Но, в отсутствие другой информации, разрешение начнется с корневого сервера. Если корневые сервера не будут доступны длительное время, DNS перестанет работать. Поэтому имеется 13 корневых NS.

Запросы бывают 2-х видов: рекурсивные и повторяющиеся (итеративные). В случае рекурсивного запроса к NS, он обязан сам найти и вернуть ответ или вернуть ошибку о несуществующем домене (современные NS могут быть сконфигурированы на отклонение рекурсивных запросов). Если спрошенный NS не авторитетен за искомые данные, он может спросить другие NS. Он может послать рекурсивный запрос, обязав их найти ответ. Он также может послать итеративный запрос и получить отсылку. В настоящее время реализуется, как правило, вторая схема.

NS, приняв рекурсивный запрос, на который не может ответить, спросит у ближайших известных NS, авторитетных за зону, ближайшую к искомому доменному имени.

Пример разрешения имени tor.rk.tusur.ru. (139.121.192.212) приведен на рисунке 7.7.

В данном случае резолвер издает запрос к своему NS и ждет ответа только от него. Тот выполняет всю работу: получив подряд три отсылки, он, наконец, спрашивает у сервера, авторитетного за rk.tusur.ru, и получает ответ (резолвер издал рекурсивный запрос). NS проверит, известен ли ему NS, авторитетный за tor.rk.tusur.ru, если да, то он пошлет запрос такому серверу, если нет, то проверит то же для rk.tusur.ru, tusur.ru, и, наконец, ru. По умолчанию, гарантированно можно обратиться к NS корневого домена и получить отсылку в нужном направлении.

Такая схема сокращает до минимума время разрешения. NS для tusur.ru, приняв запрос о tor.rk.tusur.ru, не будет спрашивать корневые NS. Он использует информацию о делегировании, содержащуюся в его зоне по поводу rk.tusur.ru (рисунок 7.8).

Корневые NS, а также серверы доменов верхнего уровня в принципе не обрабатывают рекуррентных запросов. Нагрузка на них и без того велика.

В случае последовательного разрешения, NS выдает лучший ответ, который ему уже известен (рисунок 7.9).

Существенно ускоряет процесс разрешения имен кэширование, выполняемое на NS. Смысл заключается в том, что перед тем, как начать поиск «по полной программе», NS смотрит в свой кэш и ищет ближайшее соответствие. Кэширование может быть как положительное, так и отрицательное (нет такого доменного имени).

Допустим мы уже имеем адрес fet.tusur.ru. В процессе разрешения были кэшированы имена и адреса серверов имен для fet.tusur.ru и tusur.ru. Допустим, наш NS имеет сведения о tor.rk.tusur.ru. Он распознает tusur.ru как ближайший предок tor.rk.tusur.ru, о котором ему известно и начнет поиск именно с NS для tusur.ru. Также, если сервер в ходе предыдущего запроса обнаружил отсутствие данных по fet.rk.tusur.ru, он закэширует отрицательный ответ на 10 минут и будет отвечать на последующие запросы, основываясь на данных своего кэша.

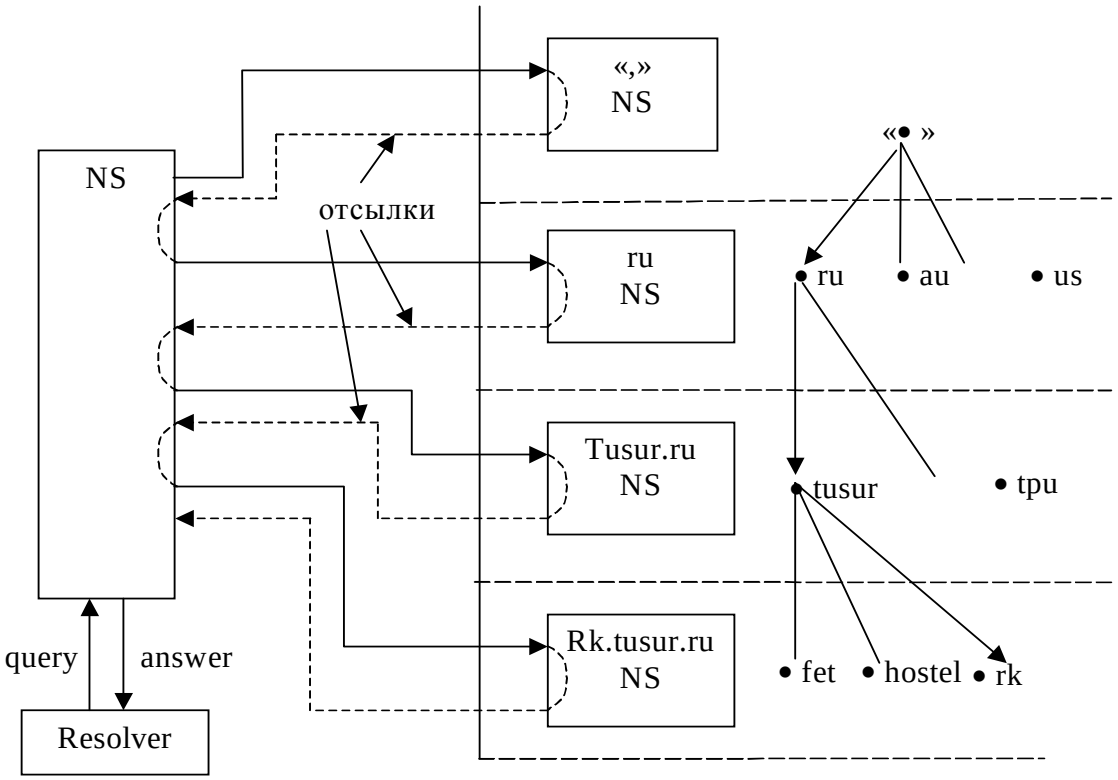



Рисунок 7.7

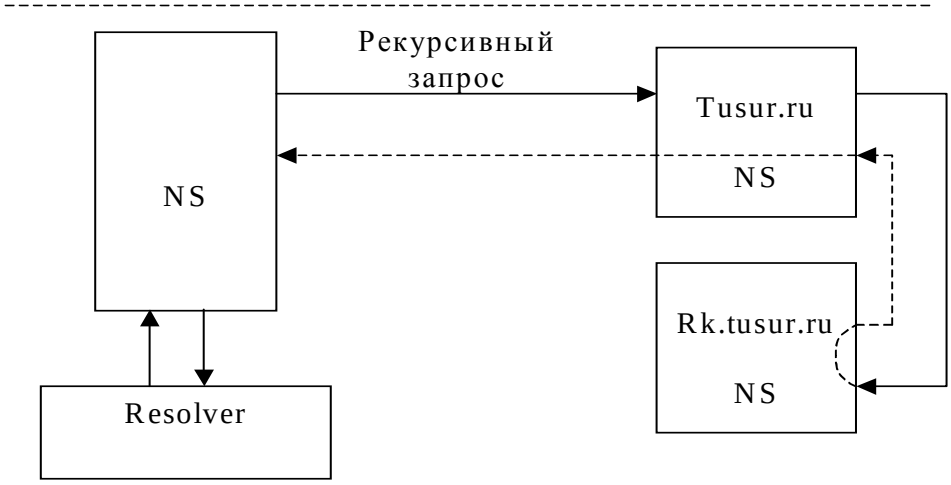



Рисунок 7.8

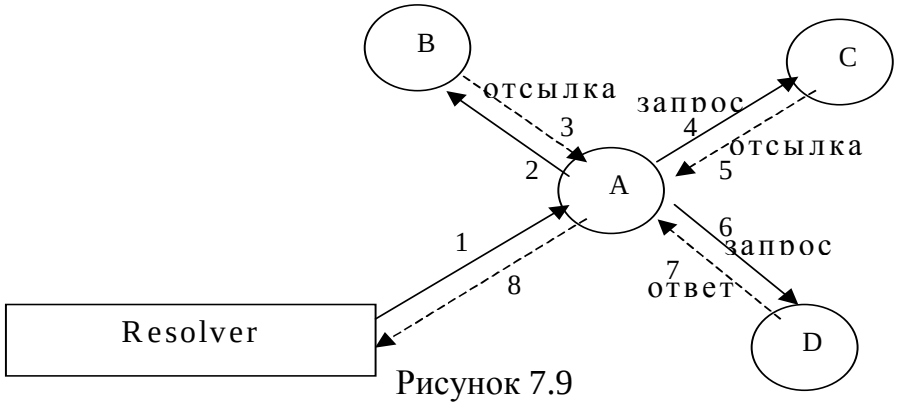
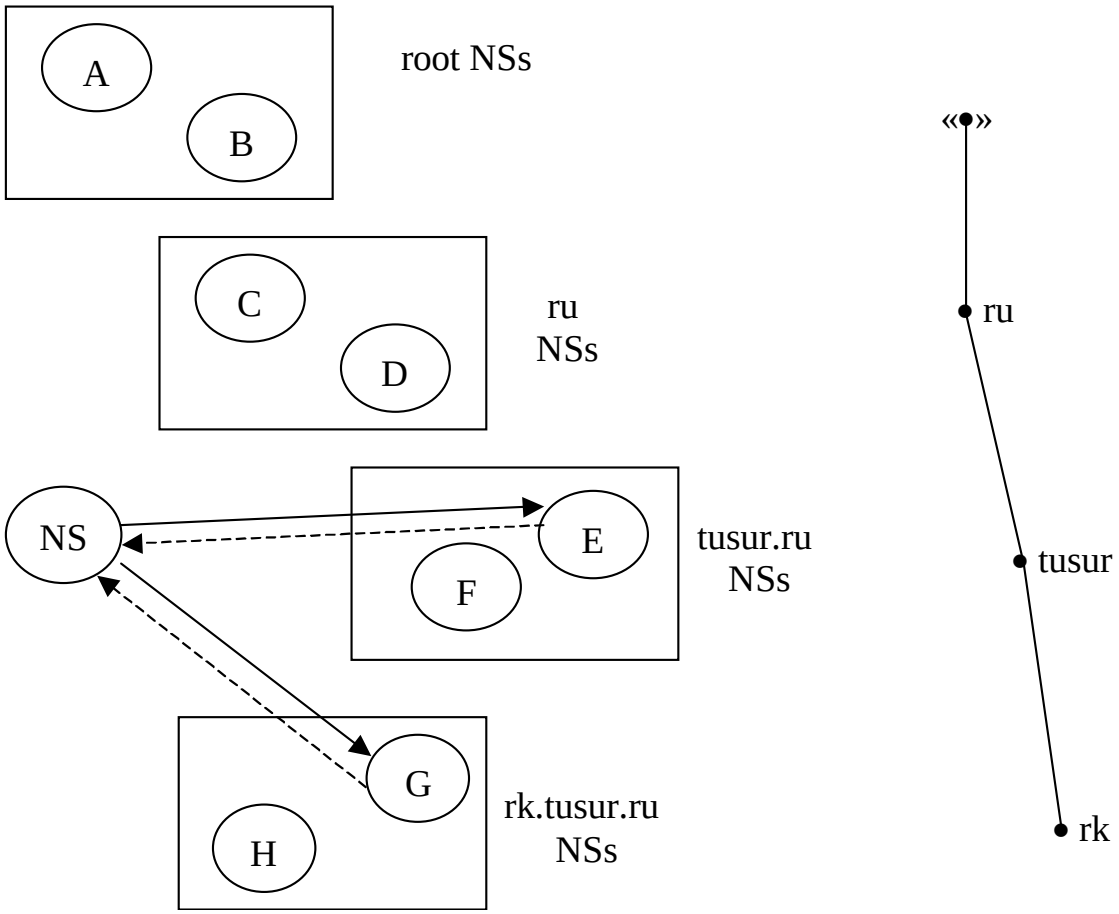



Рисунок 7.9

Рассмотрим пример, приведенный на рисунке 7.10.




1. Запрос о tor.rk.tusur.ru (NS кэширует G,H).
2. Отсылка к G и H.
3. Запрос о tor.rk.tusur.ru.
4. Ответ: адрес tor.rk.tusur.ru (NS кэширует его).

Рисунок 7.10

Кэширование, кроме того что ускоряет процесс разрешения имен, снимает нагрузку с корневых и большинства NS верхнего уровня. NS не могут кэшировать данные навсегда, поскольку данные зон меняются. Администратор зоны устанавливает для записей своей зоны TTL (time-to-live) — время, в течение которого эти данные могут быть кэшированы другими NS. Выбор TTL — компромисс между скоростью и целостностью данных. Мал TTL — хорошо с точки зрения целостности. Велик — хорошо с точки зрения нагрузки на сеть и NS. Следует отметить, что:

• Кэшированные ответы никогда не бывают авторитетными.

• Корневые серверы имен никогда не кэшируют никакой информации.

## 7.6 Разрешение адресов в имена. Реверсная зона DNS

Проблема: дан адрес, найти имя.

Поиск в файле hosts.txt был бы тривиален, но в случае с DNS, которая является базой данных, проиндексированной по доменным именам, такой поиск весьма непрост, если просматривать подряд все домены. Это может занять огромное время.

Было найдено оригинальное решение этой проблемы: почему бы в качестве доменных имен не использовать компоненты IP-адреса? Был введен специальный домен in-addr.arpa:

Размещение первых байтов сетевого адреса выше по дереву имен делает возможность делегирования полномочий вдоль всей линии сетевых адресов.

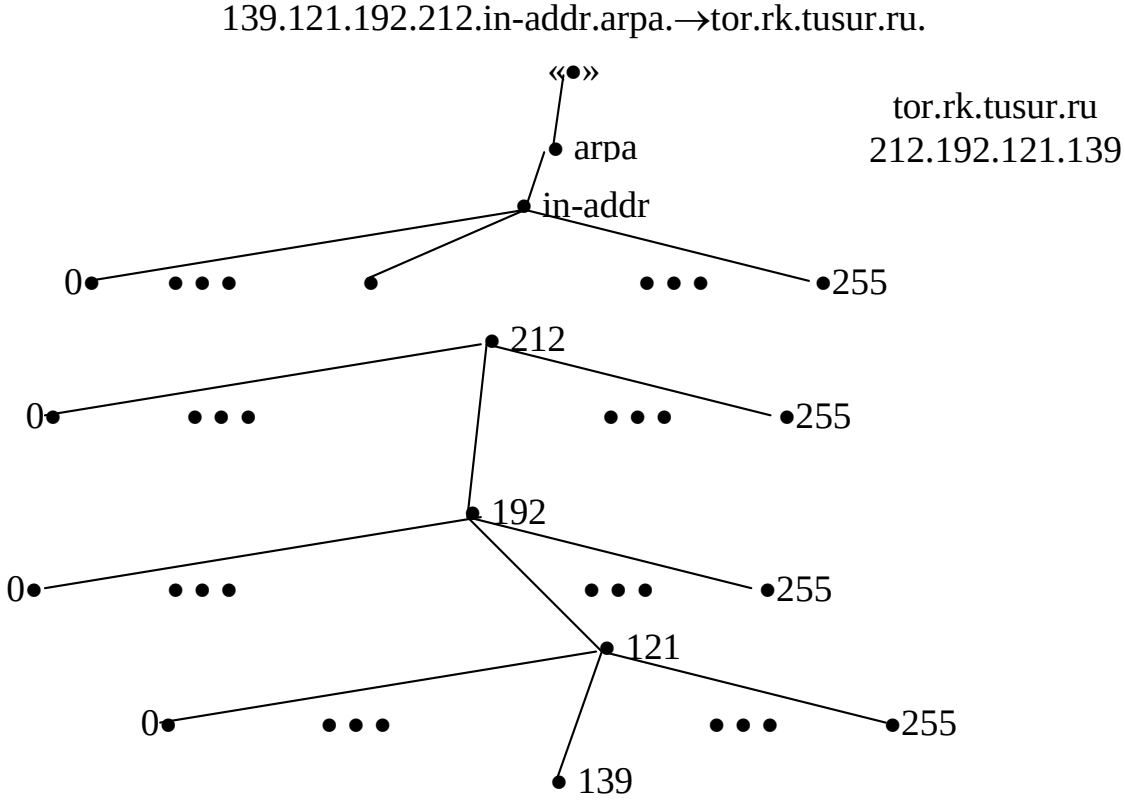



Рисунок 7.11

212.in-addr.arpa — домен, содержащий обратную информацию о компьютерах, чьи адреса начинаются с 212. Если бы порядок байтов был обратный — системы не было бы.

tor.rk.tusur.ru

139.121.192.212.in-addr.arpa

более подробно ← → менее подробно

Когда организация получает IP-сеть, она получает соответствующий домен в in-addr.arpa. Вообще, делегирование обратных доменов проводится не менее, чем для сети класса C, но существуют способы (RFC 2317) это ограничение обойти.

**Внимание!** Прямая и реверсная зоны — это совершенно **разные** зоны и могут администрироваться разными организациями.

Многие серверы при приеме сетевого соединения проверяют IP-адрес клиента по реверсной зоне, а затем найденное имя — по прямой. В случае несовпадения вероятен spoofing — подмена адреса. В соединении отказывают.

## 7.7 Типы записей о ресурсах DNS

SOA — start of authority — «начало полномочий» определяет начало зоны.

- Serial — «Серийный номер» зоны. Служит для синхронизации slave и master серверов.
- Refresh — период времени, через который slave сверяет serial зоны с master′ом.
- Retry — при не ответе master′а — периодичность повторных попыток синхронизации зоны.
- Expire — период времени, по истечении которого slave будет считать свои данные о зоне окончательно устаревшими и перестанет отвечать на вопросы по зоне.
- Minimum TTL — минимальное время удержания RR зоны в кэшах DNS серверов. Для каждой RR, тем не менее, можно явно указать TTL. Если не указывать, то используется значение minimum из SOA.

NS (name server) — авторитетный сервер имен для зоны. В файле зоны играет вспомогательную информативную роль. Важ-

ное значение имеет запись связка (NS, A ) в файле родительской зоны.

A — адрес IP хоста. Одному имени может соответствовать несколько записей т.А. Если резолвер и NS в одной сети, то некоторые NS помещают ближайший адрес первым (выдаются все адреса).

«Round robin» — «address sorting». Если не надо, то адрес ротируется.

CNAME — определяет для указанного имени его каноническое имя. Полезно для применений типа общеизвестных сервисов (ftp, www, mail, news и т.д.). Когда резолвер издает запрос на некоторое имя (типа A?), которое является алиасом, NS сначала находит его каноническое имя, затем разрешает уже из этого имени адрес и возвращает резолверу. Алиас не желательно применять в RR типа NS и MX.

MX (mail exchanger) — почтовый концентратор, используется системами электронной почты. Почта, предназначенная для доменного имени слева от MX, направляется на хост справа на MX с наименьшим значением приоритета (меньше значение — выше приоритет).

HINFO (Host information) — информация о хосте (тип машинной и ОС), из соображений безопасности используют редко или помещают в RR уклончивые сведения (или ложные).

TXT — произвольный текст, обычно носит пояснительный характер. Может содержать данные о расположении хоста, ответственной персоне и т.д. Также используется не очень часто.

WKS (Well Known Services) — хорошо известные сервисы. Поясняют, какие службы запущены на указанном хосте. Практически не используется по соображениям безопасности.

Существуют и другие типы RR (всего около 25) и появляются новые. Но DNS достаточно консервативная система и встречаются в основном указания.

PTR — указатель на доменные имена. Используется только в обратной зоне для построения соответствий адрес → имя.

## 7.8 Взаимодействие NS и резолвера

Серверы имен работают на протоколе UDP в качестве транспорта, прослушивая порт SS (udp). Каждое сообщение содержит только один запрос или ответ (ответ может содержать несколько RR, это зависит от запроса и содержимого базы DNS по заданному имени).

Резолвер издает запросы с эфемерного порта (выше 1024) и ждет на него ответа. Когда сервер спрашивает другой сервер, используются только 53 порт с обеих сторон. Во время зонных пересылок используется TCP, поскольку объем передаваемой информации в этом случае достаточно большой.

В конфигурации резолвера указывают:

domain movie.edu

search movie.edu, comedy.movie.edu

nameserver 192.249.249.3

nameserver 192.249.249.1

Резолвер взаимодействует с NS в порядке, указанном в файле конфигурации. В случаях тайм-аутов или ошибок — запрашивается следующий сервер. Если ответивший сервер вернул отрицательный ответ (NX domain), то другой NS не переспрашивается. Тайм-аут обычно составляет 5 секунд. Число ошибок — 4 (пороговое).

## 7.9 Инструменты диагностики DNS

В Unix системах используются программы:

1. nslookup
2. host
3. dig

Кроме того, существуют различные средства для облегчения администрирования DNS типа dnswalk и других.

Наиболее часто применяют утилиту nslookup — она работает в интерактивном или в неинтерактивном режимах.

Интерактивный: $ nslookup,

$ nslookup — 0.0.0.0.

IP-адрес NS, которому будут направляться запросы.

Неинтерактивный:

$nslookup [-options] host-to-find  [server],
$nslookup [-options] — интерактивный, используется NS по умолчанию,
$nslookup [-options]  server — интерактивный, использует server в качестве NS,
$nslookup [-options]  host — поиск host, default server,
$nslookup [-options]  host server — поиск host, используя указанный NS.

# 8 ТРАНСПОРТНЫЙ УРОВЕНЬ СТЕКА ПРОТОКОЛОВ TCP-IP. ПРОТОКОЛ TCP (RFC 793)

TCP — Transmission Control Protocol — протокол управления передачей.

Значение поля Protocol в заголовке IP — 6.

## 8.1 Сервис, предоставляемый TCP

Хотя TCP и UDP находятся на одном уровне стека TCP/IP (транспортном), TCP предоставляет сервис, полностью отличный от UDP. TCP поддерживает надежную передачу потока данных с предварительным установлением связи между отправителем и получателем *(connection-oriented)*. Это значит, что перед тем, как начать обмениваться данными, два приложения, использующие в качестве транспорта TCP, должны установить соединение (создать виртуальный канал).

Надежность TCP достигается при помощи следующих действий:

• Данные приложения разбиваются на порции, размер которых TCP оптимизирует. Это отлично от UDP, когда каждая запись приложением N байт в сеть вызывает генерирование UDP-сообщения такого же размера. Блок информации, передаваемый TCP, называется сегментом.

• Когда TCP отправляет сегмент, он включает таймер, ожидая подтверждения получения этого сегмента другой стороной. Если по истечении времени подтверждения не получено, TCP проводит ретрансляцию этого сегмента.

## 8.2 Заголовок TCP

Инкапсуляция TCP-сегмента

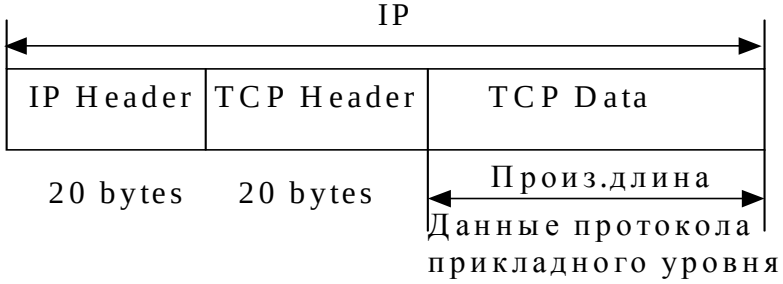



Рисунок 8.1

Структура TCP заголовка изображена на рисунке 8.2.

<table>
<tr><td colspan="8">Порт источника (16)<br>Source port number</td><td>Порт назначения (16)<br>Destination port number</td></tr>
<tr><td colspan="9">Порядковый номер (32)<br>Sequence number</td></tr>
<tr><td colspan="9">Номер подтверждения (32)<br>Acknowledgment number</td></tr>
<tr><td>Длина данных (4)<br>Header length</td><td>Резерв (6)<br>Re-served</td><td>U<br>R<br>G</td><td>A<br>C<br>K</td><td>P<br>S<br>H</td><td>R<br>S<br>T</td><td>S<br>Y<br>N</td><td>F<br>I<br>N</td><td>Размер окна (16)<br>Window size</td></tr>
<tr><td colspan="8">Контрольная сумма (16)<br>TCP checksum</td><td>Указатель срочности (16)<br>Urgent pointer</td></tr>
<tr><td colspan="9">Опции (0 и более 32-разрядных слов)<br>Options</td></tr>
<tr><td colspan="9">Данные (32)</td></tr>
</table>

Рисунок 8.2

• Каждый сегмент TCP в заголовке содержат номера портов источника и приемника (source и destination). Две эти величины вместе с IP-адресом источника и приемника уникальным образом идентифицируют каждое соединение.

Комбинация номера порта и IP-адреса называется сокетом (socket). Пара сокетов **client IP, client port : server IP, server port** однозначно и уникально идентифицируют каждое TCP-соединение в Internet.

• Поле **Порядковый номер** (Sequence Number — номер первого байта сегмента) используется для проверки того, что все блоки данных получены. Если принятый порядковый номер не соответствует очередности и срабатывает таймер TCP, все не подтвержденные блоки данных должны быть переданы повторно. Диапазон значений — $0 \div 2^{32}–1$, а затем счет начинается сначала.

• Поле **Номер подтверждения** (Acknowledgement Number) следует за порядковым номером и идентифицирует порядковый номер следующего ожидаемого блока данных. Передается принимающей стороной для подтверждения количества успешно принятых байтов. TCP предоставляет полнодуплексное соединение, это означает, что данные могут идти в обоих направлениях

одновременно. Но каждая сторона отслеживает Sequence Number данных, идущих в каждом направлении.

• Поле **Длина данных** определяет, где начинаются данные заголовка TCP, то есть сколько 32-битных слов находится в заголовке, предшествующем полю данных пользователя.

• Поле **Резерв** (Reserved) — зарезервированы 6 бит (= 0).

• TCP-заголовок содержит 6 флагов (поле flags). Одновременно могут быть установлены один или несколько флагов (соответствующий бит = 1).

Флаги TCP-заголовка:

**URG** — указывает, что сегмент содержит срочные данные и поле. **Указатель срочности (**Urgent Point) указывает их положение в сегменте.

**ACK** — указывает, что заголовок содержит подтверждение ранее полученных данных в поле **Подтверждение данных** (Acknowledgement Number).

**PSH** — приемник, должен как можно скорее передать эти данные приложению.

**RST** — сброс соединения. Безусловное уничтожение виртуального канала.

**SYN** — указывает на то, что сегмент является управляющим сообщением, являющимся частью процедуры установления связи.

Бит SYN применяется для установки соединения. У запроса соединения бит SYN=1, бит ACK=0. В ответе содержится подтверждение, поэтому значения битов равны:

SYN=1, ACK=1

**FIN** — указывает, что эта сторона прекращает передачу данных и желает закрыть виртуальный канал.

• Поле **Размер окна** (Window size) служит для управления потоком данных. Стороны должны согласовать, какое число блоков данных может быть передано до подтверждения. Это число называется размером окна (ограничено значением $2^{16} – 1 = 65535$).

• Поле **Контрольная сумма** (TCP checksum) рассчитывается по всему сегменту (заголовок + данные). Рассчитывается так же, как и для UDP, с использованием псевдозаголовка. Это поле обязательно.

• Поле **Указатель срочности** (Urgent Point) действителено только если выставлен флаг URG. Это положительное смещение, добавляемое к значению в поле Sequence Number, чтобы получить Sequence Number последнего байта срочных данных.

• Поле **Опции** (Options) присутствует не во всех сегментах. Как правило, присутствует в сегментах, инициализирующих соединение. Например, опция mss определяет максимальный размер сегмента, который приемник готов принять.

Не всегда TCP-сегмент содержит данные: на этапах установления и завершения соединения он их чаще всего не содержит.

TCP, таким образом, представляет надежный, потоковый сервис транспортного уровня с предварительным установлением соединения, реализующий алгоритм скользящего окна и не предполагающий негативных или избирательных подтверждений (только положительные).

Особенности работы TCP:

• Когда TCP принимает данные от другой стороны он отправляет ей подтверждение (acknowledgement). Обычно это происходит с задержкой на доли секунды, а не сразу.

• TCP передает в каждом сегменте контрольную сумму. Если в принятом сегменте контрольная сумма не совпадает с пересчитанной, сегмент отбрасывается.

• Поскольку TCP сегменты прибывают в IP-дейтаграммах, они могут прибывать в неверном порядке. Принимающая сторона TCP восстанавливает правильный порядок байтов в принятых данных.

• Поскольку IP-дейтаграммы могут дублироваться, TCP должен отбрасывать дубли.

• TCP имеет механизм управления потоком данных. Каждая сторона TCP-соединения имеет конечный размер буфера приема. Поэтому принимающая сторона позволяет передающей стороне передавать приемлемое количество данных в зависимости от наличия свободного места в буфере. Это предотвращает переполнение буфера удаленного хоста при TCP-обмене с быстрым хостом.

Во время TCP-сессии между двумя приложениями имеет место поток байтов. Никаких маркеров TCP в поток не вставляет. TCP не интерпретирует содержимое потока байтов. Это делает приложение, для которого они предназначены.

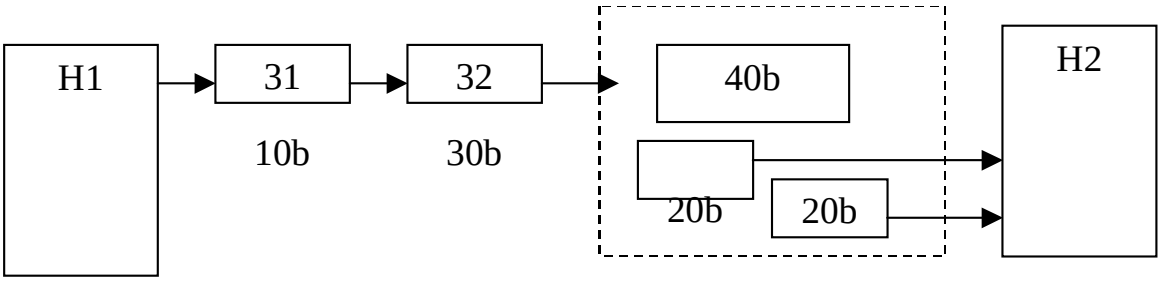



Рисунок 8.3 — Передача потока байт

На базе TCP работают очень многие (подавляющее большинство прикладных протоколов).

Интерактивные:

- Telnet, Rlogin, rsh, ssh, irc.

Протоколы передачи файлов:

- ftp, http, SMTP, POP3 и др.

## 8.3 Установление TCP-соединения

В отличие от UDP, когда сторона посылает дейтаграмму без какого-либо предварительного «рукопожатия», TCP прежде чем начать передавать данные устанавливает соединение с другой стороной.

Это можно показать на временной диаграмме.

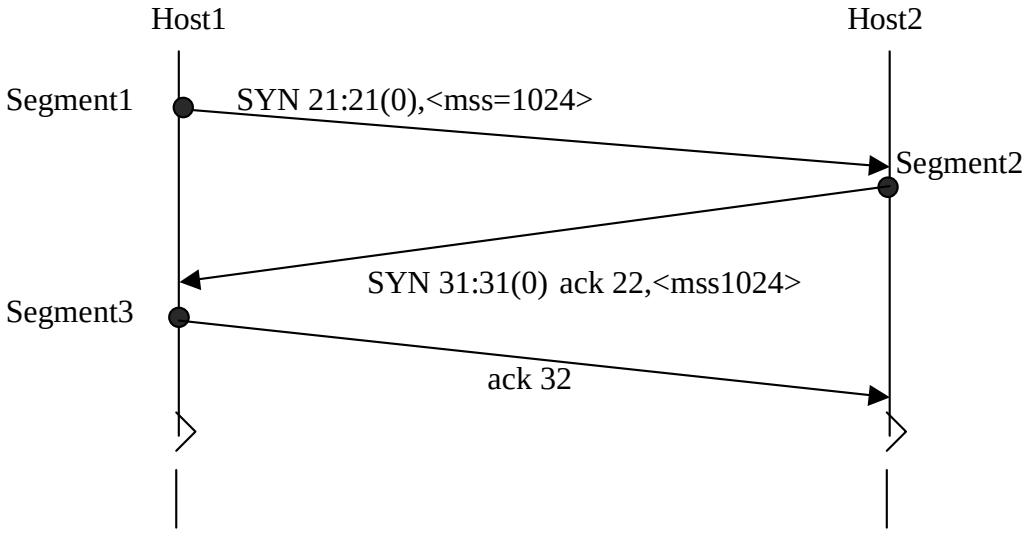



Рисунок 8.4 — Установление TCP-соединения

1. Запрашивающая сторона, обычно называемая клиентом, посылает TCP-сегмент с установленным битом SYN, содержащий номер порта, на который желательно установить соединение, ISN (Initial Sequence Number — начальный порядковый номер) и максимальный размер TCP-сегмента (mss).

2. Сервер отвечает его собственным сегментом SYN, содержащим ISN сервера. Он также подтверждает получение SYN-сегмента клиента, путем выставления флага ACK и Ack.Number=$ISN_{client}$+1. Заметим, что SYN-флаг «потребил» 1 байт.

3. Клиент также подтверждает получение SYN-сегмента от сервера: отправляет ACK и Ack.Number = $ISN_{client}$+1. Эти три сегмента полностью устанавливают соединение. Часто это называют «three-way handshake» — тройное рукопожатие.

Сторона, посылающая первый SYN, выполняет «активное открытие» соединения. Другая сторона (сервер) выполняет пассивное открытие. ISN (начальный порядковый номер) должен постоянно изменяться (желательно случайным образом). По RFC793 — это значение должно увеличиваться на 1 каждые 4 мкс. В BSD-реализациях самый первый ISN=1 и увеличивается на 64000 каждые 500 мс (не 4 мкс, а 8 мкс). И каждое новое соединение увеличивает ISN на 64000. Полный цикл составляет около 9,5 часов.

BCD — термин, принятый для описания любой версии операционной системы семейства UNIX, в основе которой лежит операционная система BCD, созданная в Калифорнийском университете в Беркли.

## 8.4 Завершение TCP-сеанса

В то время, как для установления TCP-соединения достаточно 3-х сегментов, для его завершения нужно 4 сегмента: по одному с битом FIN и по одному с битом ACK в каждом направлении. Первый бит ACK и второй бит FIN могут содержаться в одном TCP-сегменте, что уменьшит количество сегментов до трех (рис. 8.5). Поскольку TCP-соединение является полнодуплексным, каждая из сторон может закрыть канал независимо от другой, то есть послать сегмент с установленным в единицу битом FIN. Сторона, первая отправившая FIN-сегмент, выполняет активное закрытие. Другая сторона — пассивное закрытие.

Любая сторона может выполнить закрытие первой. Как правило, это делает клиент, но не обязательно.

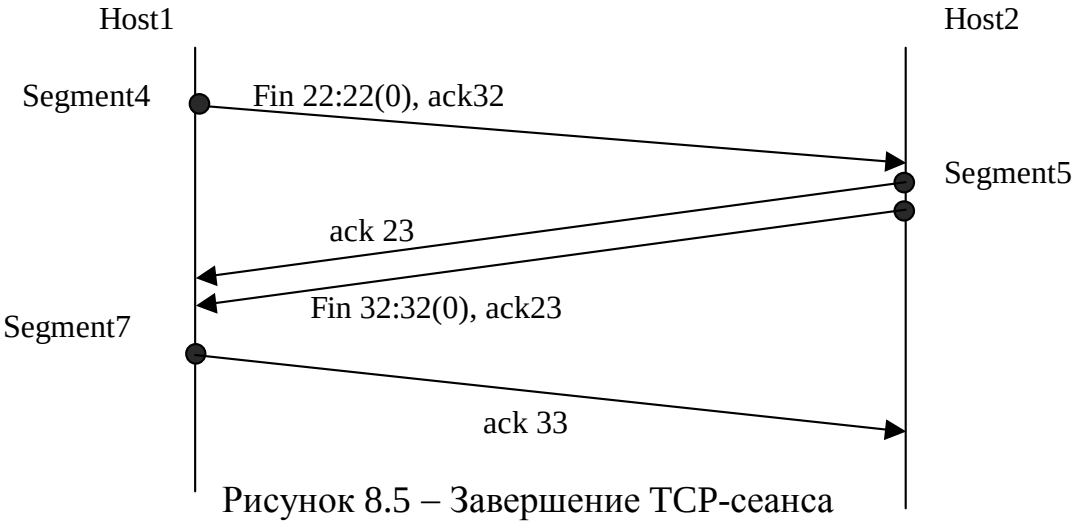



Рисунок 8.5 – Завершение TCP-сеанса

Обычно это выглядит так.

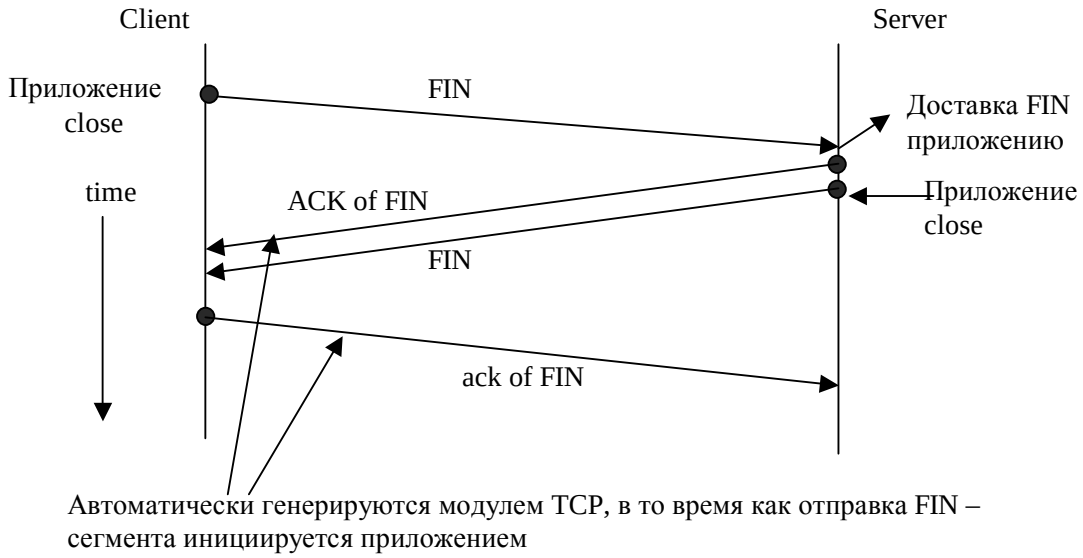



Рисунок 8.6

Особые ситуации:

1. Тайм — аут при установлении соединения: до другой стороны нет связи. Тайм-аут приблизительно 76 сек. 5÷6с+24+24+24 (operation timed out).

2. Порт не прослушивается — TCP-модуль принимающей стороны в этом случае отправляет RST-сегмент и безусловно закрывает соединение.

3. Полузакрытое состояние TCP-соединения — одна сторона отправила FIN и получила ACK, но не получила FIN.

## 8.5 Состояние TCP-сеанса

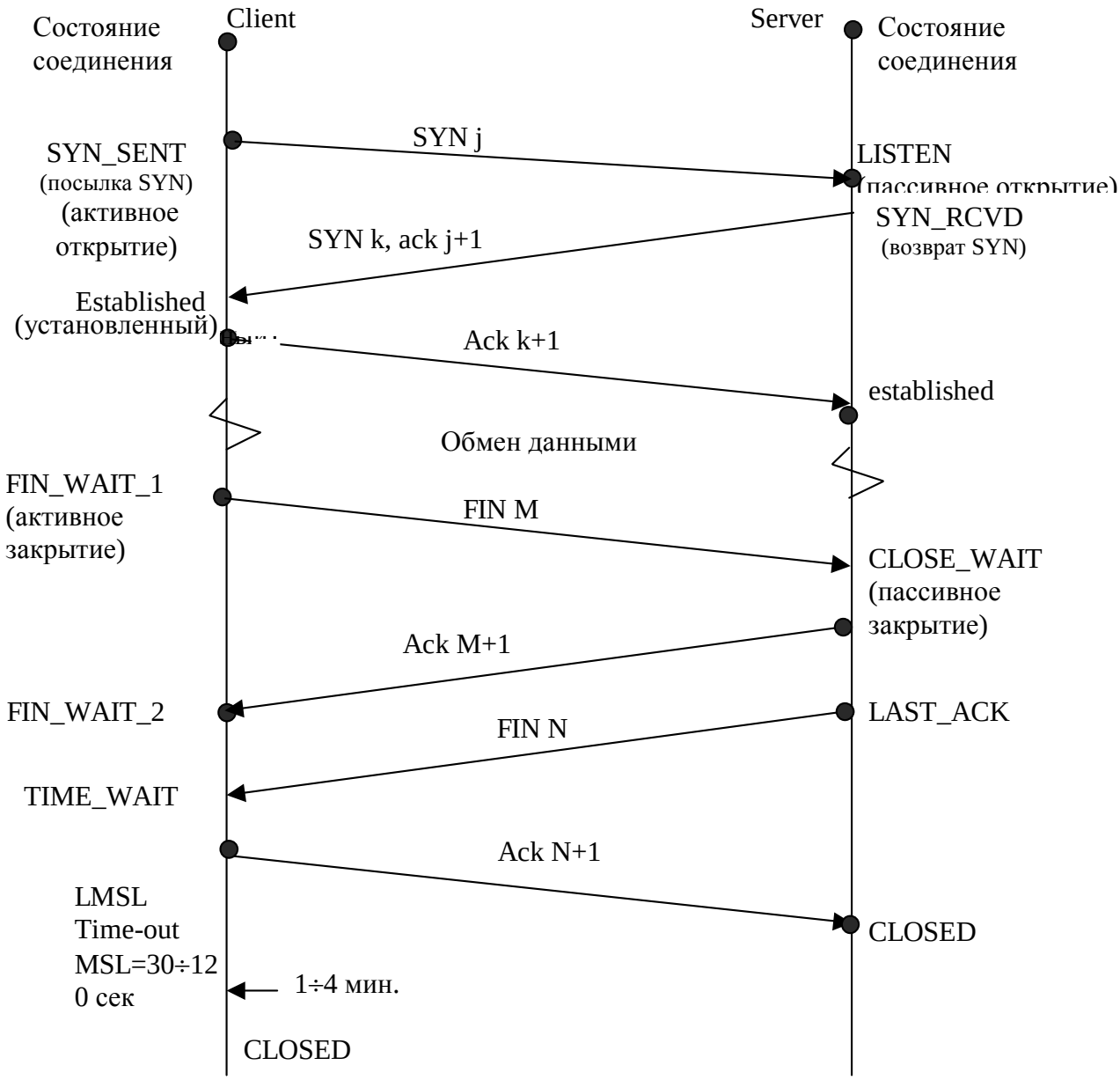



Рисунок 8.7 — Состояния TCP-сеанса

В состоянии FIN_WAIT_2 мы посылаем наш FIN, а удаленная сторона подтверждает его, посылая нам FIN. Только когда процесс на удаленном конце осуществит это закрытие, наша сторона перейдет из режима FIN_WAIT_2 в режим TIME_WAIT.

В состоянии FIN_WATT соединение может находиться 10 мин + 75 сек. (после чего TCP переводит соединение в режим

ЗАКРЫТО (CLOSED), так как, потенциально, другая сторона может вообще никогда не отправить FIN.

В течении TIME_WAIT невозможно использовать номер порта, задействованного в соединении. Если сервер не принял ACK N+1, он выполнит ретрансляцию FIN N, клиент должен быть готов его принять и подтвердить.

# 9 ПОТОК ИНТЕРАКТИВНЫХ ДАННЫХ

По статистике соотношение интерактивных (Telnet, Rlogin) и громоздких данных (FTP, SMTP) по пакетам приблизительно 50 на 50. По байтам же — приблизительно 10 на 90. В среднем пакет интерактивных данных содержит менее 10 байт данных, в то время как сегменты громоздких данных около 512 байт. TCP работает с обоими типами данных, но использует в том и другом случае несколько разные алгоритмы.

При интерактивном вводе каждое нажатие клавиши генерирует пакет данных. Другими словами, при нажатии клавиши от клиента серверу посылается 1 байт+40 байт в заголовках=41 байт (накладные расходы 4000%) за один промежуток времени. Более того, сервер отражает эхом символы, которые введены клиентом. При этом генерируется 4 сегмента: (1) интерактивный ввод символа от клиента, (2) подтверждение получения символа от сервера, (3) эхо введенного символа от сервера и (4) подтверждение на эхо от клиента. На рисунке 9.1 показан обмен данными.

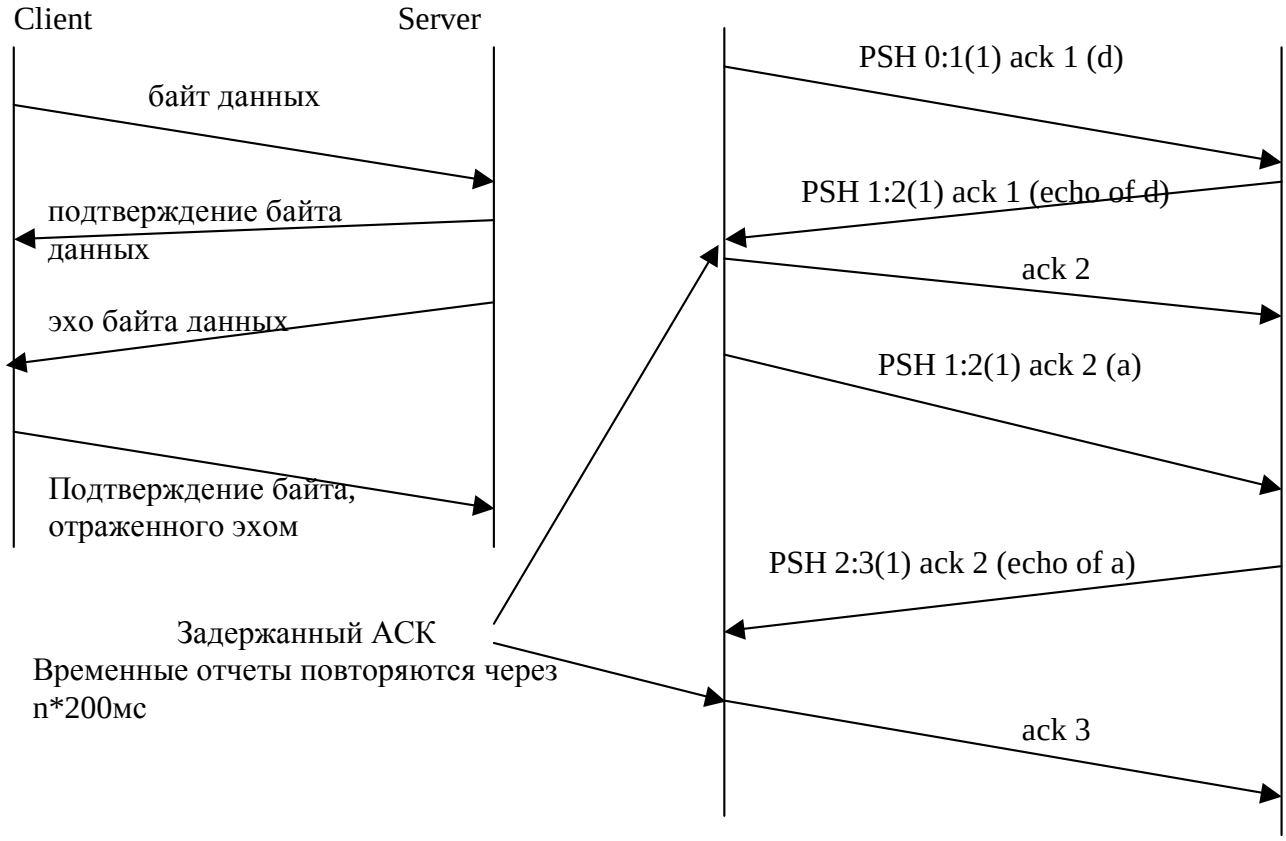



Рисунок 9.1 — TCP-обмен данными

Для того, чтобы уменьшить трафик TCP не отправляет ACK сразу по приему данных. Вместо этого он осуществляет задержку подтверждений в надежде на то, что в этом же направлении будут отправлены данные, таким образом ACK может быть отправлено вместе с данными. Большинство реализаций используют задержку, равную 200 миллисекунд, — таким образом, TCP задерживает ACK на время до 200 миллисекунд, чтобы посмотреть, не направляются ли данные в том же направлении, что и ACK.

## 9.1 Алгоритм Нейгла (Nagle Algorithm) (RFC 896)

При интерактивном вводе от клиента к серверу (например, через Rlogin) передается 1 байт за один раз. При этом генерируются пакеты размером 41 байт: 20 байт — IP заголовок, 20 байт — TCP заголовок и 1 байт данных. Маленькие пакеты (называемые тинограммами, от английского tiny — крошечный, маленький) обычно не проблема для локальных сетей, так как большинство локальных сетей не перегружаются, однако они могут привести к перегрузке глобальной сети. Простое решение было предложено в RFC 896, которое сейчас называется алгоритмом Нейгла (Nagle algorithm).

Алгоритм Нейгла заключается в том, что TCP-сессия должна содержать не более одного неподтвержденного сегмента малого размера. Малые сегменты не должны передаваться в сеть до тех пор, пока не прейдет подтверждение на ранее отправленный сегмент. Вместо этого малые количества данных должны собираться в буфере отправки в более крупный сегмент и отправляться с АСК после прихода сегмента АСК с той стороны.

Алгоритм Нейгла самосинхронизирующийся — чем быстрее приходят АСК, тем быстрее передаются данные. В медленных глобальных сетях, где необходимо уменьшить количество маленьких пакетов, отправляется меньше сегментов.

Малый сегмент — это значит, его размер меньше <MSS> (максимальный размер сегмента).

Есть, однако, ситуации, в которых нужно отключать алгоритм Нейгла. Типичный пример — X Window System или использование функциональных клавиш, когда генерируется несколько байт вместо одного. В этих случаях, чтобы не возникало нежелательных задержек, Алгоритм Нейгла выключается. Это делается

обычно самой прикладной программой путем передачи соответствующей опции модулю TCP (TCP_NODELAY) в BSD-системах. Можно отключить delayed ack и на общесистемном уровне.

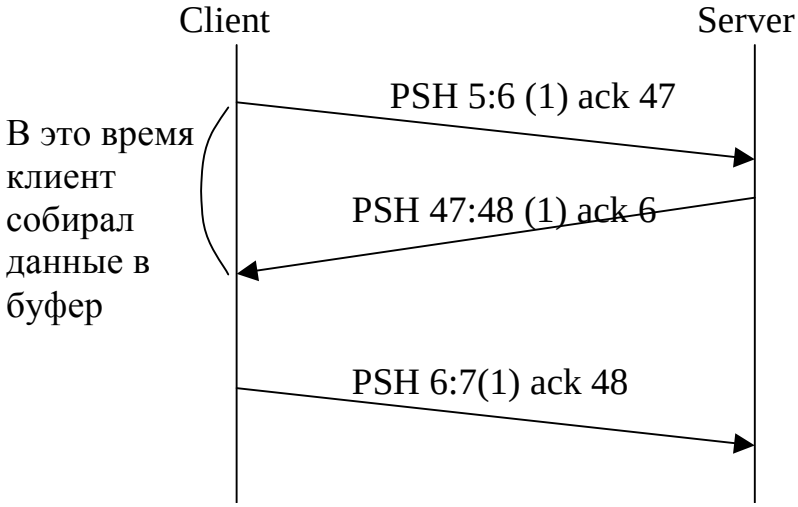



Рисунок 9.2

## 9.2 Передача большого объема данных

О передачи в рамках TCP-соединения большого количества данных можно говорить, когда происходит неинтерактивный обмен данными, обычно при передаче файлов по протоколам FTP, HTTP, SMTP и др. В такой ситуации один TCP-сегмент несет, как правило, число байт, близкое к MSS, объявленному приемной стороной.

Рассмотрим одностороннюю передачу 8192 байт от хоста svr4 к хосту bsdi.

Клиент осуществляет в сеть восемь записей размером 1024 байта каждая. На рисунке 9.3 показана временная диаграмма этого обмена. Мы оставили первые 3 сегмента вывода, чтобы показать значение MSS (максимальный размер сегмента) для каждой стороны.

Во-первых, отправитель передает три сегмента данных (4-6). Следующий сегмент (7) подтверждает только первые два сегмента данных. Мы знаем об этом, потому что номер последовательности подтверждения равен 2049, а не 3073.

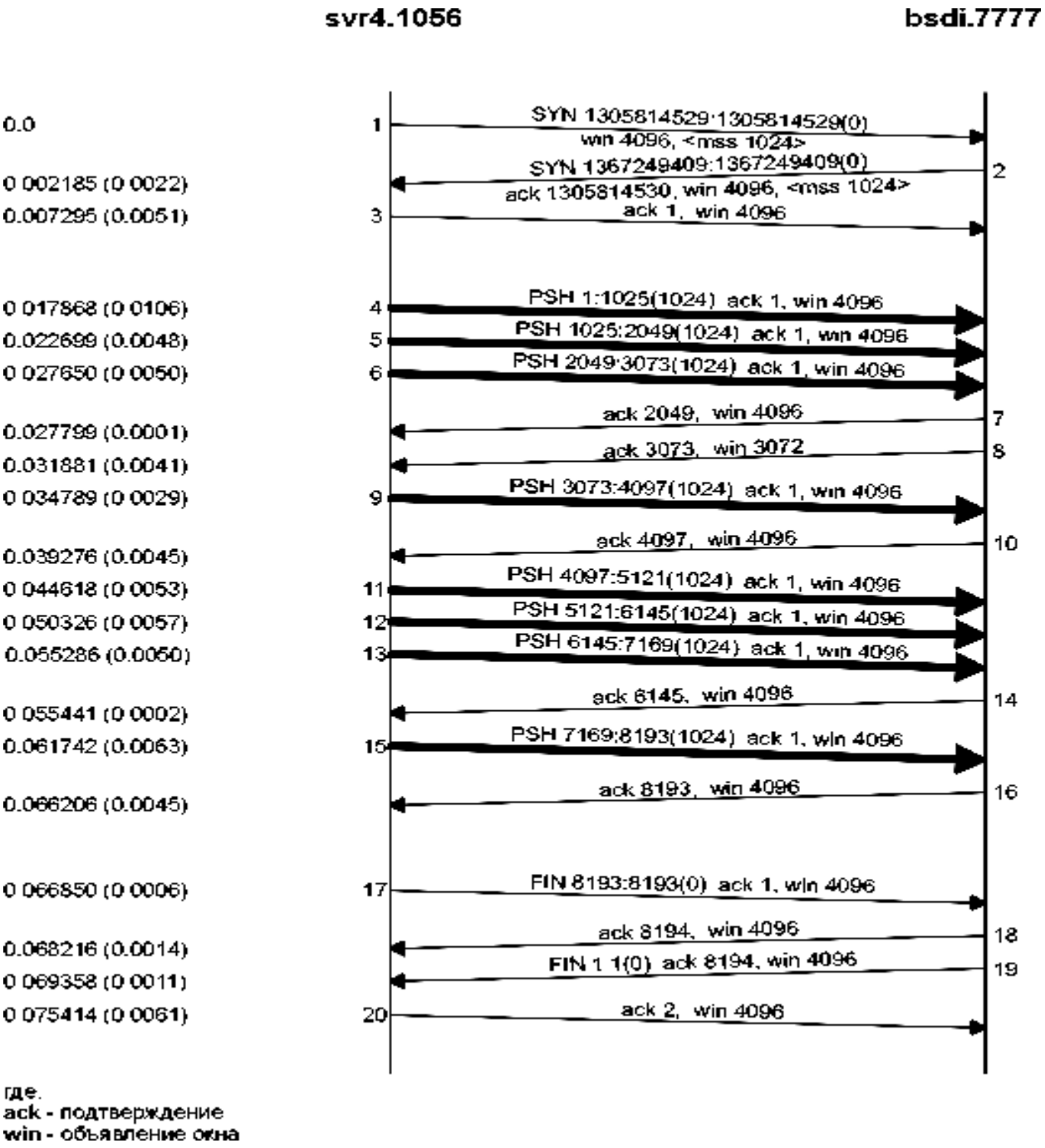



Рисунок 9.3 — Передача 8192 байт от svr4 к bsdi

Сегмент 7 содержит ACK с номером 2049, а не 3073 по следующей причине. Когда пакет прибывает, он первоначально обрабатывается драйвером устройства, а затем помещается во входную очередь IP. Три сегмента 4, 5 и 6 прибывают один за другим и помещаются во входную очередь IP в том порядке, как они были приняты. IP затем передаст их в TCP в том же самом

порядке. Когда TCP обрабатывает сегмент 4, в соединении генерируется задержанный ACK. TCP обрабатывает следующий сегмент (5), и теперь TCP имеет два сегмента, на которые необходимо сгенерировать подтверждение (ACK), поэтому генерируется подтверждение с номером 2049 (сегмент 7), а флаг задержанного ACK для этого соединения снимается. TCP обрабатывает следующий входной сегмент (6), а в соединение снова генерируется задержанный ACK. Перед тем как прибывает сегмент 9, выключается таймер задержанного ACK и генерируется подтверждение с номером 3073 (сегмент 8). В сегменте 8 окно объявляется размером 3072 байта, так как 1024 байта данных в приемном буфере TCP до сих пор не прочитаны приложением.

В случае сегментов 11-16 подтверждение осуществляется на каждый сегмент. Сегменты 11, 12 и 13 прибывают и помещаются во входную очередь IP. Когда сегмент 11 обрабатывается TCP, соединение помечается как использующее задержанное ACK. Когда обрабатывается сегмент 12, генерируется ACK (сегмент 14) на сегменты 11 и 12, а флаг задержанного ACK для данного соединения снимается. При обработке сегмента 13 соединение вновь помечается как использующее задержанное ACK, однако перед тем как задержанный ACK снимается по таймеру, обрабатывается сегмент 15, при этом ACK (сегмент 16) отправляется немедленно.

Очень важно обратить внимание на то, что ACK в сегментах 7, 14 и 16 подтверждает два принятых сегмента. В случае использования протокола TCP с изменяющимся окном, принимающая сторона не должна подтверждать каждый принятый пакет. В случае TCP, подтверждения накапливаются — они подтверждают, что получатель корректно принял все байты до номера последовательности подтверждения минус один. В этом примере три из ACK подтвердили 2048 байт данных, а два подтвердили 1024 байта данных. (За исключением ACK, появлявшихся при установлении и разрыве соединения.)

## 9.3 Протокол «скользящего окна»

При передаче большого объема данных важную роль играет механизм управления потоком данных протокола TCP, так называемый протокол «скользящего окна». Соответственно, важное значение приобретает поле window в заголовке TCP — сегмента, которое, по сути, отражает состояние буфера приема. Это позволяет передающей стороне отправить много пакетов перед тем, как остановиться и ожидать подтверждения. Это приводит к более быстрой передаче данных, поскольку передающая сторона, каждый раз послав пакет, не останавливает передачу для ожидания подтверждения.

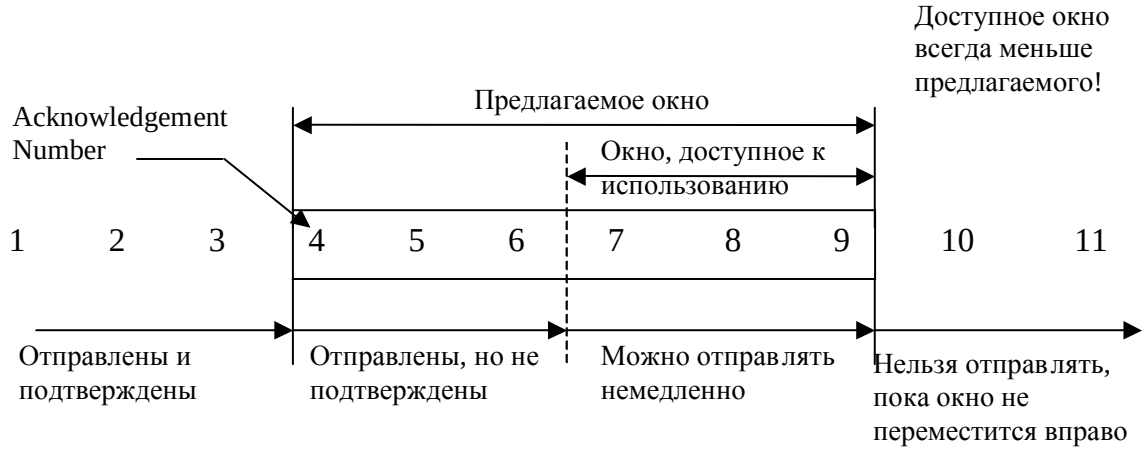



Рисунок 9.4 — Иллюстрация работы протокола «скользящего окна»

В этом примере принимающая сторона предлагает окно в 6 байт, покрывающее байты с 4 по 9 включительно, что означает, что приемник подтвердил все байты до 3 включительно. Передатчик вычисляет доступное окно (количество байт, которые могут быть переданы немедленно). С течением времени скользящее окно движется вправо, когда приемник подтверждает данные. Относительное движение двух сторон окна увеличивает или уменьшает размер окна. Для описания движения левого и правого края окна используются следующие термины:

1. Окно *закрывается* по мере смещения левого края вправо. Это происходит, когда данные переданы и подтверждены.

2. Окно *открывается*, когда правый край окна движется вправо, позволяя отправлять больше данных. Это происходит,

когда принимающий процесс на другой стороне читает уже подтвержденные данные, освобождая место в буфере приема TCP.

3. Окно *сжимается*, когда правый край движется влево. RFC предостерегает от этого, но TCP модуль должен уметь обработать такую ситуацию.

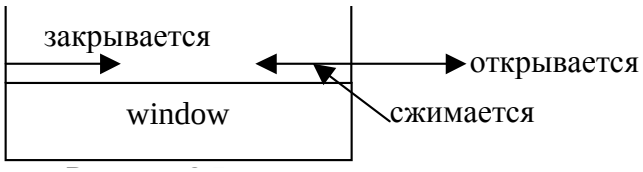



Рисунок 9.5

Если принимается АСК, подразумевающий движение левого края влево, это дублирующий АСК, он уничтожается. Если левый край достигает правого, это «нулевое» окно. Передатчик останавливается до window update.

На рисунке 9.6 показана динамика работы протокола TCP с изменением окна для передачи данных, показанной на рисунке 9.3.

1. Отправитель не передает данных в пределах полного окна.

2. Один сегмент от принимающего подтверждает данные и сдвигает окно вправо, поскольку размер окна относителен Acknowledgement Number.

3. Размер окна может уменьшаться, как видно на примере сегментов 7 и 8, но при этом правый край окна не должен смещаться вправо.

4. Принимающий не ждет заполнения окна перед отправкой АСК-сегмента. Многие реализации передают АСК для каждых двух принятых сегментов.

### *9.3.1 Размер окна*

Размер окна, предлагаемый получателем, обычно определяется получающим процессом. Это может влиять на производительность. TCP Размер приемного буфера TCP — это максимально возможный размер предлагаемого окна.

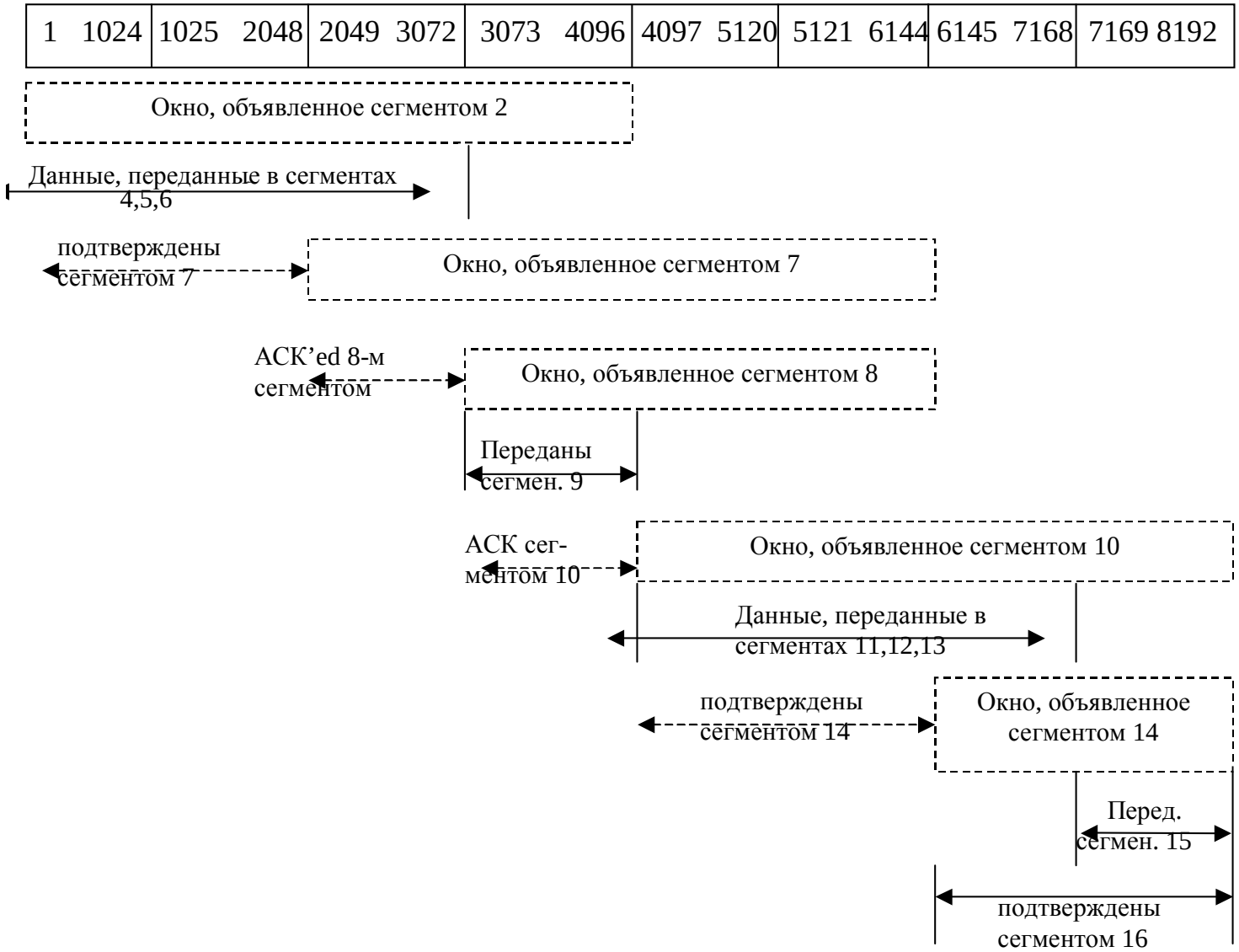



Рисунок 9.6 — Протокол изменения размера окна для рисунка 9.3

В 4.2BSD приемные и отправляющие буферы по умолчанию устанавливаются в 2048 байт каждый. В 4.3BSD оба были увеличены до 4096 байт. Как мы можем видеть из всех примеров, приведенных в тексте, SunOS 4.1.3, BSD/386 и SVR4 все еще используют по умолчанию размер буфера 4096 байт. Другие системы, такие как Solaris 2.2, 4.4BSD и AIX 3.2, используют по умолчанию большие размеры буферов, 8192 или 16384 байта. Было показано, что увеличение буфера с 4096 до 16384 для хостов, подключенных к Ethernet, приводит приблизительно к 40% увеличению производительности TCP-соединений. Минимальный размер буфера вычисляется исходя из емкости канала.

### *9.3.2 Флаг PUSH*

Наличие этого флага в сегменте указывает принимающей стороне протолкнуть данные сегмента *вместе* с накопленными до текущего момента в буфере данными. Оригинальная спецификация TCP говорит о том, что API должен предоставить возможность передающему процессу указывать своему TCP о необходимости выставления этого флага. При этом клиент говорит своему TCP, что он не желает, чтобы данные задерживались в буфере, ожидая дополнительных данных перед отправкой сегмента. Аналогично, когда TCP приемной стороны принимает сегмент с флагом PUSH, это является для TCP указанием на то, что данные нужно немедленно отдать приложению, не ожидая прихода дополнительных данных. В наше время большинство реализаций TCP сами определяют, когда нужно выставлять флаг PUSH. В BSD реализациях PUSH выставляется, когда передаваемые данные освобождают буфер передачи, а также, в основном, игнорируют принятый PUSH, поскольку обычно никогда не задерживают принятые данные в буфере приема.

### *9.3.3 Алгоритм медленного старта*

Старые реализации TCP начинали отправку данных в сеть в пределах предлагаемого окна, не дожидаясь прихода подтверждений. Когда речь идет о передаче данных между двумя хостами на

одной LAN, это не вызывает проблем, но если между ними имеются промежуточные маршрутизаторы или медленные каналы, могут возникнуть проблемы. Некоторые промежуточные маршрутизаторы должны ставить пакеты в очередь, и это может вызвать переполнение их буферов и, как следствие, — потерю пакетов и т.д.

Поэтому от TCP требуется, чтобы он поддерживал алгоритм, который называется медленный старт. Он заключается в том, что TCP следит за тем, чтобы скорость ухода сегментов в сеть соответствовала бы скорости прихода подтверждений с той стороны.

Для этого отправляющему TCP добавляется еще одно окно: окно переполнения, которое называется CWND. Когда устанавливается новое соединение, CWND равно размеру одного сегмента (например, MSS). Каждый раз, когда приходит ACK, CWND увеличивается на размер этого сегмента. Отправитель может передать объем данных величиной до минимального размера окна переполнения и объявленного окна. Окно переполнения — это контроль потока, осуществляемый передающим, в то время, как предлагаемое окно — контроль потока со стороны принимающего.

SZ — размер сегмента, например, SZ=MSS.

$CWND_0=SZ$;

$CWND_1=CWND_0+SZ=2CWND_0$;

$CWND_1=CWND_0+(CWND_0/SZ)*SZ$;

$CWND_n=2*CWND_{n-1}=2^n*SZ$, где n — номер пришедшего ACK, виден экспоненциальный рост.

Начиная с некоторого момента времени, пропускная способность канала будет полностью использована, и пакеты начнут теряться. Это значит, что CWND передающего слишком велико. Далее включаются механизмы преодоления затора, на основе тайм-аутов и ретрансляции.

Пример медленного старта (рис. 9.7).

На рисунке 9.7 показаны данные, которые отправляются от хоста sun на хост vangogh. Эти данные проходят по медленному каналу, который в данном случае будет узким местом передачи. (Из этой временной диаграммы удалено все, что связанно с установлением соединения.)

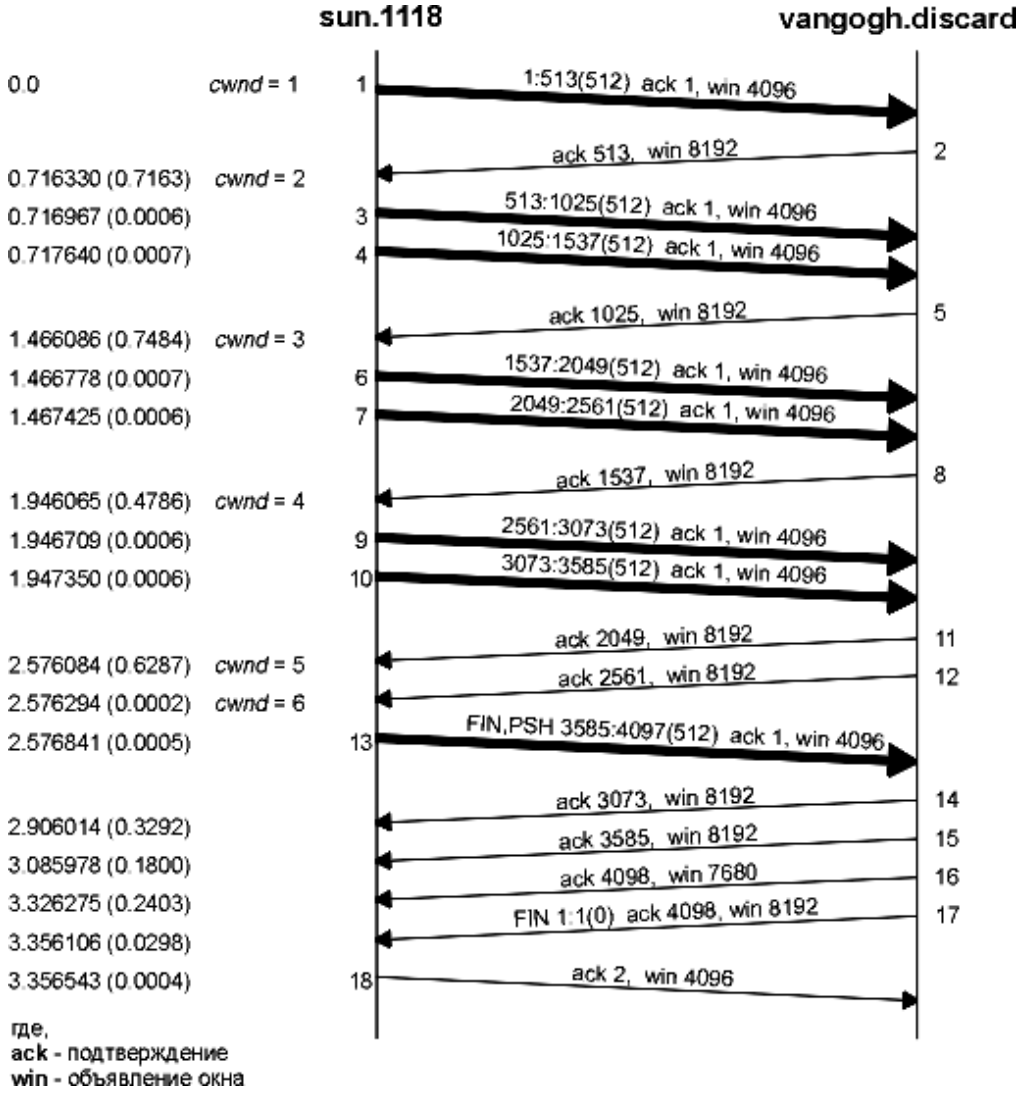



Рисунок 9.7 — Пример медленного старта

Мы видим, что отправитель отправляет один сегмент с 512 байтами данных, а затем ожидает ACK. ACK получено через 716 миллисекунд, что определяет время возврата. Затем окно переполнения увеличивается до двух сегментов, и эти два сегмента отправляются. Когда ACK получен в сегменте 5, окно переполнения увеличивается до трех сегментов. Несмотря на то, что может быть послано три сегмента, отправляется только два, перед тем как не будет получен еще один ACK.

На рисунке 9.8 показана работа подобного соединения, отправитель находится слева, получатель — справа. На рисунке показано 16 моментов времени. Для простоты время показано дискретно. В верхней половине каждого рисунка мы показываем сегменты, переносящие данные слева направо, и пронумерован-

ные как 1, 2, 3 и так далее. ACK двигаются в другом направлении и показаны в нижней половине каждого рисунка. Мы рисуем ACK меньше и указываем номера сегментов, которые подтверждаются.

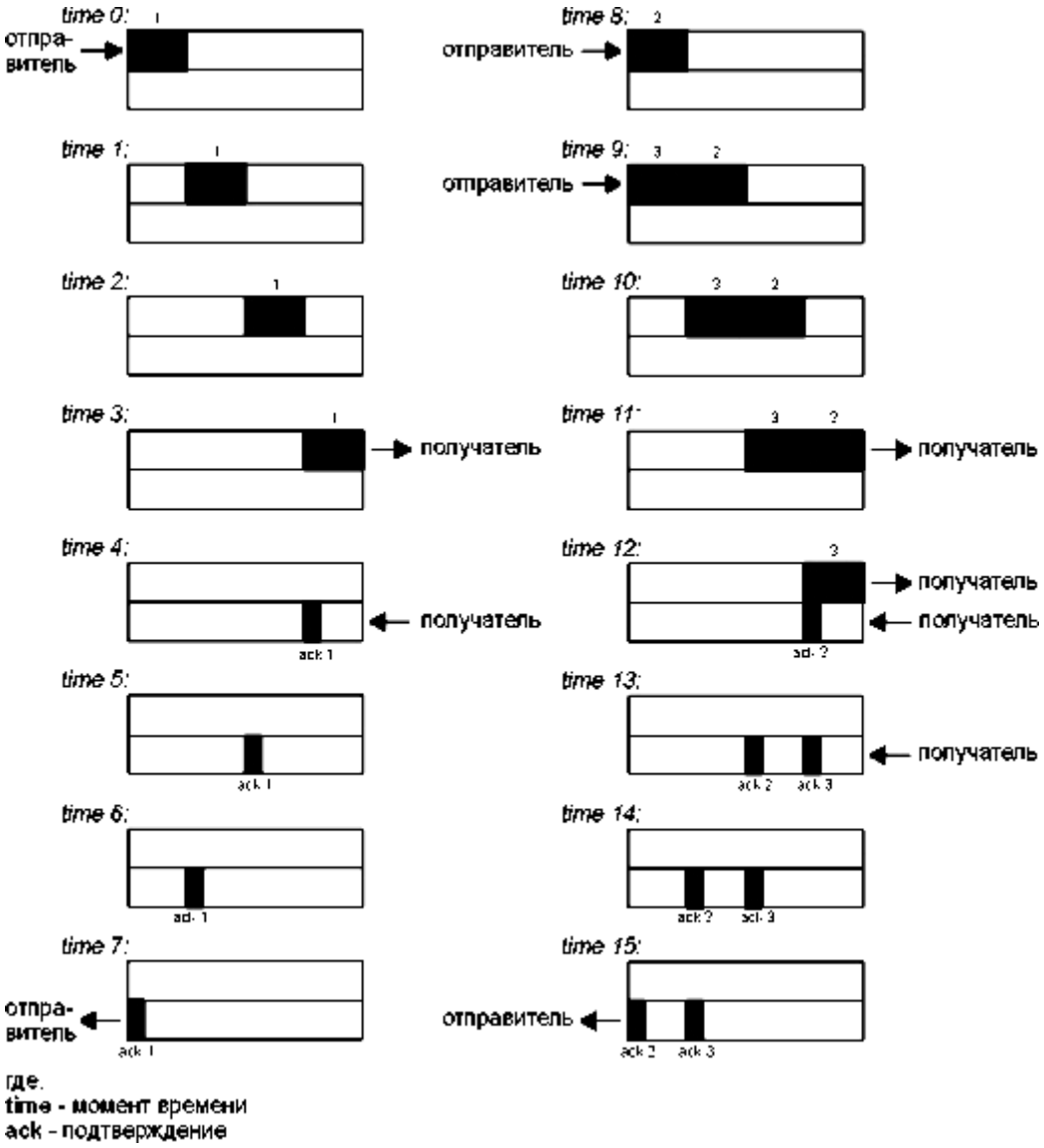



Рисунок 9.8 — Моменты времени 0-15, иллюстрирующие пропускную способность при передаче неинтерактивных данных

В момент времени 0 отправитель посылает один сегмент. Так как отправитель работает с медленным стартом (окно переполнения установлено в один сегмент), он должен ждать подтверждения на этот сегмент, перед тем как продолжить работу.

В моменты времени 1, 2 и 3 сегмент проходит по одному промежутку времени вправо. В момент времени 4 получатель читает сегмент и генерирует подтверждение. В моменты времени 5, 6 и 7 подтверждение движется по одному промежутку времени влево, обратно к отправителю. Таким образом, время возврата (RTT) составляет 8 промежутков времени.

Сегмент подтверждения (ACK) специально нарисован меньше, чем сегмент данных, так как обычно он состоит из IP заголовка и TCP заголовка. Здесь показан поток данных без учета направления в один и тот же момент времени. Также сделано предположение, что ACK двигается с той же самой скоростью, что и сегмент данных, что в действительности не всегда верно.

В общем, время отправки пакета зависит от двух факторов: задержки прохождения (которая вызвана конечной скоростью света, временами ожидания и аппаратурой передачи) и задержки передачи, которая зависит от скорости среды передачи (количество бит, которое может быть передано в среде передачи за секунду). Для данного пути между двумя узлами задержка прохождения фиксированна, тогда как задержка передачи зависит от размера пакета. При небольших скоростях определяющими являются задержки передачи, однако для скоростей равных гигабитам определяющей является задержка прохождения.

Когда получатель принимает ACK, он может передать два сегмента (которые мы пронумеровали как 2 и 3) в моменты времени 8 и 9. Окно переполнения сейчас составляет два сегмента. Эти два сегмента двигаются вправо по направлению к приемнику, где генерируются подтверждения в моменты времени 12 и 13. Промежутки времени между подтверждениями (ACK) идентичны промежуткам между сегментами данных. Это поведение TCP называется самонастройкой по времени. Так как получатель может генерировать ACK только тогда, когда данные получены, по промежуткам между подтверждениями можно определить скорость прибытия данных к приемнику.

# 10 ПРИКЛАДНЫЕ СЕРВИСЫ TCP/IP

## 10.1 Протокол FTP (File Transfer Protocol, RFC 959)

FTP — один из первых (первые спецификации 1971 г.) сервисов Internet. Он является стандартом Internet для передачи файлов. Необходимо различать передачу файлов, именно то, что предоставляет FTP, и доступ к файлам, что предоставляется такими приложениями как NFS (Network File System — сетевая файловая система). Передача файлов заключается в копировании целого файла из одной системы в другую.

FTP отличается от других приложений тем, что он использует два TCP соединения для передачи файла.

1. Управляющее соединение устанавливается как обычное соединение клиент-сервер. Сервер осуществляет пассивное открытие на заранее известный порт FTP (21) и ожидает запроса на соединение от клиента. Клиент осуществляет активное открытие на TCP порт 21, чтобы установить управляющее соединение. Управляющее соединение существует все время, пока клиент общается с сервером. Это соединение используется для передачи команд от клиента к серверу и для передачи откликов от сервера. Тип IP сервиса для управляющего соединения устанавливается для получения «минимальной задержки», так как команды обычно вводятся пользователем.

2. Соединение данных открывается каждый раз, когда осуществляется передача файла между клиентом и сервером. Тип сервиса IP для соединения данных должен быть «максимальная пропускная способность», так как это соединение используется для передачи файлов.

На рисунке 10.1 показано общение клиента и сервера по двум соединениям.

Из рисунка видно, что интерактивный пользователь обычно не видит команды и отклики, которые передаются по управляющему соединению. Эти детали оставлены двум интерпретаторам протокола. Квадратик, помеченный как «пользовательский интерфейс», это именно то, что видит интерактивный пользователь. Отклики, возвращаемые сервером по управляющему соединению, конвертируются в формат, удобный для пользователя.

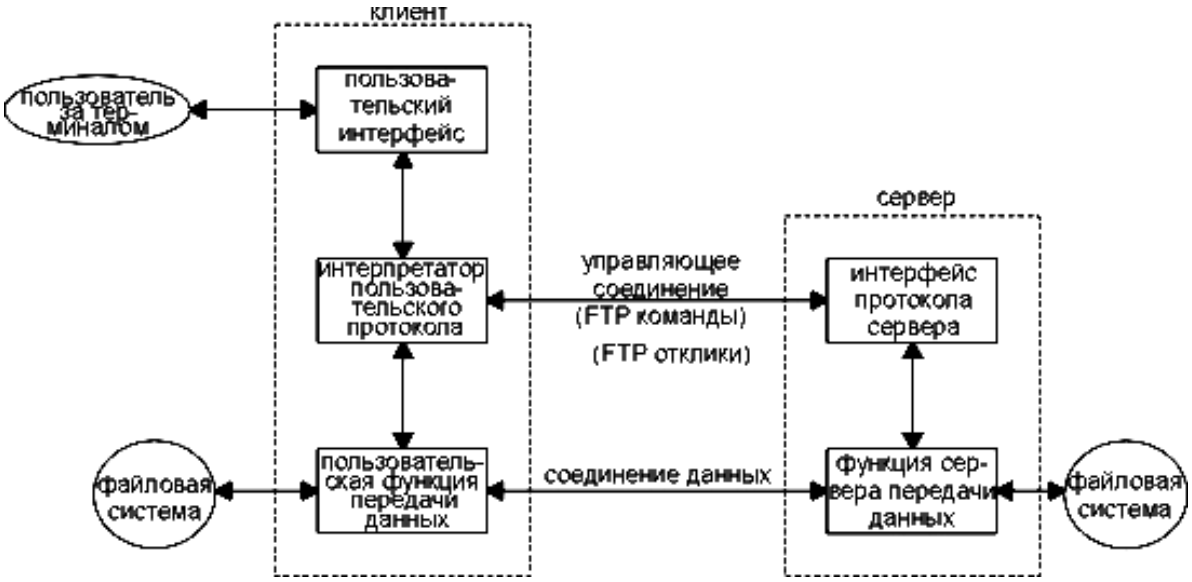



Рисунок 10.1 — Процессы, участвующие в передаче файлов

## 10.2 Представление данных

Протокол FTP предоставляет различные способы управления передачей и хранения файлов. Необходимо сделать выбор по четырем пунктам.

1. Тип файла.

(a) ASCII файлы.

(По умолчанию). Текстовый файл передается по соединению данных как NVT ASCII. При этом требуется, чтобы отправитель конвертировал локальный текстовый файл в NVT ASCII, а получатель конвертировал NVT ASCII в текстовый файл. Конец каждой строки передается в виде NVT ASCII символа возврата каретки, после чего следует перевод строки. Это означает, что получатель должен просматривать каждый байт в поисках пары символов CR, LF.

(b) Двоичные или бинарные файлы. (Image.)

Данные передаются как непрерывный поток битов.

2. Управление форматом. Применяется только для ASCII.

(a) Nonprint (По умолчанию).

Файл не содержит информацию вертикального формата.

(b) Telnet format control.

Файл содержит управляющие символы вертикального формата Telnet, которые интерпретируются принтером.

3. Структура.

(a) Структура файла.

(По умолчанию). Файл воспринимается в виде непрерывного потока байтов. Файл не имеет внутренней структуры.

(b) Структура записи.

Эта структура используется только в случае текстовых файлов (ASCII или EBCDIC).

(c) Структура страницы.

Каждая страница передается с номером страницы, что позволяет получателю хранить страницы в случайном порядке. Предоставляется операционной системой TOPS-20.

4. Режим передачи. Указывает на то, как файл передается по соединению данных.

(a) Режим потока.

(По умолчанию). Файл передается как поток байтов. Для файловой структуры конец файла указывает на то, что отправитель закрывает соединение данных. Для структуры записи специальная 2-байтовая последовательность обозначает конец записи и конец файла.

(b) Режим блоков.

Файл передается как последовательность блоков, перед каждым из них стоит один или несколько байт заголовков.

(c) Сжатый режим.

Простое кодирование неоднократно встречающихся повторяющихся байт. В текстовых файлах обычно сжимаются пустые строки или строки из пробелов, а в бинарных строки из нулевых байт. (Этот режим поддерживается редко. Существуют более оптимальные способы сжатия файлов для FTP.)

Если посчитать количество комбинаций из приведенных вариантов, то получится 72 способа передачи и хранения файла. Но многие из этих опций можно игнорировать, потому что они не поддерживаются в большинстве реализаций.

Самые распространенные Unix реализации FTP клиента и сервера предоставляют следующий выбор:

- Тип: ASCII или двоичный.

- Управление форматом: только nonprint.
- Структура: только файловая структура.
- Режим передачи: только потоковый режим.

Это ограничивает нас одним из двух режимов: ASCII или двоичный.

## 10.3 Команды FTP

Команды и отклики передаются по управляющему соединению между клиентом и сервером.

Команды состоят из 3 или 4 байт, а именно из заглавных ASCII символов. Клиент может отправить серверу более чем 30 различных FTP команд. В таблице 10.1 показаны некоторые наиболее широко используемые команды.

Таблица 10.1 — Распространенные FTP команды

| Команда | Описание |
|---|---|
| ABOR | прервать предыдущую команду FTP и любую передачу данных |
| LIST список файлов | список файлов или директорий |
| PASS пароль | пароль на сервере |
| PORT n1,n2,n3,n4,n5,n6 | IP-адрес клиента (n1.n2.n3.n4) и порт (n5 x 256 + n6) |
| RETR имя файла | получить (get) файл |
| STOR имя файла | положить (put) файл |
| TYPE тип | указать тип файла: A для ASCII, I для двоичного |
| QUIT | закрыть бюджет на сервере |

Некоторые команды полностью совпадают с тем, что вводит интерактивный пользователь в качестве FTP команд. В этом случае они передаются по управляющему соединению, однако некоторые вводимые пользователем команды генерируют несколько FTP команд, которые, в свою очередь, передаются по управляющему соединению.

## 10.4 FTP отклики

Отклики состоят из 3-циферных значений в формате ASCII и необязательных сообщений, которые следуют за числами. Подобное представление откликов объясняется тем, что программному обеспечению необходимо посмотреть только цифровые значения, чтобы понять, что ответил процесс, а дополнительную строку может прочитать человек. Поэтому пользователю достаточно просто прочитать сообщение (причем нет необходимости запоминать все цифровые коды откликов).

Каждая из трех цифр в коде отклика имеет собственный смысл. В таблице 10.1 показаны значения первых и вторых цифр в коде отклика.

Таблица 10.2 — Значения первой и второй цифр в 3-циферном коде отклика

| Отклик | Описание |
|---|---|
| 1yz | Положительный предварительный отклик. Действие началось, однако необходимо дождаться еще одного отклика перед отправкой следующей команды. |
| 2yz | Положительный отклик о завершении. Может быть отправлена новая команда |
| x0z | Синтаксическая ошибка. |

Третья цифра дает дополнительное объяснение сообщению об ошибке. Ниже приведены некоторые типичные отклики с возможными объясняющими строками.

- 125 Соединение данных уже открыто; начало передачи.
- 200 Команда исполнена.
- 214 Сообщение о помощи (для пользователя).
- 331 Имя пользователя принято, требуется пароль.
- 425 Невозможно открыть соединение данных.
- 452 Ошибка записи файла.
- 500 Синтаксическая ошибка (неизвестная команда).
- 501 Синтаксическая ошибка (неверные аргументы).

Обычно каждая FTP команда генерирует отклик в одну строку. Например, команда QUIT генерирует следующий отклик:

221 Goodbye.

## 10.5 Управление соединением

Использовать соединение данных можно следующими способами.

1. Отправка файлов от клиента к серверу.
2. Отправка файлов от сервера к клиенту.

Управляющее соединение остается в активизированном состоянии все время, пока установлено соединение клиент-сервер, однако соединение данных может выключаться и включаться по необходимости. Как выбираются номера портов для соединения данных, и кто осуществляет активное открытие, а кто пассивное открытие?

Распространенный режим передачи (в случае Unix это единственный режим передачи) — это потоковый режим. В этом режиме конец файла обозначает закрытие соединения данных. Из этого следует, что для передачи каждого файла или списка директории требуется новое соединение данных.

На рисунке 10.2 показано, как сервер принимает номер порта с управляющего соединения и осуществляет активное открытие на этот порт хоста клиента. Сервер всегда использует порт 20 для соединения данных. Мы предполагаем, что динамически назначаемый порт клиента для управляющего соединения имеет номер 1173, а динамически назначаемый порт клиента для соединения данных имеет номер 1174. Команда, посылаемая клиентом, — PORT, а ее аргументы — это шесть десятичных цифр в формате ASCII, разделенные запятыми. Четыре первых числа — это IP-адрес клиента, на который сервер должен осуществить активное открытие (140.252.13.34 в данном примере), а следующие два — это 16-битный номер порта. Так как 16-битный номер порта формируется из двух цифр, его значение в этом примере будет 4 x 256 + 150 = 1174.

На рисунке 10.3 показано состояние соединений, когда сервер осуществляет активное открытие на конец клиента соединения данных. Конечная точка сервера это порт 20.

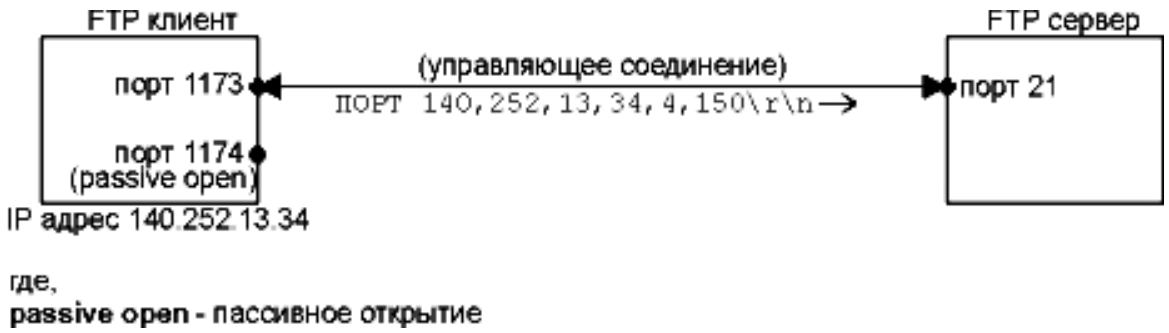



Рисунок 10.2 — Команда PORT, передаваемая по управляющему соединению FTP

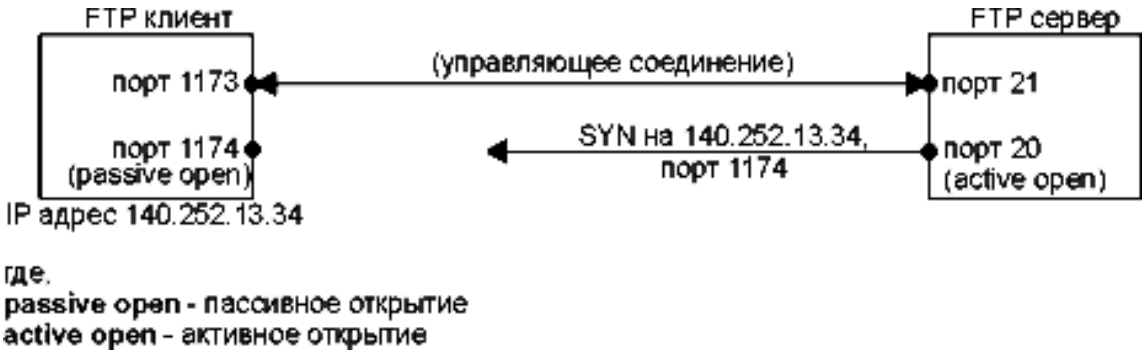



Рисунок 10.3 — FTP сервер осуществляет активное открытие соединения данных

Сервер всегда осуществляет активное открытие соединения данных. Обычно сервер также осуществляет активное закрытие соединения данных, за исключением тех случаев, когда клиент отправляет файл на сервер в потоковом режиме, который требует, чтобы клиент закрыл соединение (что делается с помощью уведомления сервера о конце файла).

Если клиент не выдает команду PORT, сервер осуществляет активное открытие на тот же самый номер порта, который использовался клиентом для управляющего соединения (1173 в данном примере). В этом случае все работает корректно, так как номера порта сервера для двух соединений различны: один 20, другой 21.

# 11 СЕРВИСЫ TCP/IP. ЭЛЕКТРОННАЯ ПОЧТА (E-MAIL)

Электронная почта является едва ли не самым популярным сервисом Internet. Примерно половина всех TCP-соединений занята передачей почтовых сообщений. Однако, с точки зрения переданных байт, по FTP-соединениям передается значительно больше данных. Среднее почтовое сообщение содержит примерно 1500 байт, но некоторые сообщения содержат мегабайты данных (как правило, используются для посылки файлов).

Internet-почта основана на протоколе SMTP (RFS 821), который предполагает прямую доставку сообщения путем установления непосредственного соединения с машиной-адресом. Если доставка не удалась, отправляющая система сразу узнает об этом. Она может проинформировать пользователя, поставить сообщение в очередь. При такой схеме обе системы должны постоянно поддерживать работу с почтой, что не для всех хостов реально. Чаще всего непосредственно между собой передают через Internet сообщения почтовые ретрансляторы (постоянно on-line), и почта концентрируется на почтовых серверах, хранящих почтовые ящики пользователей. К этим ящикам пользователи затем периодически обращаются со своих компьютеров по какому-либо протоколу (POP, IMAP).

Основная задача протокола SMTP (Simple Mail Transfer Protocol) заключается в том, чтобы обеспечивать передачу электронных сообщений (почту). Для работы через протокол SMTP клиент создает TCP соединение с сервером через порт 25. Затем клиент и SMTP сервер обмениваются информацией, пока соединение не будет закрыто или прервано. Основной процедурой в SMTP является передача почты (Mail Procedure). Далее идут процедуры форвардинга почты (Mail Forwarding), проверка имен почтового ящика и вывод списков почтовых групп. Самой первой процедурой является открытие канала передачи, а последней — его закрытие. Этот протокол относительно прост и содержит гораздо меньше команд, чем, например, FTP. Команды SMTP указывают серверу, какую операцию хочет произвести клиент. Основными командами SMTP являются:

HELO <отправитель> — идентифицирует сервер отправителя;

MAIL FROM: <адрес отправителя> — задает адрес отправителя;

RCPT TO: <адрес получателя> — задает адрес получателя;
DATA — указывает на начало сообщения;
RSET — прерывает передачу сообщения;
VRFY <адрес>  — проверяет адрес получателя;
EXPN <адрес> — выводит ассоциируемый с адресом список адресатов E-mail;
HELP [команда] — выводит справку по указанной команде;
QUIT — завершает SMTP-сеанс.

Обычный ответ SMTP сервера состоит из номера ответа, за которым через пробел следует дополнительный текст. Номер ответа служит индикатором состояния сервера.

## 11.1 Отправка почты

SMTP использует MTA (Mail Transfer Agent — программа рассылки электронной почты) для доставки почты через Internet из одной сети в другую.

Получателем сообщения может быть как хост назначения, так и промежуточный хост, который затем переправит сообщение дальше. Команды SMTP генерируются отправителем и направляются получателю SMTP, который отправляет кадры обработки полученных команд обратно (аналогично FTP).

Простейший сценарий работы SMTP-обмена может выглядеть, например, так (S — сервер, C — клиент):

Передаем серверу команду HELLO и наш IP-адрес:
C: HELLO 195.161.101.33
S: 250 smtp.mail.ru is ready

При отправке почты передаем некоторые нужные данные (отправитель, получатель и само письмо):
C: MAIL FROM:<drozd> 'указываем отправителя
S: 250 OK
C: RCPT TO:<drol@mail.ru> 'указываем получателя
S: 250 OK

Указываем серверу, что будем передавать содержание письма (заголовок и тело письма)
C: DATA

S: 354 Start mail input; end with <CRLF>.<CRLF>
Передачу письма необходимо завершить символами CRLF.CRLF
S: 250 OK
C: From: Drozd <drozd@mail.ru>
C: To: Drol <drol@mail.ru>
C: Subject: Hello
Между заголовком письма и его текстом не одна пара CRLF, а две.
C: Hello Drol!
C: You will be die on next week!
Заканчиваем передачу символами CRLF.CRLF
S: 250 OK
Теперь завершаем работу, отправляем команду QUIT:
S: QUIT
C: 221 smtp.mail.ru is closing transmission channel

Управляющие команды и непосредственно данные передаются по одному TCP-соединению (в отличиие от FTP). Все команды передаются в виде текстовых строк ASCII символов. Увидеть обмен командами SMTP можно на UNIX-хосте, запустив программу mail с ключом –v:

```
$ mail –v daniel@peanut.nuts.com
.
.
.        ← SMTP-обмен
.
sent.
```

## 11.2 Формат почтового сообщения Internet (RFC-822)

Почтовое сообщение состоит из трех частей: конверта, заголовка и тела сообщения. Пользователь видит только заголовок и тело сообщения. Конверт используется только программами доставки.

1. Конверт — данные, используемые транспортным агентом для доставки. В рассмотренном примере это две команды:

MAIL FROM:<drozd>

RCPT TO:<drol@mail.ru>

Содержимое и интерпретация конверта определяется RFC 821.

2. Заголовок всегда находится перед телом сообщения и отделен от него пустой строкой. Заголовок состоит из полей. Поля состоят из имени поля и содержания поля. Имя поля отделено от содержания символом «:». Минимально необходимыми являются поля `Date, From, cc` или `To`, например:

```
Date:    26 Aug 76 1429 EDT
From:    Jones@Registry.org
cc:
```

или

```
Date:    26 Aug 76 1429 EDT
From:    Jones@Registry.org
To:      Smith@Registry.org
```

Поле `Date` определяет дату отправки сообщения, поле `From` — отправителя, а поля `cc` и `To` — получателя(ей).

Кроме того, в заголовке могут использоваться следующие поля: `Subject` — определяет тему сообщения, `Reply-To` — пользователя, которому отвечают, `Comment` — комментарий, `In-Reply-To` — показывает, что сообщение относится к типу «В ответ на Ваше сообщение, отвечающее на сообщение, отвечающее ...».

3. Тело сообщения (body) — это содержимое сообщения. Представляет собой строки ASCII-текста длиной до 1000 байт. Тело сообщения отделяется от заголовка одной пустой строкой.

MUA (Mail User Agent — почтовый агент пользователя. Программа, при помощи которой пользователь читает и отправляет электронную почту) берет наши данные, добавляет некоторые заголовки, затем передает результат транспортному агенту. MTA (Mail Transfer Agent — программа рассылки электронной почты) добавляет некоторые заголовки, добавляет конверт и передает результат другому MTA.

Содержимое (content) часто определяется как комбинация заголовков и тела сообщения. Content передается клиентом при помощи команды DATA.

## 11.3 Расширения протокола SMTP. Протокол MIME

Предшественником MIME (Multipurpose Internet Mail Extensions — многоцелевые расширения почты Internet) является стандарт почтового сообщения ARPA (RFC822). Стандарт RFC822 был разработан для обмена текстовыми сообщениями, так в тело сообщения нельзя включить графику, аудио, видео, символы национальных алфавитов, бинарные данные и т.п. Эти ограничения преодолены путем определения расширений, структурирующих тело сообщения (RFC 1521, 1993 г.). Стандарт MIME добавляет несколько новых заголовков, определяющих структуру сообщения для реципиента. Он сориентирован на описание в заголовке письма структуры тела почтового сообщения и возможности составления письма из информационных единиц различных типов.

MIME определяет 5 заголовков:

1. Поле версии MIME (MIME-Version).

Поле версии указывается в заголовке почтового сообщения и позволяет определить, что сообщение подготовлено в стандарте MIME. Формат поля выглядит как:

MIME-Version: 1.0

2. Поле типа содержания тела почтового сообщения (Content-Type).

Поле типа используется для описания типа данных, которые содержатся в теле почтового сообщения. Это поле сообщает программе чтения почты какого сорта преобразования необходимы для того, чтобы сообщение правильно проинтерпретировать. Эта же информация используется и программой рассылки при кодировании/декодировании почты. Стандарт MIME определяет семь типов данных, которые можно передавать в теле письма: текст (text); смешанный тип (multipart); почтовое сообщение (message); графический образ (image); аудио информация (audio); фильм или видео (video); приложение (application); кавычки.

| Content-Type | Subtype | Description |
|---|---|---|
| text | plain<br>richtext<br>enriched | обычный текст<br>несложный форматированный текст |
| multiparty | mixed<br>parallel<br>digest<br>alternative | – несколько частей, обрабатывать отдельно<br>– то же, обрабатывать параллельно<br>– подпись<br>– семантически одинаковый контент, несколько вариантов представления |
| message | rfc 822<br>partial<br>external-body | – содержит другое RFC822 почт. сообщение<br>– фрагмент почтового сообщения<br>– указатель на местоположение сообщения |
| application | octet-stream<br>postscript | – произвольные бинарные данные<br>– программа PostScript |
| image | JPEG<br>GIF | – изображение в формате JPEG<br>– изображение в формате GIF |
| audio | basic | Звуковые данные |
| video | mpeg | Видеоизображение |

Заголовок Content-Type указывает получателю как именно нужно интерпретировать содержимое сообщения.

3. Заголовок Content-Transfer-Encoding поясняет каким образом извлечь закодированные данные из тела сообщения.

RSC 1521 определяет 5 различных форматов кодирования:

1) «7 бит», по умолчанию;

2) «quoted-printable»: каждый символ сообщения представляется в виде символа «=« и двузначного шестнадцатеричного кода символа.

Пример: 1 : «=49», = : «=61»

Этот способ кодирования применяется чаще для передачи текста. С его применением связаны значительные накладные расходы.

3) «base 64»: битовый поток разбивается на сегменты по 24 бита, которые делятся на части по 6 бит. Каждая такая часть кодируется одним из 64 ASCII символов. Когда число кодируемых

символов не кратно трем, в качестве заполнения используют символ «=»;

4) «8 bit» — содержит строки символов, некоторые из которых не-ASCII и содержат 8-й бит установленным в «1»;

5) «binary» — 8-ми битные данные, не содержат строк. Используются с расширением ESMTP 8BITMIME.

Многие данные передаются по почте в их исходном виде. Это могут быть 7bit символы, 8bit символы, 64base символы и т.п. Однако при работе в разнородных почтовых средах необходимо определить механизм их представления в стандартном виде — US-ASCII. Для этого существуют процедуры кодирования такого сорта данных. Наиболее широко применяемая — uuencode. Для того, чтобы при получении данные были бы правильно распакованы и введено в стандарт **поле «Content-Transfer-Encoding».** Синтаксис этого поля следующий:

Content-Transfer-Encoding:= «BASE64» / «QUOTED-PRINTABLE» /

«8BIT» / «7BIT» /

«BINARY» / x-token

Каждая из альтернатив применяется в своем подходящем случае. Альтернативы «8bit», «7bit», «BINARY» реально никакого преобразования не требуют, так как почта передается байтами и SMTP не делает различия между ними. Однако они введены для строгости описания типов. «BASE64» обычно используется в связке с типом «text/ISO-8859-1», «x-token» позволяет пользователю описать свою процедуру преобразования.

4. Content-ID — определяет уникальный идентификатор содержания.

5. Content-Description — служит для комментария содержания.

4, 5 — дополнительные необязательные поля. Ни то, ни другое программами просмотра обычно не отображаются.

## 11.4 Доступ пользователя к своему почтовому ящику

**Протокол POP3 (Post Office Protocol) (RFC 1725, 1939)**

Протокол обмена почтовой информацией POP3 предназначен для разбора почты из почтовых ящиков пользователей на их

рабочие места при помощи программ-клиентов. Если по протоколу SMTP пользователи отправляют корреспонденцию через Internet, то по протоколу POP3 пользователи получают корреспонденцию из своих почтовых ящиков на почтовом сервере в локальные файлы.

Почтовый ящик пользователя обычно представляет собой текстовый файл, состоящий из сообщений в формате RFC 822. Бывают и другие форматы, но это зависит от системы и реализации электронной почты. Пользователь может открыть телнет-сессию на почтовый хост и читать почту при помощи любого MUA (Mail User Agent — почтовый агент пользователя; программа, при помощи которой пользователь читает и отправляет электронную почту), запущенного на этом хосте. Это не всегда удобно. Если у пользователя нет shell (функция Unix Shell позволяет работать в текстовом режиме операционной системы семейства Unix) на хосте, он не сможет этого сделать. Чаще всего почта перемещается на ПК пользователя, где он читает ее при помощи более удобных программ (Outlook Express, The Bat! и т.п.). Для этого перемещения часто используют протокол POP3 (post office protocol). Сервер POP3 работает на порту 110/ TCP.

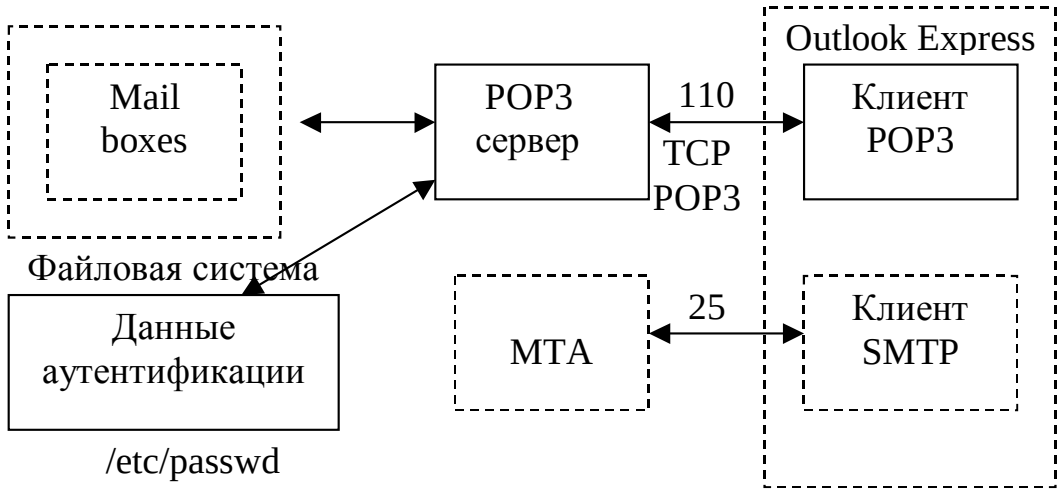



Рисунок 11.1 — Работы протокола POP3

POP3 — простой протокол, он позволяет только забрать почту на хост клиента и удалить сообщение на сервере. POP3 не предназначен для отправки почты. Команды протокола состоят

из ключевых слов и параметров и разделяются пробелом. Ответы сервера состоят из строки статус-индикатора, ключевого слова, строки дополнительной информации.

«+OK» или «-ERR» — статус-индикатор.

Сессия POP3 состоит из нескольких фаз:

1. Authorization state — клиент регистрируется на сервере: команды USER и PASS.

2. Transaction state — клиенту предоставляется его почтовый ящик. Клиент может читать и удалять сообщения из почтового ящика. Фактически, почта копируется во временный файл, который блокируется.

Основные команды:

STAT — просмотр состояния почтового ящика;

LIST [n] — информация о каждом сообщении в отдельности;

RETR msg — запросить конкретное сообщение с номером msg;

DELE msg — удалить сообщение msg;

LAST — номер последнего прочитанного сообщения;

TOP msg n — выводит «n» первых строк сообщения (сервер отдает заголовок +n строк);

Rset — откат транзакции, удаленные сообщения восстанавливаются;

Noop — пустая команда;

3. Update state — клиент выдает команду QUIT — завершение сеанса, все изменения подтверждаются.

Сервер отсчитывает паузы после каждой команды. Если активности нет — откат транзакции и завершение соединения (часто 10 минут).

Существуют расширения протокола POP3, например команды:

UIDL — уникальный «слепок» письма, используется POP3 клиентами, чтобы повторно не забирать уже прочитанные сообщения с сервера;

APOP — более защищенная аутентификация пользователя.

Современные POP-серверы поддерживают различные методы аутентификации, не только через системный файл паролей, но также могут использовать другие схемы и системы аутентификации (например, RADIUS).

POP3 — простой, но недостаточно гибкий протокол и в настоящее время заменяется на IMAP4 (Inernet Message Access Protocol v.4).

# 12 КОНТРОЛЬНЫЕ РАБОТЫ

По дисциплине необходимо выполнить две контрольные работы. Контрольная работа № 1 выполняется в компьютерном виде, контрольная работа № 2 — текстовая, содержит 30 вариантов, в каждом варианте три задачи. Выбор вариантов по общим правилам.

## 12.1 Примеры решения задач к контрольной работе № 2

### Задание 1

***Условие задачи***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500002F116A00001E0612F4C02AFC01C02AFC14.

По принятой информации определить IP-адрес узла назначения и протокол верхнего уровня, использующий данный пакет.

***Решение***

Во-первых, в принятой последовательности символов следует выделить компоненты кода, размещенные в полях заголовка IP-пакета

«Протокол» и «IP-адрес назначения».

Каждый байт заголовка представлен двумя символами шестнадцатеричного кода. В соответствии с рис. 3.2 учебного пособия полю «Протокол» соответствует байт заголовка с номером 10, полю «IP-адрес назначения» — байты заголовка с номерами от 17 до 20. После определения местоположения этих байтов получаем, что в этих полях размещены следующие сегменты кода заголовка:

0x06 и 0xC02AFC14.

Код адреса удобнее разбить на байты

0xC0, 0x2A, 0xFC, 0x14.

Это позволяет побайтно перевести шестнадцатеричный код в десятичный и записать IP-адрес в десятичной нотации

192.42.252.20.

Из таблицы 3.1 учебного пособия определяем, что сеть с таким адресом относится к сетям класса А.

И, наконец, шестнадцатеричный код 0x06 соответствует десятичному коду 06. Из таблицы 1 определяем, что этому коду в поле «Протокол» соответствует протокол транспортного уровня TCP.

В таблице 1 для сведения приведены номера некоторых протоколов, помещаемых в поле «Протокол» заголовка IP-па-кета. Полный перечень номеров протоколов опубликован в стандарте RFC 1700.

Таблица 1 — **Коды протоколов**

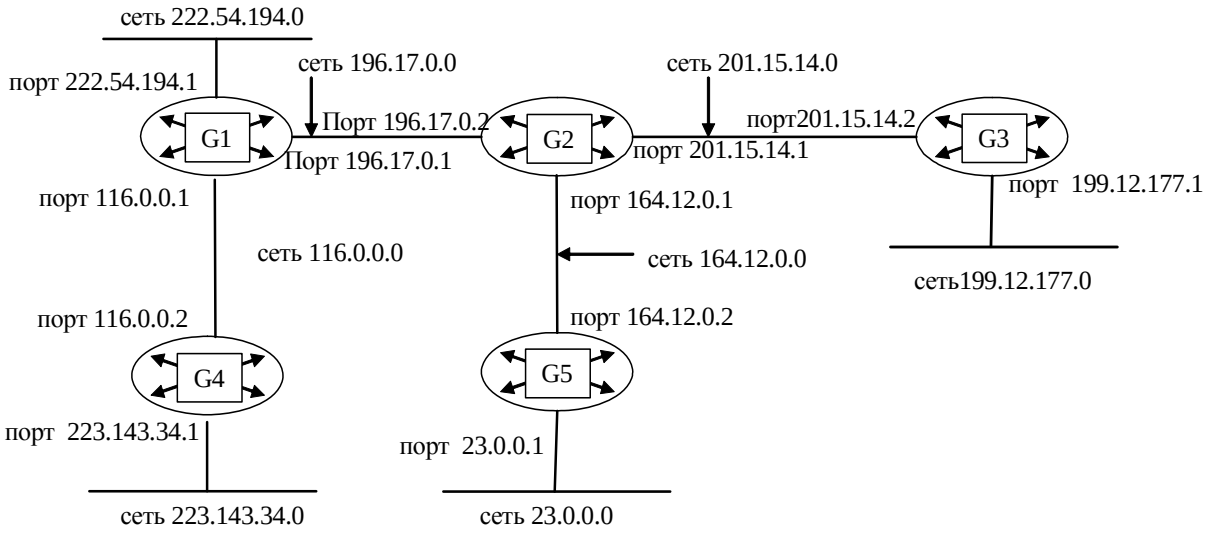
| Код протокола | Название протокола | Назначение протокола |
|---|---|---|
| 1 | ICMP | Протокол контрольных сообщений [RFC 792] |
| 2 | IGMP | Групповой протокол управления [RFC 1112] |
| 3 | GGP | Протокол маршрутизатор-маршрутизатор [RFC-823] |
| 4 | IP | IP поверх IP (инкапсуляция/туннели) |
| 5 | ST | Поток [RFC 1190] |
| 6 | TCP | Протокол управления передачей [RFC-793] |
| 8 | EGP | Протокол внешней маршрутизации [RFC-888] |
| 9 | IGP | Протокол внутренней маршрутизации |
| 10 | BBN-MON | BBN-RCC мониторирование |
| 11 | NVP-II | Сетевой протокол для голосовой связи [RFC-741] |
| 15 | Xnet | Перекрестный сетевой отладчик [IEN158] |
| 17 | UDP | Протокол дейтограмм пользователя [RFC-768] |
| 18 | MUX | Мультиплексирование [IEN90] |
| 19 | DCN-MEAS | DCN измерительные субсистемы |
| 20 | HMP | Протокол мониторирования ЭВМ (host [RFC-869]) |
| 27 | RDP | Протокол для надежной передачи данных [RFC-908] |
| 28 | IRTP | Надежный TP для Интернет [RFC-938] |
| 29 | ISO-TP4 | ISO транспортный класс 4 [RFC-905] |

Коды остальных полей заголовка пакета определяются аналогичным образом.

## Задание 2

***Условие задачи***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G2, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G2;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Решение***

Таблицу маршрутизации удобнее начинать с номеров сетей, непосредственно подключенных к данному маршрутизатору. Первые строчки таблицы в этом случае будут иметь вид:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| | | | |
| 196.17.0.0 | – | 196.17.0.2 | 0 (подсоединена) |
| 201.15.14.0 | – | 201.15.14.1 | 0 (подсоединена) |
| ... | | | |
| ... | | | |
| ... | | | |

Затем следует поочередно описывать сети по мере удаления от данного маршрутизатора. В конце концов, для наиболее удаленных от маршрутизатора сетей будут сформированы записи вида:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| ... | | | |
| ... | | | |
| ... | | | |
| 203.143.34.0 | 196.17.0.1 | 196.17.0.2 | 2 |
| 199.12.177.0 | 201.15.14.2 | 201.15.14.1 | 1 |

## Задание 3

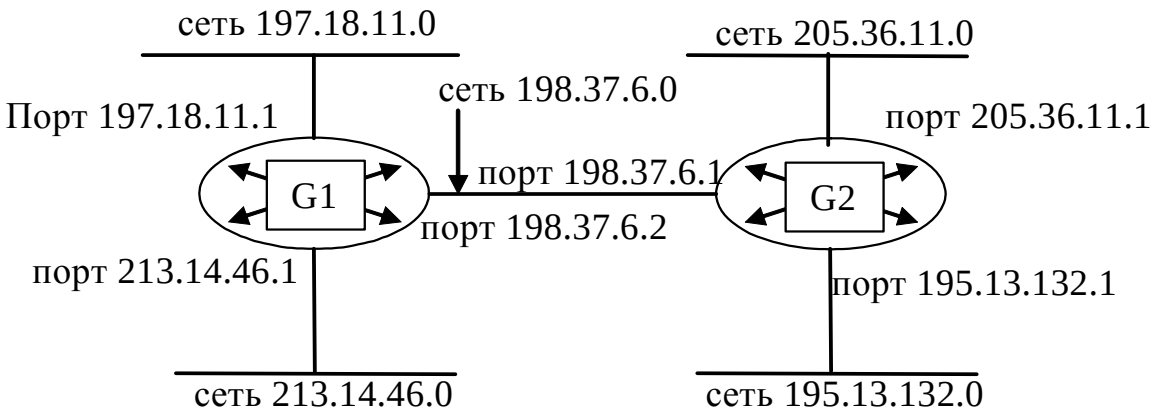
***Условие задачи***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрути-затора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 195.13.1323.0 | 198.37.6.1 | 198.37.6.2 | 1 |
| 197.18.11.0 | – | 197.18.11.1 | 0 (подсоединена) |
| 198.37.6.0 | – | 198.37.6.2 | 0 (подсоединена) |
| 205.36.11.0 | 198.37.6.1 | 198.37.6.2 | 1 |
| 213.14.46.0 | – | 213.14.46.1 | 0 (подсоединена) |

***Решение***

Построение схемы сети также следует начинать с сетей, непосредственно подсоединенных к данному маршрутизатору. Затем поочередно подключают сети по мере их удаления от исходного узла. Окончательный вариант схемы рассматриваемой сети может иметь вид:




## 12.2 Задания к контрольной работе № 2

### Вариант № 1

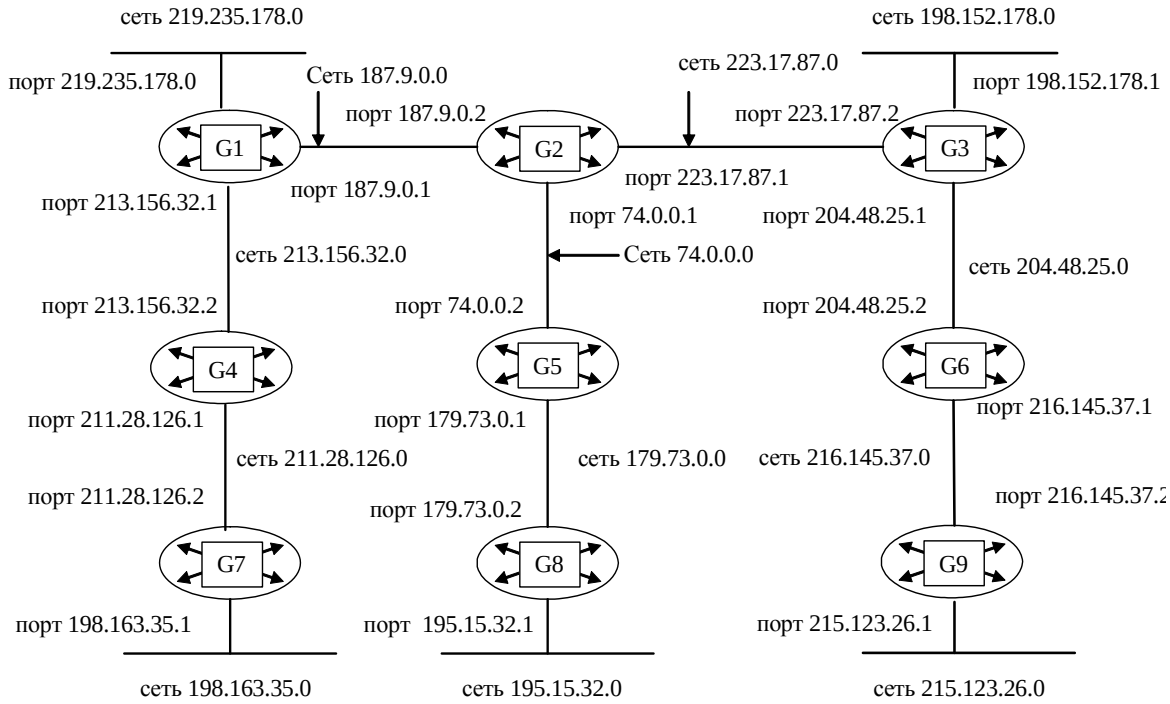
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x45000103116A000043111256C24A7C32C32B5D13.

По принятой информации определить параметр «Время жизни пакета» и IP-адрес узла источника (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G1, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G1;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 139.6.0.0 | – | 139.6.0.2 | 0 (подсоединена) |
| 191.132.144.0 | 198.152.0.1 | 198.152.0.2 | 1 |

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 196.9.98.0 | 198.152.0.1 | 198.152.0.2 | 1 |
| 198.152.0.0 | – | 198.152.0.2 | 0 (подсоединена) |
| 209.175.136.0 | 198.152.0.1 | 198.152.0.2 | 2 |
| 214.198.126.0 | – | 214.198.126.2 | 0 (подсоединена) |

**Вариант № 2**

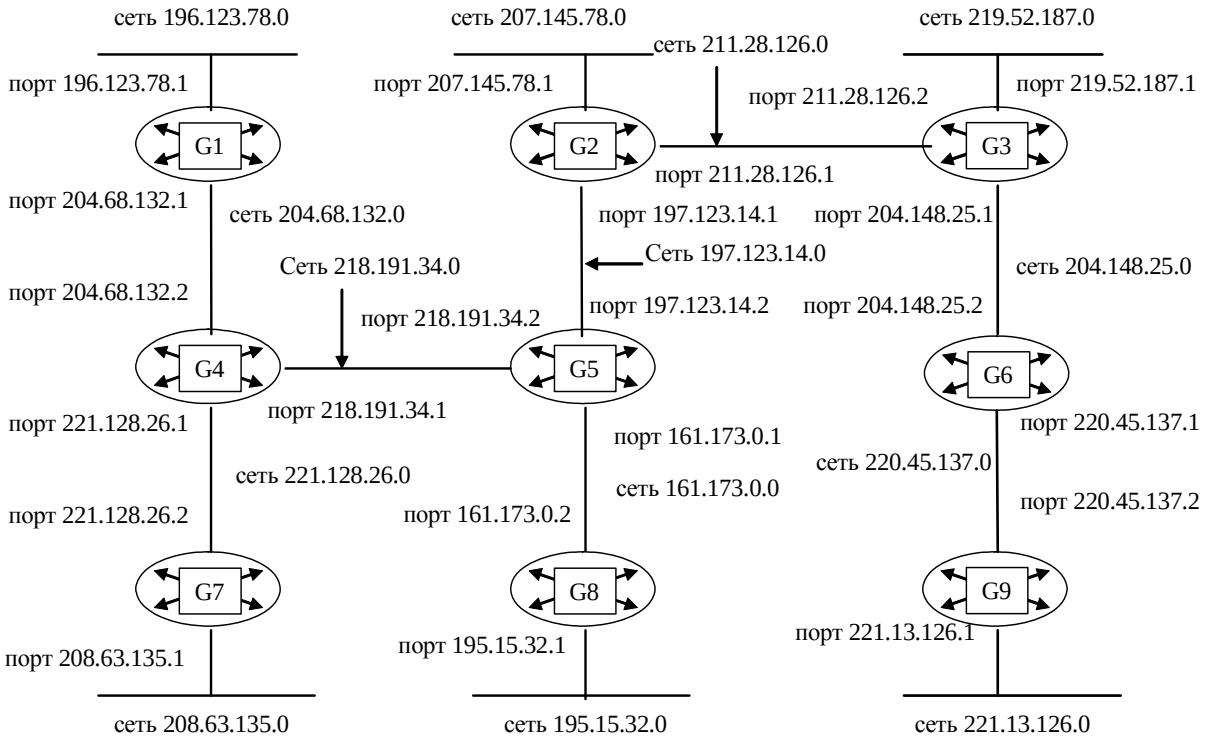
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x450000342356000013082 56AA0357C32A17B5D14.

По принятой информации определить протокол верхнего уровня, использующий данный пакет; а также указать класс адресов сети, в которой расположен узел источника, и класс адресов сети, в которой расположен узел приемника.

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G2, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G2;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 19.0.0.0 | – | 19.0.0.2 | 0 (подсоединена) |
| 124.89.0.0 | 195.12.14.1 | 195.12.14.2 | 2 |
| 195.12.14.0 | – | 195.12.14.2 | 0 (подсоединена) |
| 196.134.0.0 | 195.12.14.1 | 195.12.14.2 | 1 |
| 197.49.0.0 | 195.12.14.1 | 195.12.14.2 | 1 |
| 212.245.55.0 | – | 212.245.55.2 | 0 (подсоединена) |

**Вариант № 3**

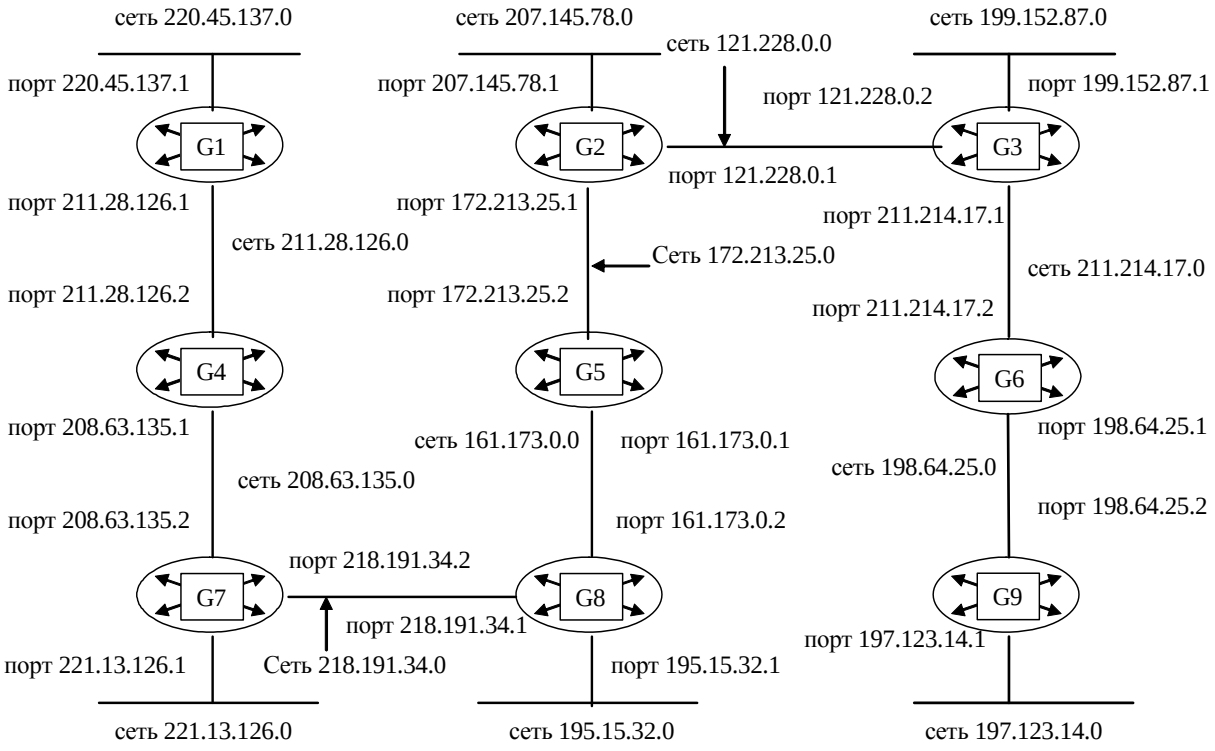
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500002134020000220B3A45126F037D127B8D52.

По принятой информации определить общую длину дейтаграммы и IP-адрес узла назначения (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G3, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G3;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 164.108.0.0 | 196.213.48.1 | 196.213.48.2 | 1 |
| 196.213.48.0 | – | 196.213.48.2 | 0 (подсоединена) |

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 199.87.152.0 | 201.44.37.1 | 201.44.37.2 | 1 |
| 201.44.37.0 | – | 201.44.37.2 | 0 (подсоединена) |
| 204.89.167.0 | 201.44.37.1 | 201.44.37.2 | 1 |
| 219.168.54.0 | – | 219.168.54.2 | 0 (подсоединена) |

**Вариант № 4**

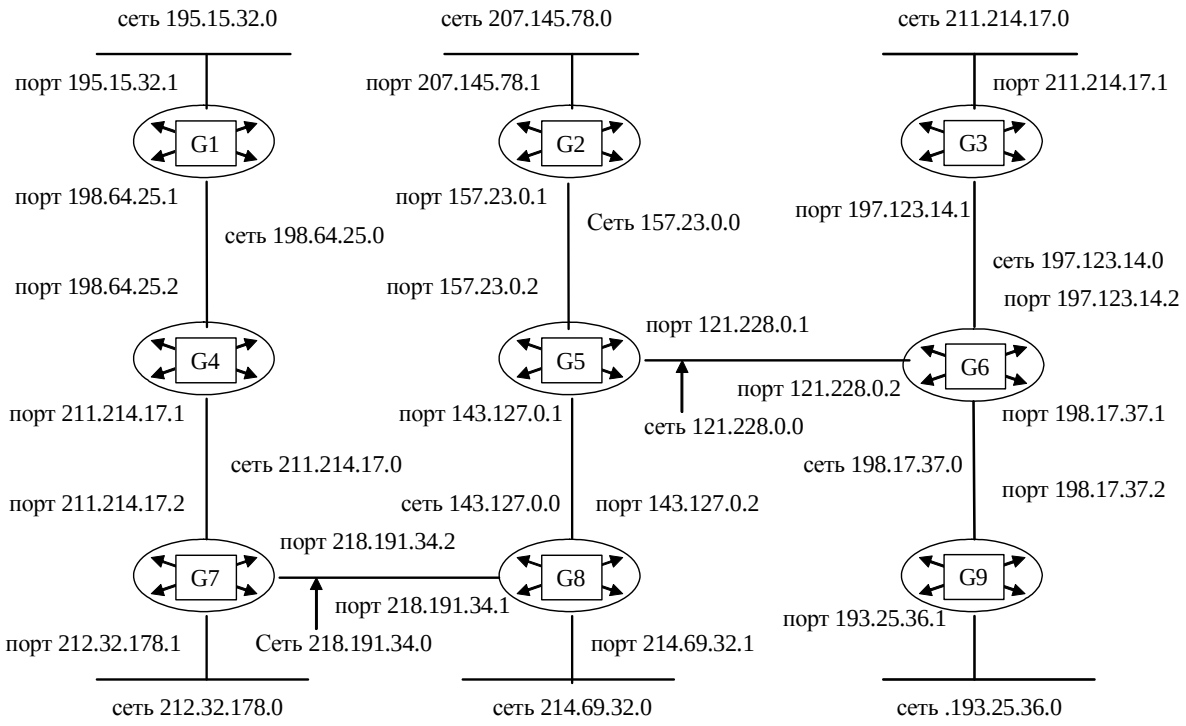
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500004478D200000D1120B3B1326F07B27C8B7D.

По принятой информации определить время жизни пакета и IP-адрес узла источника (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G4, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G4;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 172.132.0.0 | 209.132.95.1 | 209.132.95.2 | 1 |
| 197.163.47.0 | 214.157.13.1 | 214.157.13.2 | 1 |
| 198.145.17.0 | – | 198.145.17.2 | 0 (подсоединена) |
| 209.132.95.0 | – | 209.132.95.2 | 0 (подсоединена) |
| 214.157.13.0 | – | 214.157.13.2 | 0 (подсоединена) |
| 217.136.47.0 | 209.132.95.1 | 209.132.95.2 | 1 |

## Вариант № 5

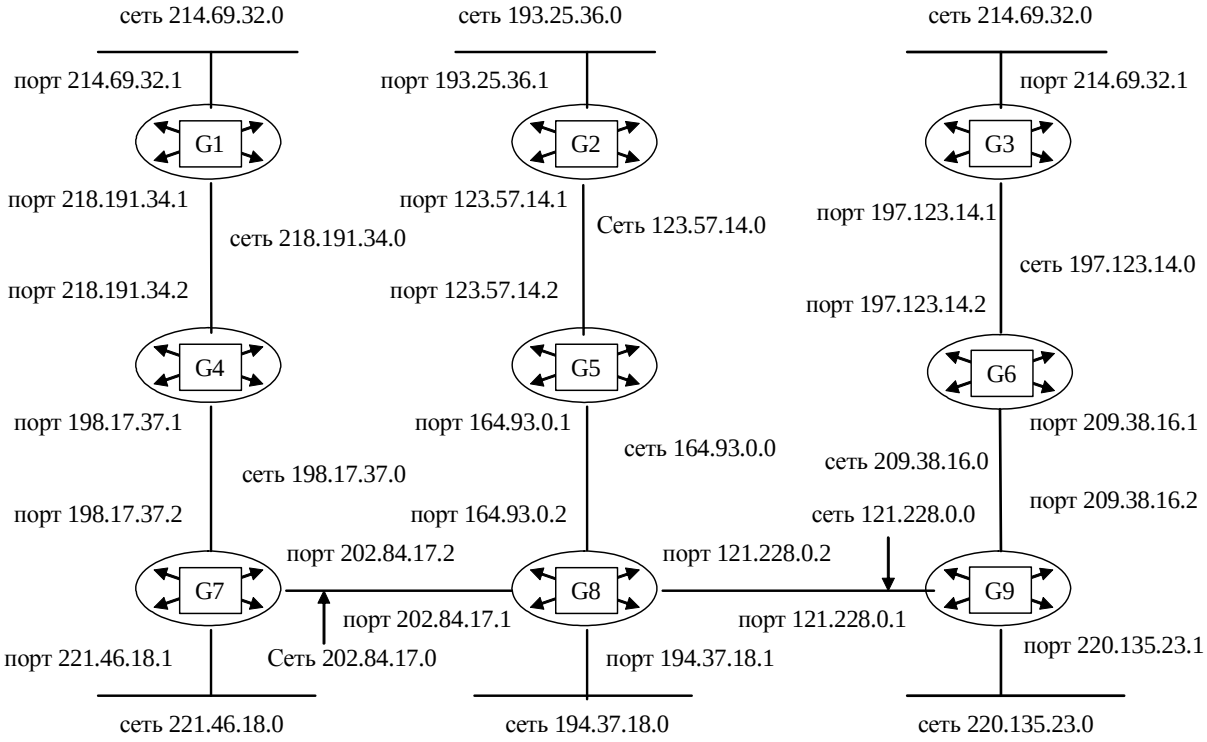
### *Задание 1*

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x450000A4B132000017018B7A600720B360078D27.

По принятой информации определить общую длину дейтаграммы и IP-адрес узла назначения (в десятичной нотации).

### *Задание 2*

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G5, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G5;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 137.217.0.0 | 203.52.126.1 | 203.52.126.2 | 1 |
| 195.164.17.0 | 203.52.126.1 | 203.52.126.2 | 1 |
| 198.156.32.0 | – | 198.156.32.2 | 0 (подсоединена) |
| 203.52.126.0 | – | 203.52.126.2 | 0 (подсоединена) |
| 212.203.78.0 | – | 212.203.78.2 | 0 (подсоединена) |
| 213.136.89.0 | 203.52.126.1 | 203.52.126.2 | 2 |

**Вариант № 6**

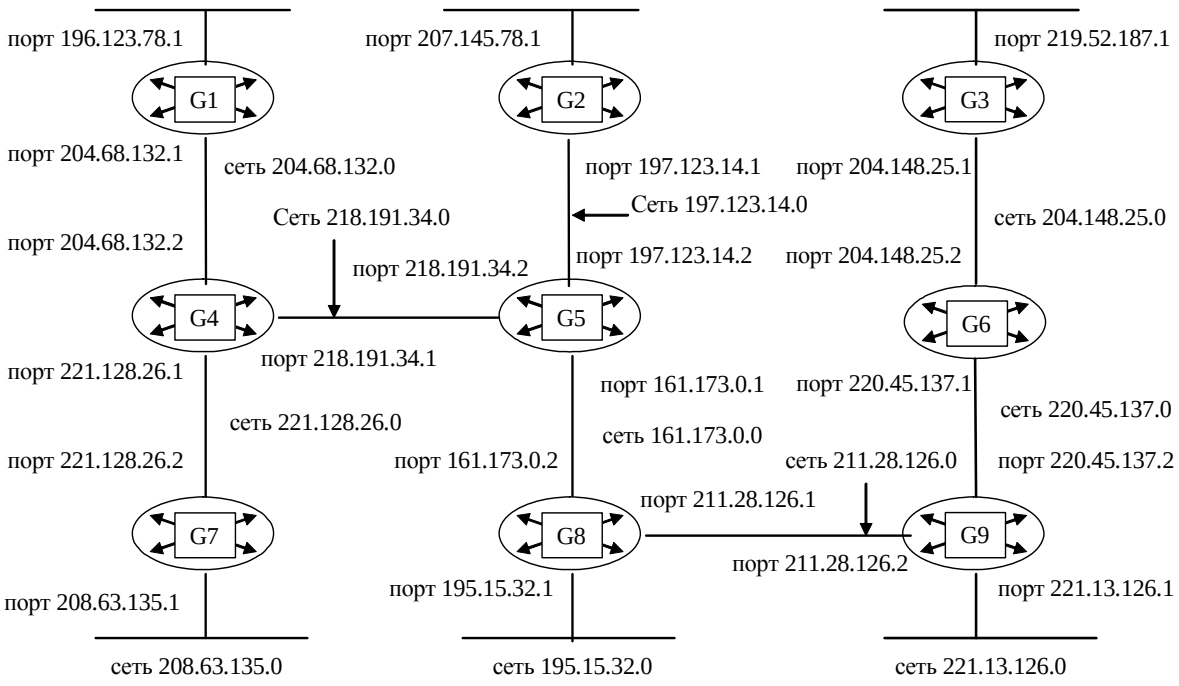
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500002320B3000067046007C1372B51C25D8B7A.

По принятой информации определить протокол верхнего уровня, использующий данный пакет, а также указать класс адресов сети, в которой расположен узел источника, и класс адресов сети, в которой расположен узел приемника.

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G6, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G6;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 146.163.0.0 | 194.59.27.1 | 194.59.27.2 | 1 |
| 194.59.27.0 | – | 194.59.27.2 | 0 (подсоединена) |
| 197.138.59.0 | 194.59.27.1 | 194.59.27.2 | 1 |
| 203.48.135.0 | – | 203.48.135.2 | 0 (подсоединена) |
| 205.201.129.0 | 194.59.27.1 | 194.59.27.2 | 2 |
| 218.215.31.0 | – | 218.215.31.2 | 0 (подсоединена) |

**Вариант № 7**

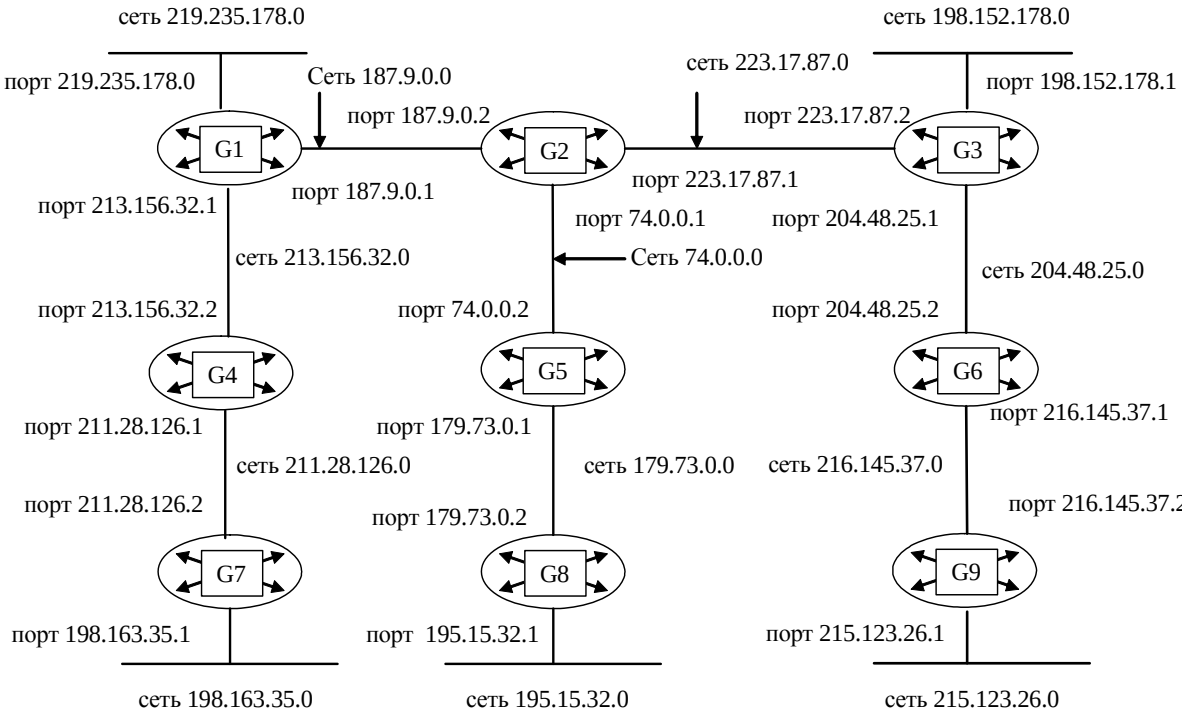
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x450001058B7A000089062B51AC137825A146C25D.

По принятой информации определить время жизни пакета и IP-адрес узла источника (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G7, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G7;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 165.204.0.0 | 193.48.97.1 | 193.48.97.2 | 1 |
| 193.48.97.0 | – | 193.48.97.2 | 0 (подсоединена) |
| 194.76.187.0 | 200.137.94.1 | 200.137.94.2 | 1 |

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 200.137.94.0 | – | 200.137.94.2 | 0 (подсоединена) |
| 212.134.65.0 | – | 212.134.65.2 | 0 (подсоединена) |
| 219.142.153.0 | 200.137.94.1 | 200.137.94.2 | 1 |

**Вариант № 8**

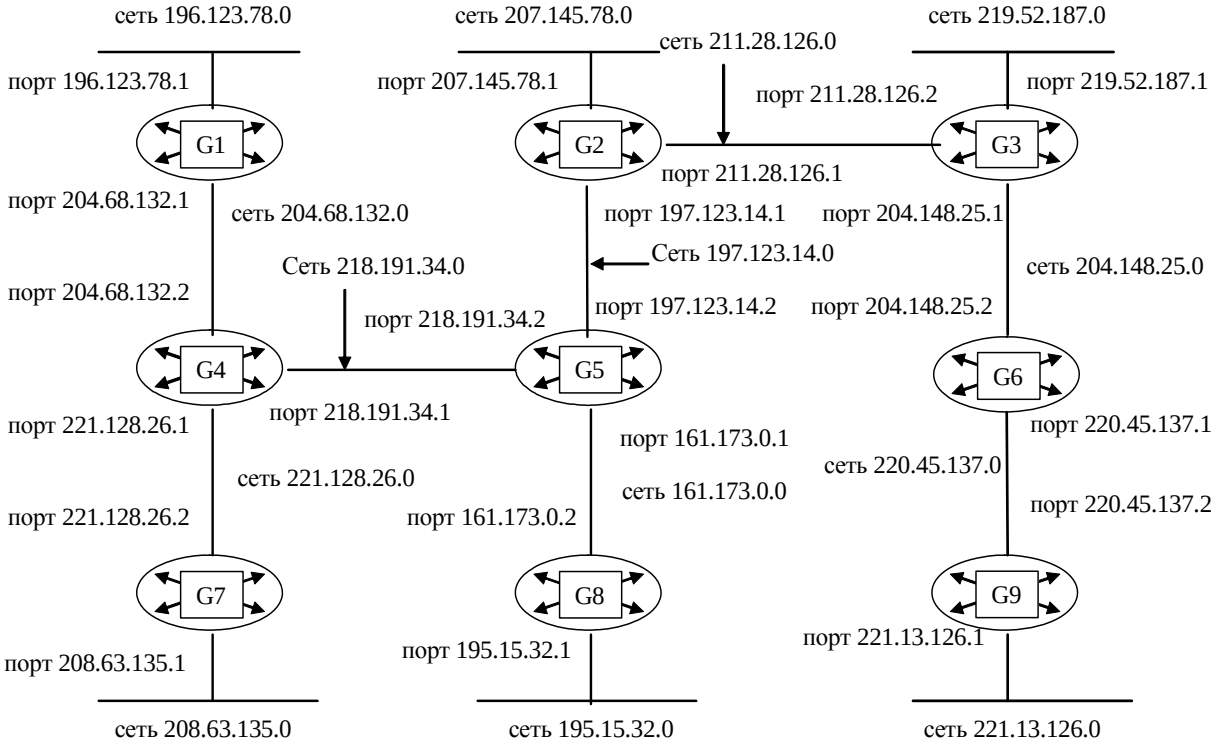
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500004078250000 8909C25D20B5AC1320A7F146.

По принятой информации определить общую длину дейтаграммы и IP-адрес узла назначения (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G8, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G8;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 132.194.0.0 | 208.132.64.1 | 208.132.64.2 | 1 |
| 195.164.138.0 | – | 195.164.138.2 | 0 (подсоединена) |
| 196.47.135.0 | 195.164.138.1 | 195.164.138.2 | 1 |
| 208.132.64.0 | – | 208.132.64.2 | 0 (подсоединена) |
| 213.164.127.0 | – | 213.164.127.2 | 0 (подсоединена) |
| 221.143.79.0 | 195.164.138.1 | 195.164.138.2 | 1 |

**Вариант № 9**

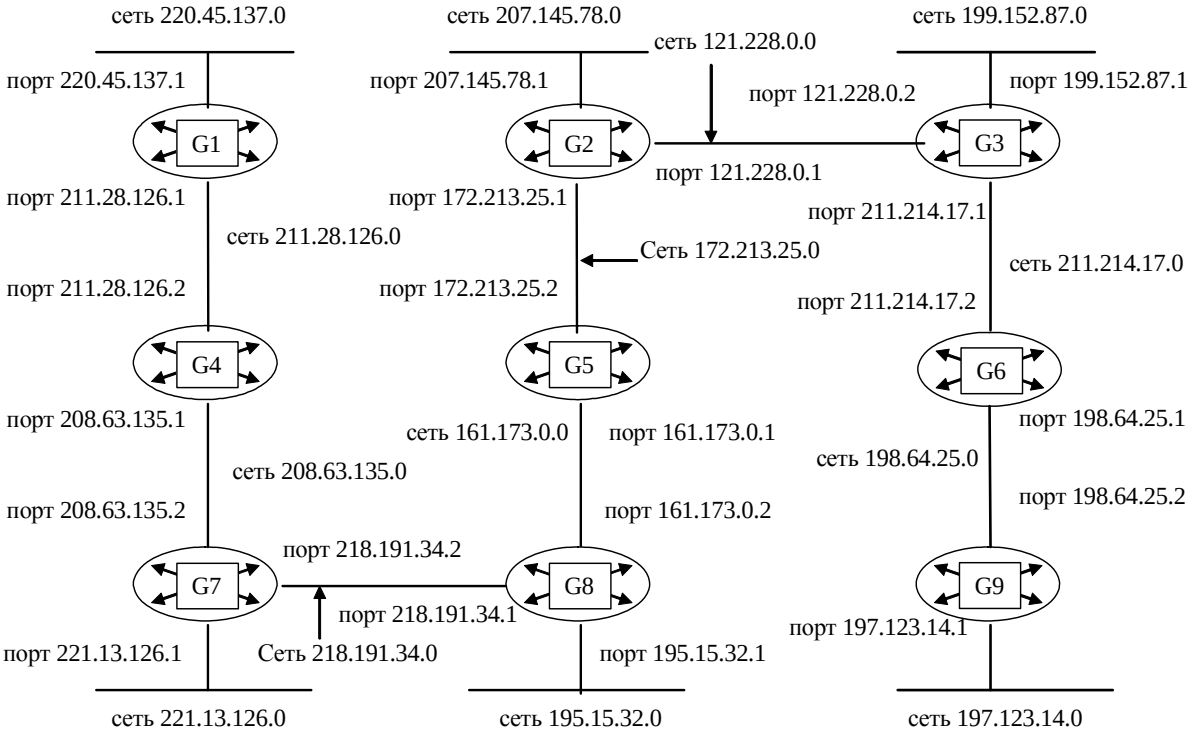
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x45000035AC1300003702F146C25D20B5C1A320A7.

По принятой информации определить протокол верхнего уровня, использующий данный пакет, а также указать класс адресов сети, в которой расположен узел источника, и класс адресов сети, в которой расположен узел приемника.

***Задание 2***

Сеть некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G9, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G9;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизато- | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|

| | ра | | |
|---|---|---|---|
| 144.216.0.0 | 201.48.137.1 | 201.48.137.2 | 1 |
| 195.84.132.0 | 201.48.137.1 | 201.48.137.2 | 1 |
| 197.164.38.0 | – | 197.164.38.2 | 0 (подсоединена) |
| 201.48.137.0 | – | 201.48.137.2 | 0 (подсоединена) |
| 213.89.165.0 | 201.48.137.1 | 201.48.137.2 | 2 |
| 217.68.213.0 | – | 217.68.213.2 | 0 (подсоединена) |

**Вариант № 10**

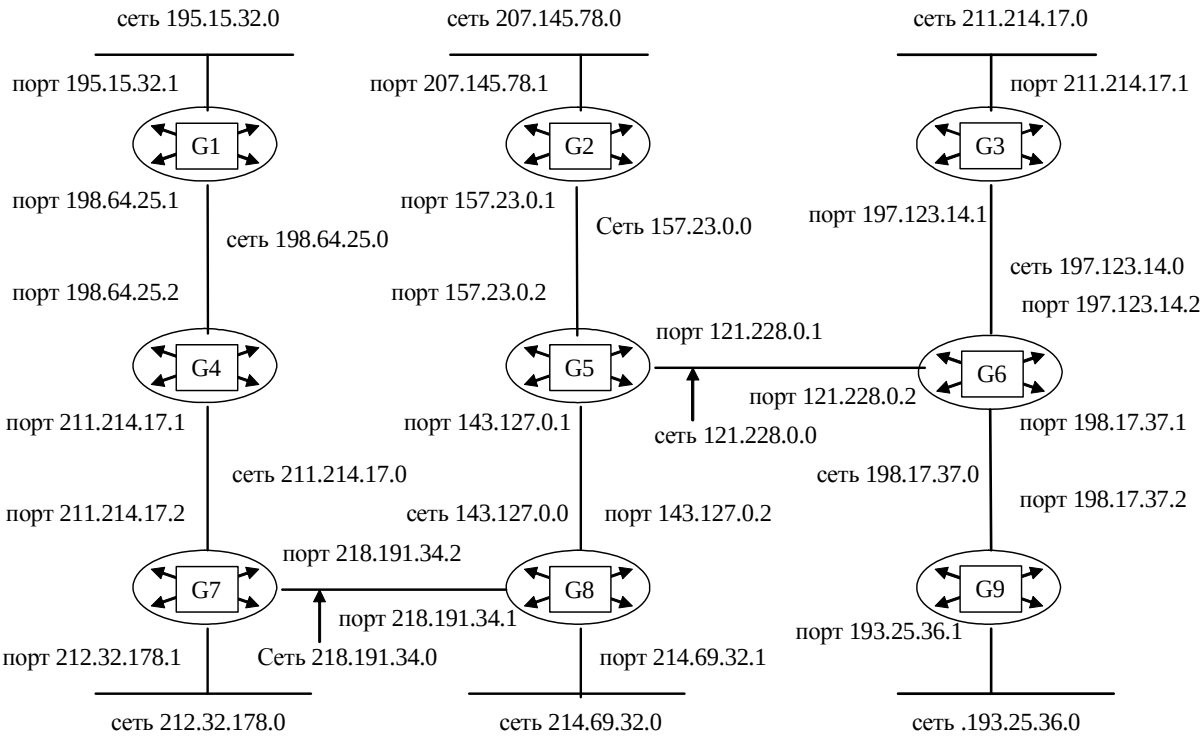
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500003020A700004506C25DB147B520B2C1A315.

По принятой информации определить время жизни пакета и IP-адрес узла источника (в десятичной нотации).

***Задание 2***

Сеть некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G1, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G1;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 157.186.0.0 | 202.154.73.1 | 202.154.73.2 | 1 |
| 193.178.93.0 | – | 193.178.93.2 | 0 (подсоединена) |
| 195.78.141.0 | 202.154.73.1 | 202.154.73.2 | 1 |
| 202.154.73.0 | – | 202.154.73.2 | 0 (подсоединена) |
| 215.74.146.0 | – | 215.74.146.2 | 0 (подсоединена) |
| 217.154.67.0 | 202.154.73.1 | 202.154.73.2 | 2 |

**Вариант № 11**

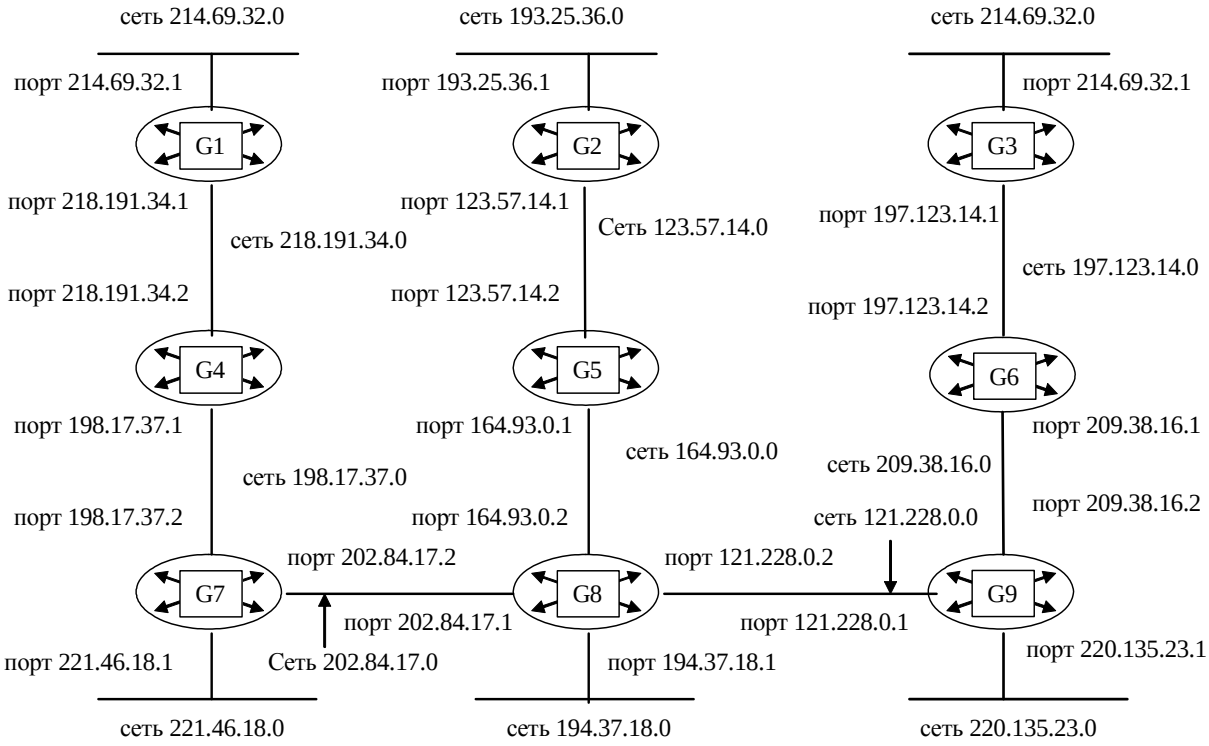
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x45000103116A000043111256C24A7C32C32B5D13.

По принятой информации определить общую длину дейтаграммы и IP-адрес узла назначения (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G2, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G2;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 173.148.0.0 | 206.87.153.1 | 206.87.153.2 | 1 |
| 194.132.61.0 | – | 194.132.61.2 | 0 (подсоединена) |

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 198.164.95.0 | 206.87.153.1 | 206.87.153.2 | 1 |
| 206.87.153.0 | – | 206.87.153.2 | 0 (подсоединена) |
| 211.75.146.0 | 206.87.153.1 | 206.87.153.2 | 2 |
| 218.76.148.0 | – | 218.76.148.2 | 0 (подсоединена) |

**Вариант № 12**

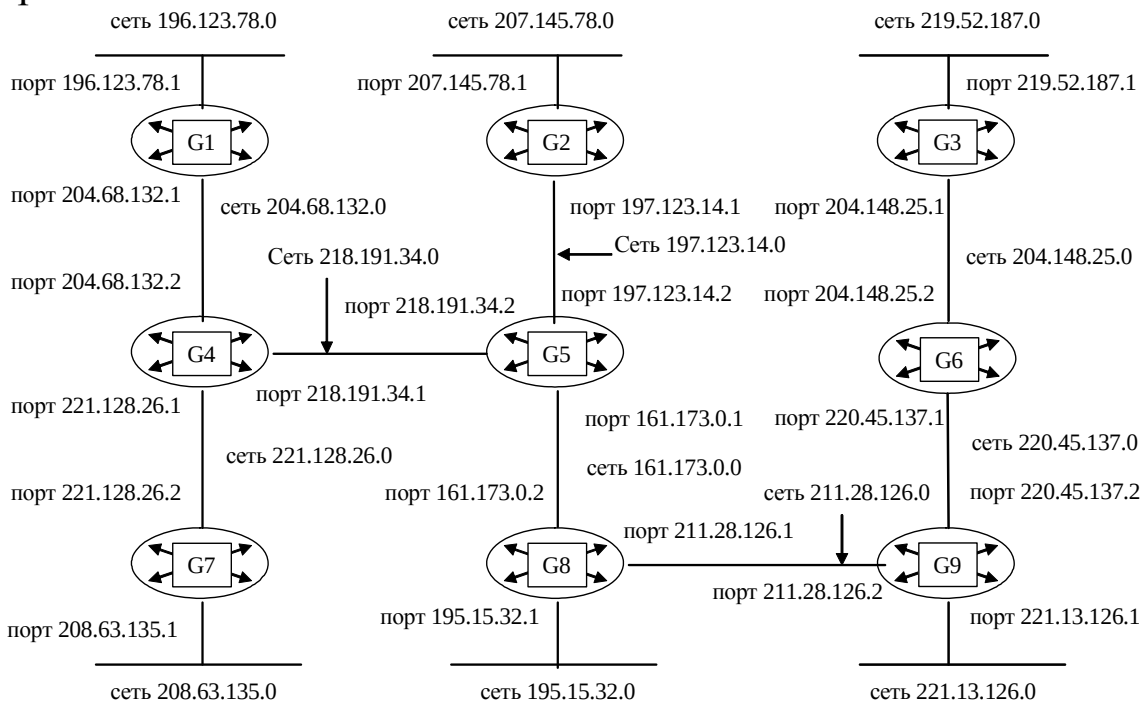
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x450000342356000013082 56AA0357C32A17B5D14.

По принятой информации определить протокол верхнего уровня, использующий данный пакет; а также указать класс адресов сети, в которой расположен узел источника, и класс адресов сети, в которой расположен узел приемника.

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G3, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G3;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 187.219.0.0 | 206.148.97.1 | 206.148.97.2 | 1 |
| 196.57.142.0 | – | 196.57.142.2 | 0 (подсоединена) |
| 197.86.124.0 | 206.148.97.1 | 206.148.97.2 | 1 |
| 206.148.97.0 | – | 206.148.97.2 | 0 (подсоединена) |
| 213.156.21.0 | 206.148.97.1 | 206.148.97.2 | 2 |
| 214.85.139.0 | – | 214.85.139.2 | 0 (подсоединена) |

**Вариант № 13**

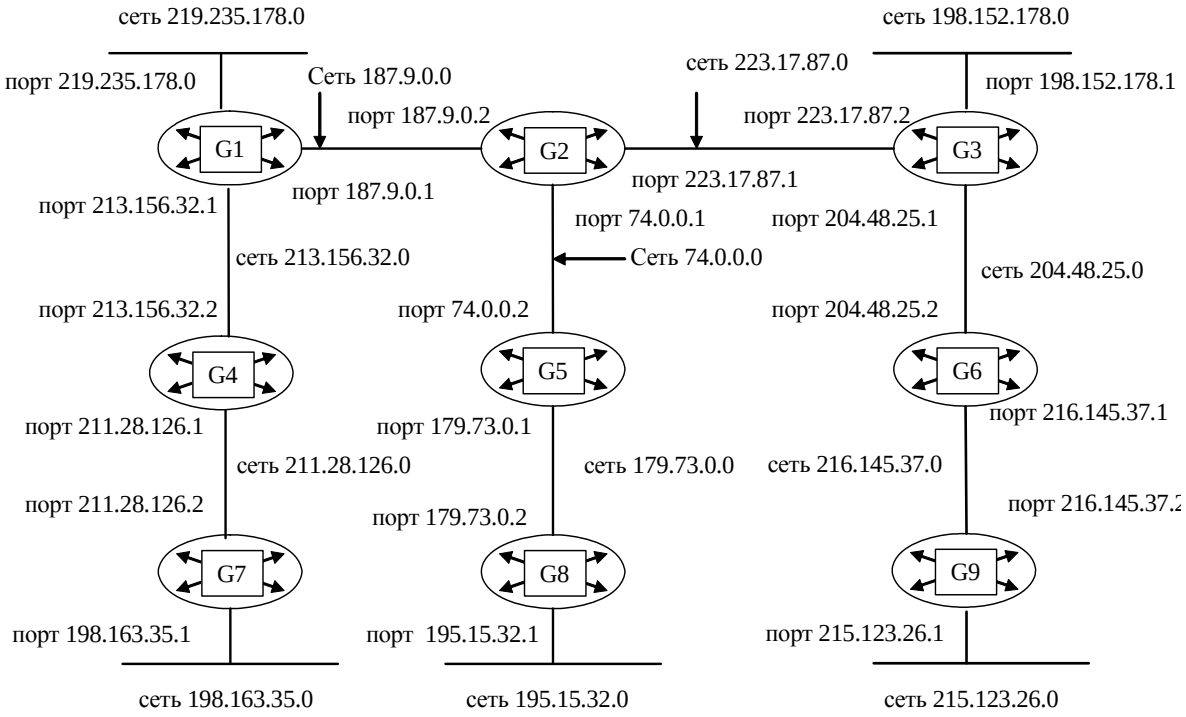
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500002134020000220B3A45126F037D127B8D52.

По принятой информации определить параметр «Время жизни пакета» и IP-адрес узла источника (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G4, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G4;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 178.120.0.0 | 199.147.68.1 | 199.147.68.2 | 1 |
| 195.138.24.0 | 204.86.213.1 | 204.86.213.2 | 1 |
| 199.147.68.0 | – | 199.147.68.2 | 0 (подсоединена) |

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 204.86.213.0 | – | 204.86.213.2 | 0 (подсоединена) |
| 213.54.134.0 | – | 213.54.134.2 | 0 (подсоединена) |
| 218.62.173.0 | 204.86.213.1 | 204.86.213.2 | 1 |

**Вариант № 14**

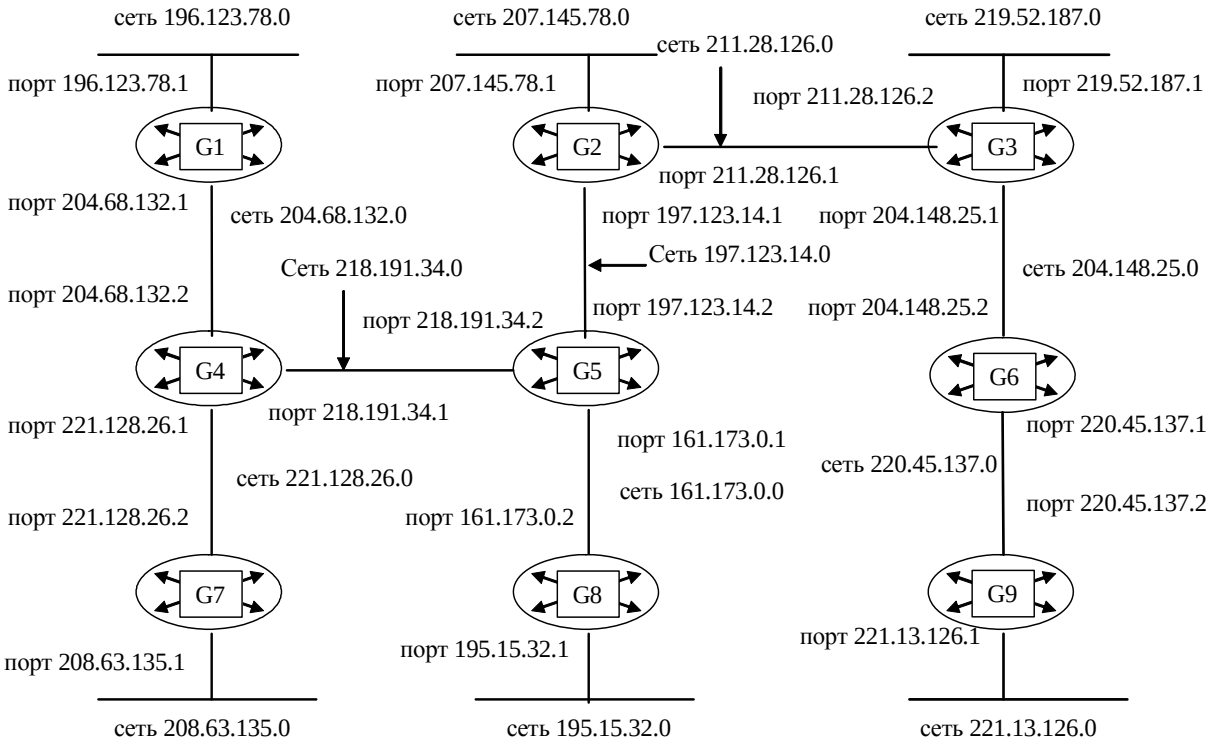
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500004478D200000D1120B3B1326F07B27C8B7D.

По принятой информации определить общую длину дейтаграммы и IP-адрес узла назначения (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G5, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G5;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 148.176.0.0 | 205.14.132.1 | 205.14.132.2 | 1 |
| 193.211.62.0 | 196.29.175.1 | 196.29.175.2 | 1 |
| 196.29.175.0 | – | 196.29.175.2 | 0 (подсоединена) |
| 205.14.132.0 | – | 205.14.132.2 | 0 (подсоединена) |
| 216.23.146.0 | – | 216.23.146.2 | 0 (подсоединена) |
| 217.154.63.0 | 196.29.175.1 | 196.29.175.2 | 1 |

**Вариант № 15**

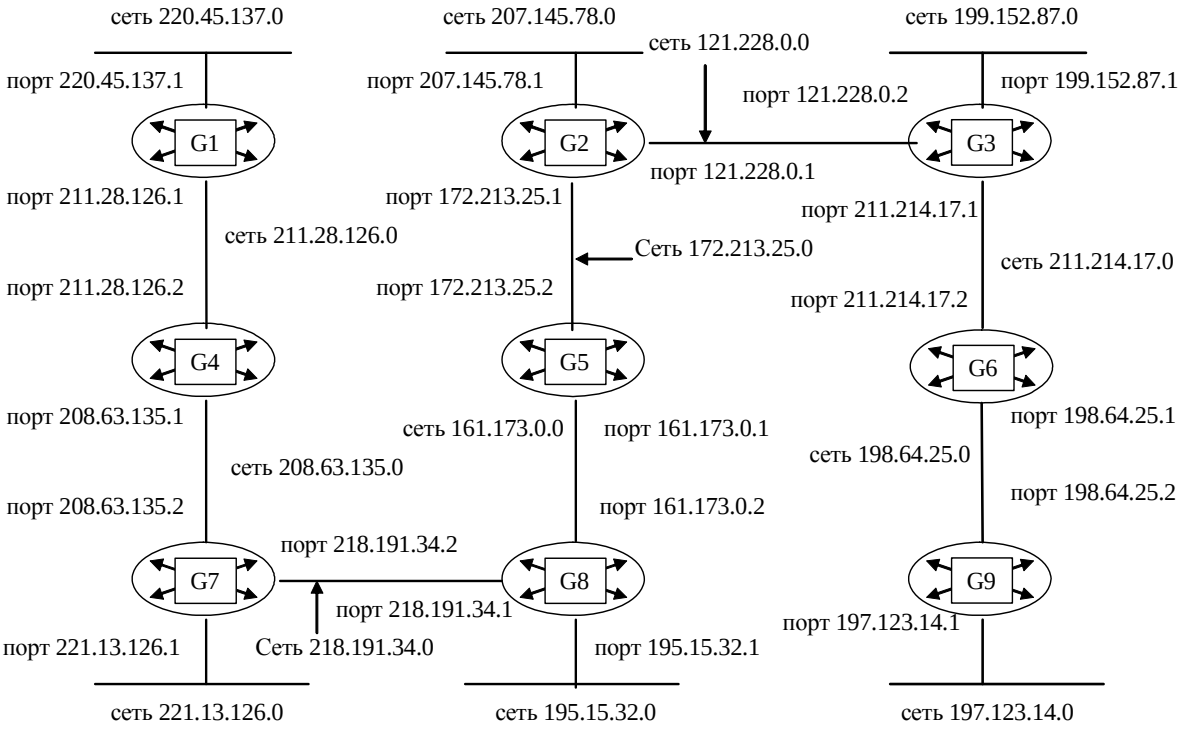
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x450000A4B132000017018B7A600720B360078D27.

По принятой информации определить протокол верхнего уровня, использующий данный пакет; а также указать класс адресов сети, в которой расположен узел источника, и класс адресов сети, в которой расположен узел приемника.

### *Задание 2*

Сеть некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G6, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G6;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 139.41.0.0 | 208.75.148.1 | 208.75.148.2 | 1 |
| 193.167.45.0 | – | 193.167.45.2 | 0 (подсоединена) |
| 196.127.35.0 | 208.75.148.1 | 208.75.148.2 | 1 |
| 208.75.148.0 | – | 208.75.148.2 | 0 (подсоединена) |
| 212.145.37.0 | 208.75.148.1 | 208.75.148.2 | 2 |
| 217.147.56.0 | – | 217.147.56.2 | 0 (подсоединена) |

**Вариант № 16**

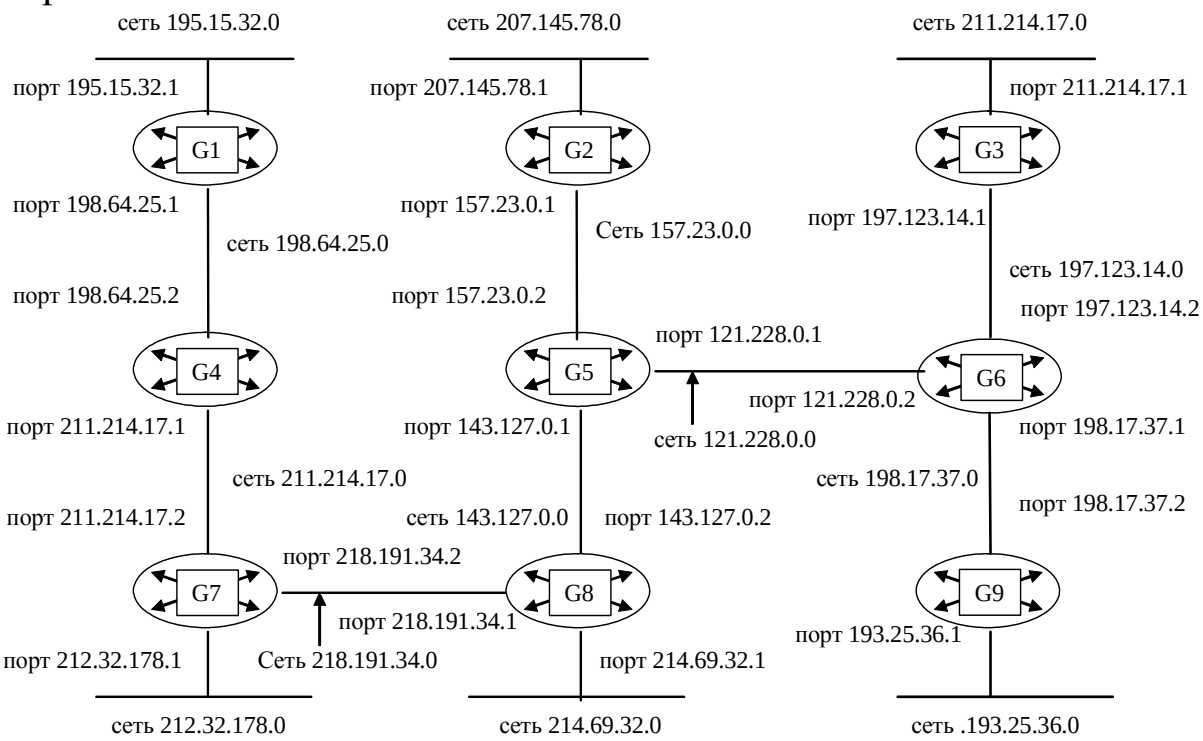
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500002320B3000067046007C1372B51C25D8B7A.

По принятой информации определить время жизни пакета и IP-адрес узла источника (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G7, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G7;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 177.184.0.0 | 198.146.17.1 | 198.146.17.2 | 1 |
| 194.113.56.0 | 204.196.78.1 | 204.196.78.2 | 1 |
| 198.146.17.0 | – | 198.146.17.2 | 0 (подсоединена) |
| 204.196.78.0 | – | 204.196.78.2 | 0 (подсоединена) |
| 217.37.154.0 | – | 217.37.154.2 | 0 (подсоединена) |
| 218.167.48.0 | 204.196.78.1 | 204.196.78.2 | 1 |

**Вариант № 17**

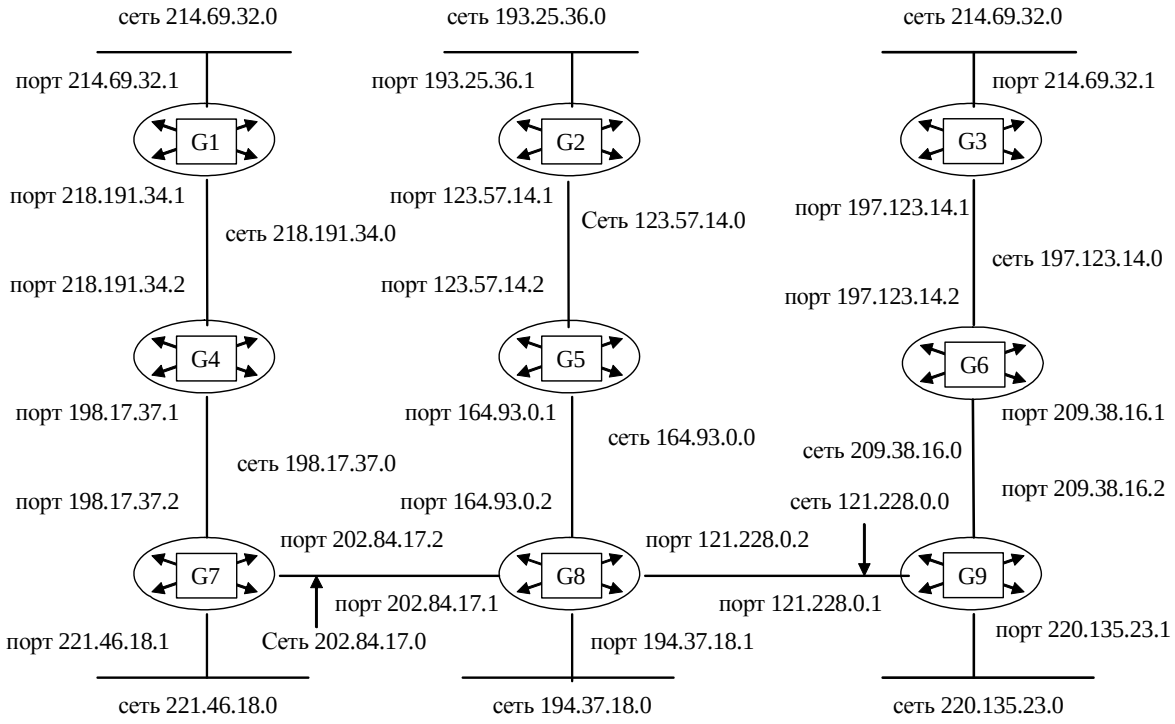
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x450001058B7A000089062B51AC137825A146C25D.

По принятой информации определить общую длину дейтаграммы и IP-адрес узла назначения (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G8, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G8;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 190.205.0.0 | 216.117.64.1 | 216.117.64.2 | 1 |
| 194.28.132.0 | – | 194.28.132.2 | 0 (подсоединена) |
| 196.178.37.0 | 216.117.64.1 | 216.117.64.2 | 1 |

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 203.145.13.0 | – | 203.145.13.2 | 0 (подсоединена) |
| 216.117.64.0 | – | 216.117.64.2 | 0 (подсоединена) |
| 217.154.58.0 | 216.117.64.1 | 216.117.64.2 | 2 |

**Вариант № 18**

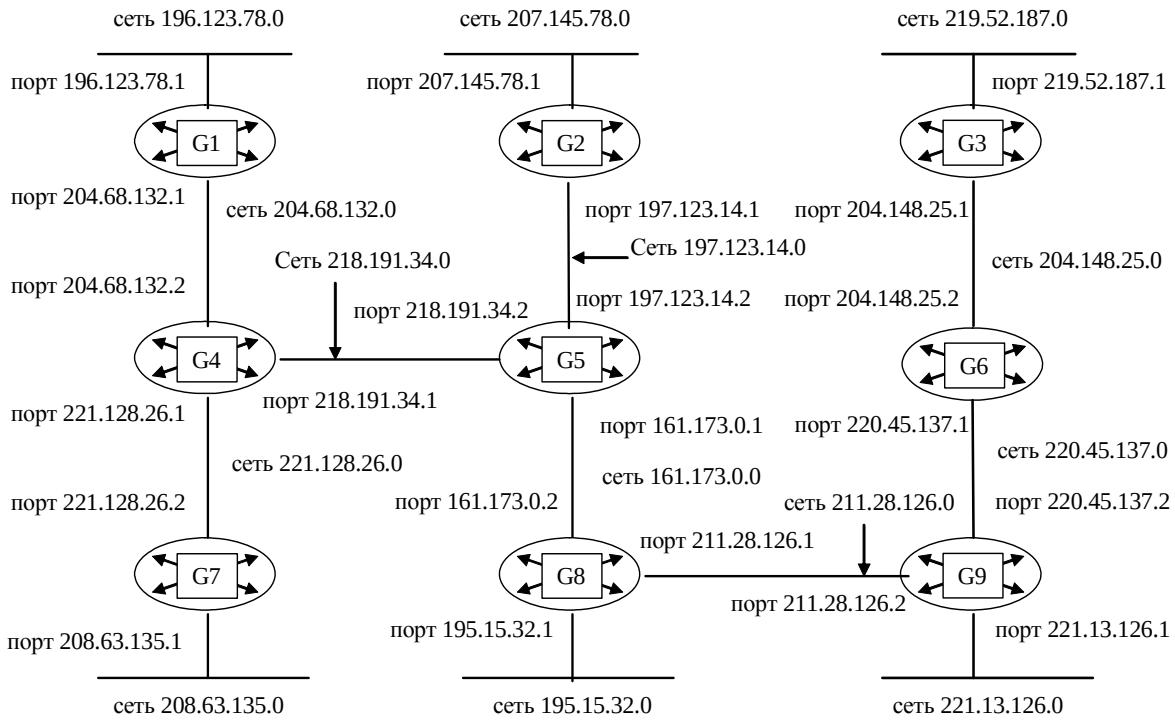
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x45000040782500008909C25D20B5AC1320A7F146.

По принятой информации определить протокол верхнего уровня, использующий данный пакет, а также указать класс адресов сети, в которой расположен узел источника, и класс адресов сети, в которой расположен узел приемника.

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G9, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G9;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 162.218.0.0 | 206.214.13.1 | 206.214.13.2 | 1 |
| 195.49.163.0 | – | 195.49.163.2 | 0 (подсоединена) |
| 196.185.61.0 | 206.214.13.1 | 206.214.13.2 | 1 |
| 206.214.13.0 | – | 206.214.13.2 | 0 (подсоединена) |
| 214.31.167.0 | 206.214.13.1 | 206.214.13.2 | 2 |
| 217.59.123.0 | – | 217.59.123.2 | 0 (подсоединена) |

**Вариант № 19**

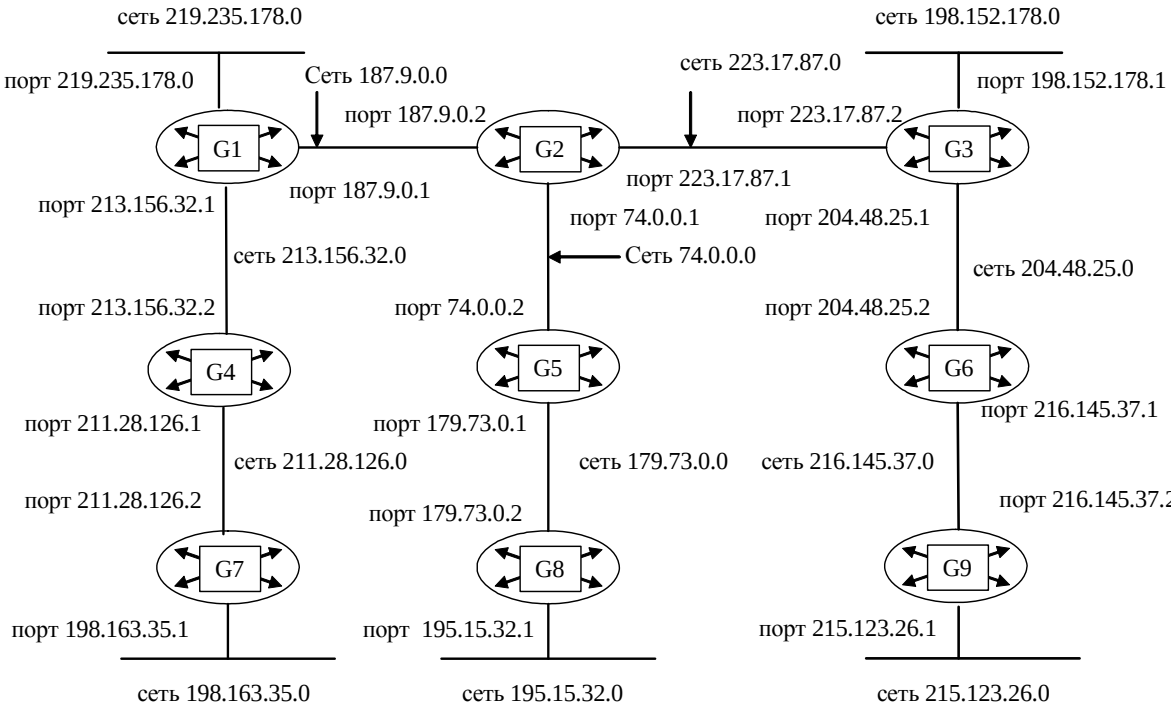
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x45000035AC1300003702F146C25D20B5C1A320A7.

По принятой информации определить время жизни пакета и IP-адрес узла источника (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G1, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G1;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 169.187.0.0 | 197.41.154.1 | 197.41.154.2 | 1 |
| 195.59.128.0 | 205.64.173.1 | 205.64.173.2 | 1 |
| 197.41.154.0 | – | 197.41.154.2 | 0 (подсоединена) |

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 205.64.173.0 | – | 205.64.173.2 | 0 (подсоединена) |
| 216.135.49.0 | 205.64.173.1 | 205.64.173.2 | 1 |
| 218.194.49.0 | – | 218.194.49.2 | 0 (подсоединена) |

**Вариант № 20**

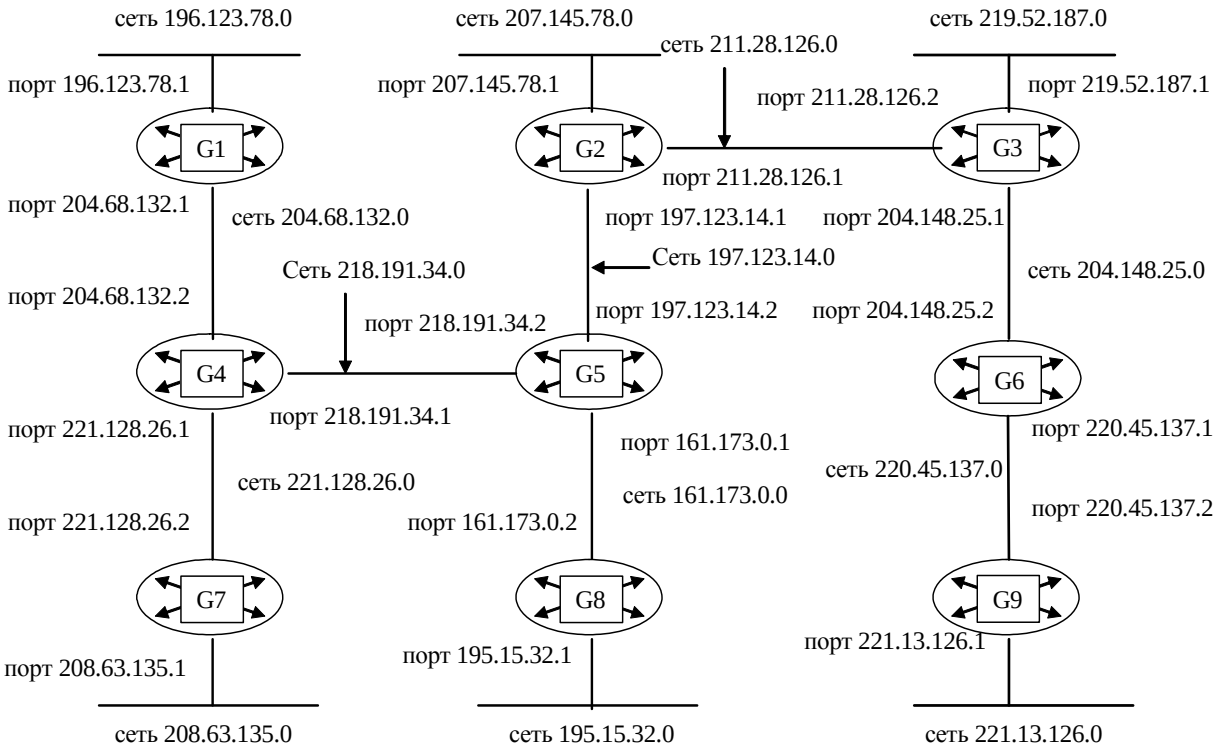
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500003020A700004506C25DB147B520B2C1A315.

По принятой информации определить общую длину дейтаграммы и IP-адрес узла назначения (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G2, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G2;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 134.243.0.0 | 196.51.129.1 | 196.51.129.2 | 1 |
| 196.51.129.0 | – | 196.51.129.2 | 0 (подсоединена) |
| 196.215.61.0 | 196.51.129.1 | 196.51.129.2 | 1 |
| 207.237.98.0 | – | 207.237.98.2 | 0 (подсоединена) |
| 212.52.216.0 | 196.51.129.1 | 196.51.129.2 | 2 |
| 214.36.149.0 | – | 214.36.149.2 | 0 (подсоединена) |

**Вариант № 21**

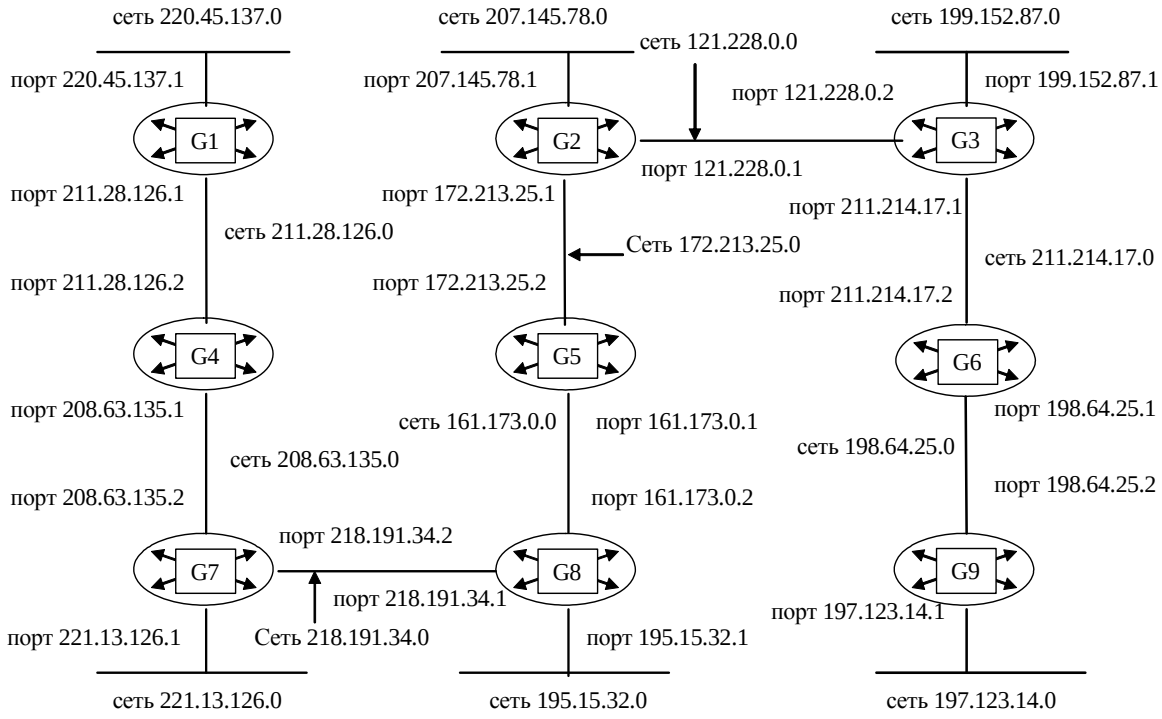
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x45000103116A000043111256C24A7C32C32B5D13.

По принятой информации определить протокол верхнего уровня, использующий данный пакет, а также указать класс адресов сети, в которой расположен узел источника, и класс адресов сети, в которой расположен узел приемника.

### *Задание 2*

Сеть некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G3, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G3;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 139.6.0.0 | 191.132.14.1 | 191.132.14.2 | 1 |
| 191.132.14.0 | – | 191.132.14.2 | 0 (подсоединена) |
| 196.192.98.0 | – | 196.192.98.2 | 0 (подсоединена) |
| 198.152.0.0 | 191.132.14.1 | 191.132.14.2 | 2 |
| 209.175.36.0 | – | 209.175.36.2 | 0 (подсоединена) |
| 214.198.26.0 | 191.132.14.1 | 191.132.14.2 | 1 |

**Вариант № 22**

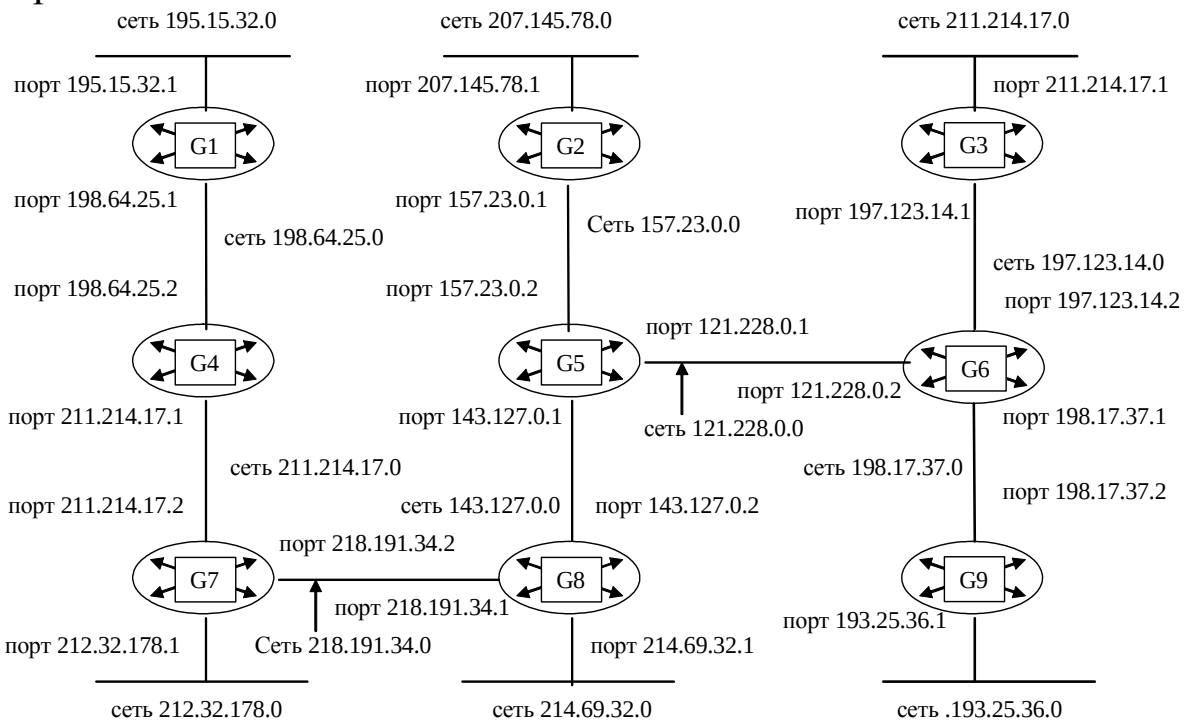
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x450000342356000013082 56AA0357C32A17B5D14.

По принятой информации определить время жизни пакета и IP-адрес узла источника (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G4, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G4;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 19.0.0.0 | 197.49.23.1 | 197.49.23.2 | 1 |
| 194.89.37.0 | – | 194.89.37.2 | 0 (подсоединена) |
| 195.12.14.0 | 197.49.23.1 | 197.49.23.2 | 2 |
| 196.134.0.0 | – | 196.134.0.2 | 0 (подсоединена) |
| 197.49.23.0 | – | 197.49.23.2 | 0 (подсоединена) |
| 212.245.55.0 | 197.49.23.1 | 197.49.23.2 | 1 |

**Вариант № 23**

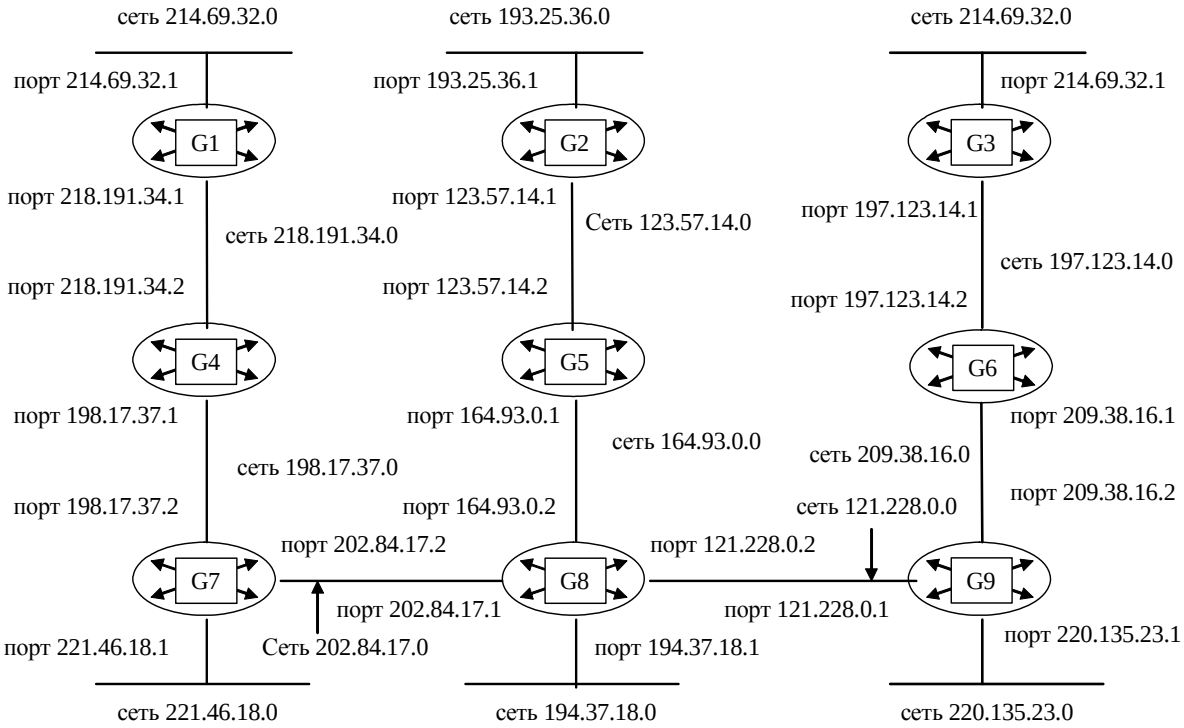
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500002134020000220B3A45126F037D127B8D52.

По принятой информации определить общую длину дейтаграммы и IP-адрес узла назначения (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G5, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G5;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 164.108.0.0 | – | 164.108.0.2 | 0 (подсоединена) |
| 196.213.48.0 | 204.89.167.1 | 204.89.167.2 | 1 |
| 199.87.152.0 | – | 199.87.152.2 | 0 (подсоединена) |
| 201.44.37.0 | 204.89.167.1 | 204.89.167.2 | 2 |

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 204.89.167.0 | – | 204.89.167.2 | 0 (подсоединена) |
| 219.168.54.0 | 204.89.167.1 | 204.89.167.2 | 1 |

**Вариант № 24**

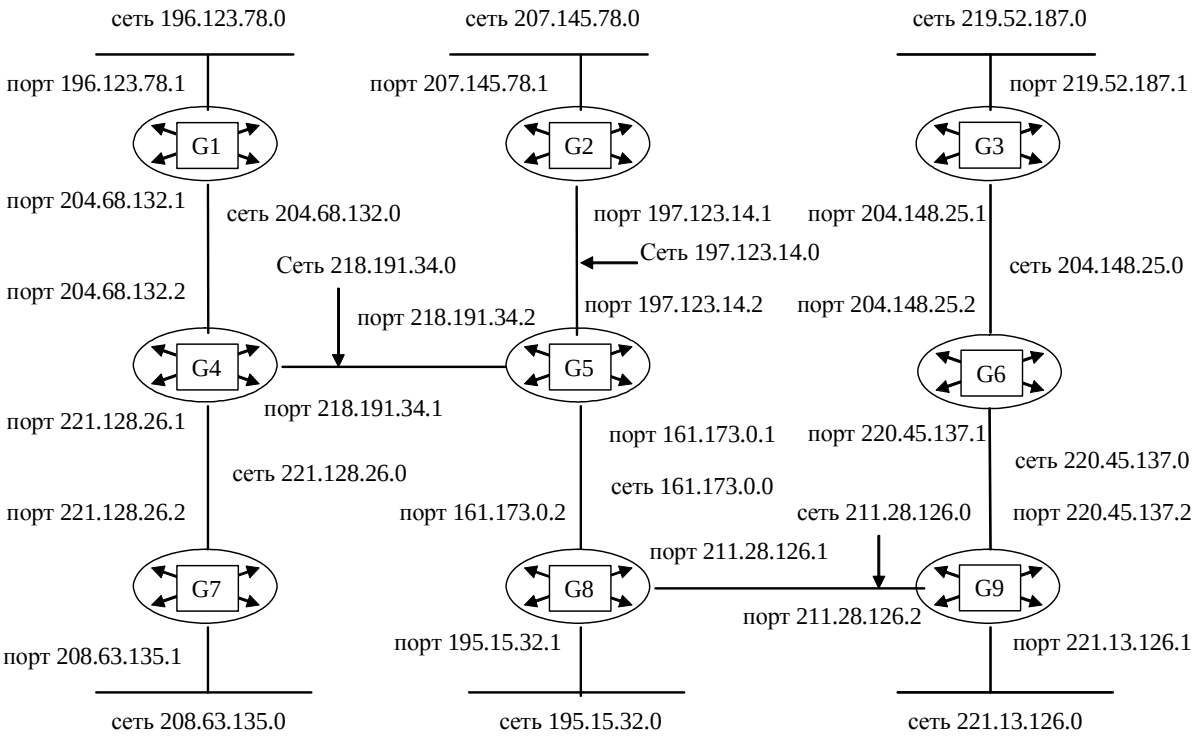
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500004478D200000D1120B3B1326F07B27C8B7D.

По принятой информации определить протокол верхнего уровня, использующий данный пакет; а также указать класс адресов сети, в которой расположен узел источника, и класс адресов сети, в которой расположен узел приемника.

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G6, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G6;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 172.132.0.0 | – | 172.132.0.2 | 0 (подсоединена) |
| 197.163.47.0 | – | 197.163.47.2 | 0 (подсоединена) |
| 198.145.17.0 | 172.132.0.1 | 172.132.0.2 | 1 |
| 209.132.95.0 | 197.163.47.1 | 197.163.47.2 | 1 |
| 214.157.13.0 | 197.163.47.1 | 197.163.47.2 | 1 |
| 217.136.47.0 | – | 217.136.47.2 | 0 (подсоединена) |

**Вариант № 25**

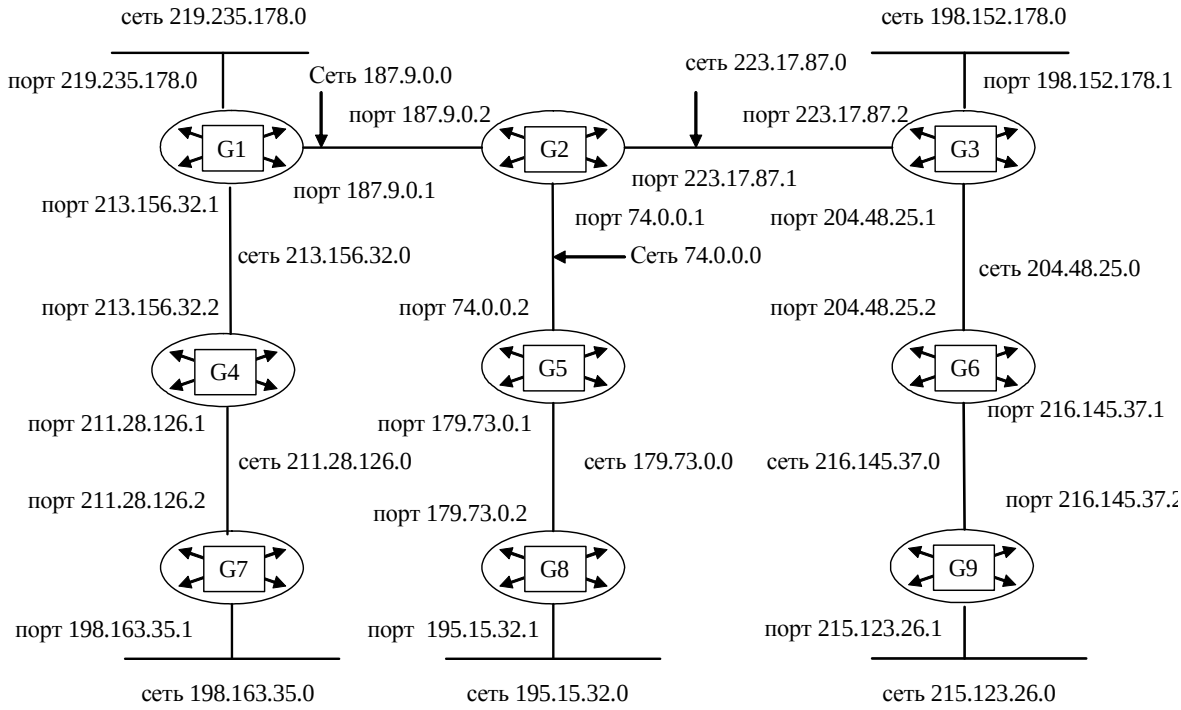
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x450000A4B132000017018B7A600720B360078D27.

По принятой информации определить параметр «Время жизни пакета» и IP-адрес узла источника (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G7, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G7;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следу-ющего маршрутизато-ра | Сетевой адрес выходного пор-та | Расстояние до сети назначения |
|---|---|---|---|
| 137.217.0.0 | – | 137.217.0.2 | 0 (подсоединена) |
| 195.164.17.0 | – | 195.164.17.2 | 0 (подсоединена) |
| 198.156.32.0 | 213.136.89.1 | 213.136.89.2 | 1 |

| Номер сети назначения | Сетевой адрес следу-ющего маршрутизато-ра | Сетевой адрес выходного пор-та | Расстояние до сети назначения |
|---|---|---|---|
| 203.52.126.0 | 195.164.17.1 | 195.164.17.2 | 1 |
| 212.203.78.0 | 195.164.17.1 | 195.164.17.2 | 1 |
| 213.136.89.0 | – | 213.136.89.2 | 0 (подсоединена) |

**Вариант № 26**

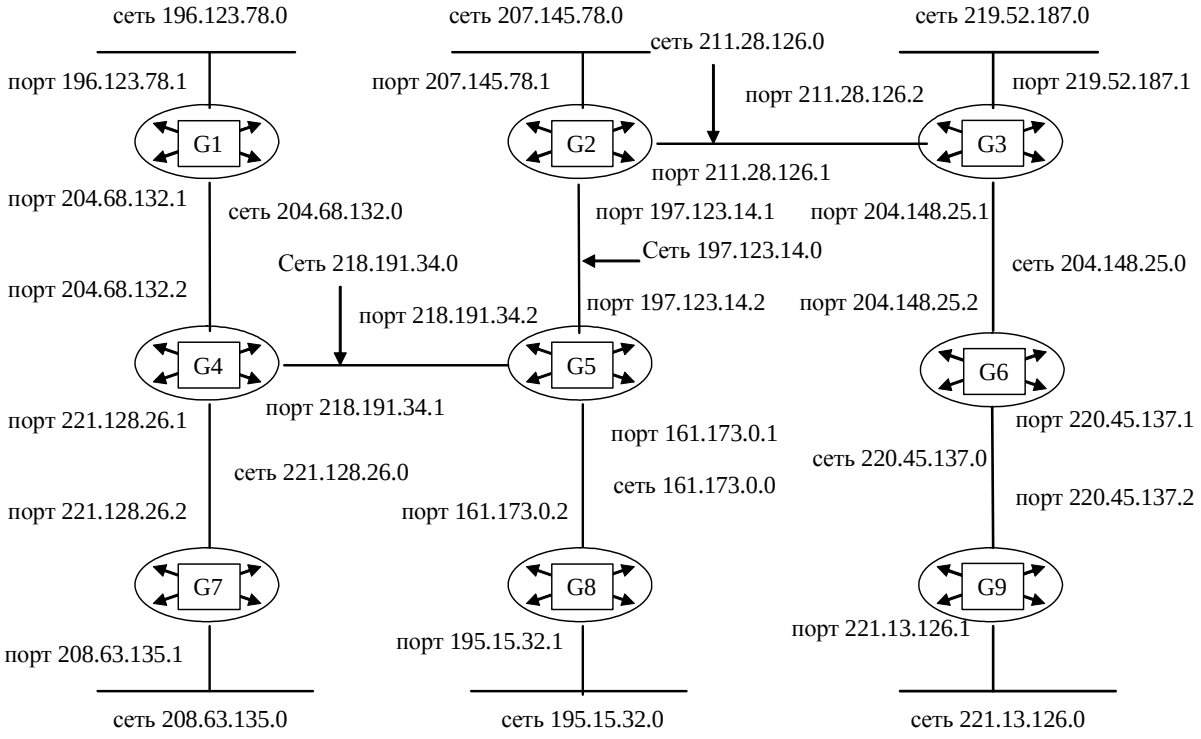
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500002320B3000067046007C1372B51C25D8B7A.

По принятой информации определить общую длину дейтаграммы и IP-адрес узла назначения (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G8, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G8;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 146.163.0.0 | – | 146.163.0.2 | 0 (подсоединена) |
| 194.59.27.0 | 205.201.129.1 | 205.201.129.2 | 2 |
| 197.138.59.0 | – | 197.138.59.2 | 0 (подсоединена) |
| 203.48.135.0 | 205.201.129.1 | 205.201.129.2 | 1 |
| 205.201.129.0 | – | 205.201.129.2 | 0 (подсоединена) |
| 218.215.31.0 | 205.201.129.1 | 205.201.129.2 | 1 |

**Вариант № 27**

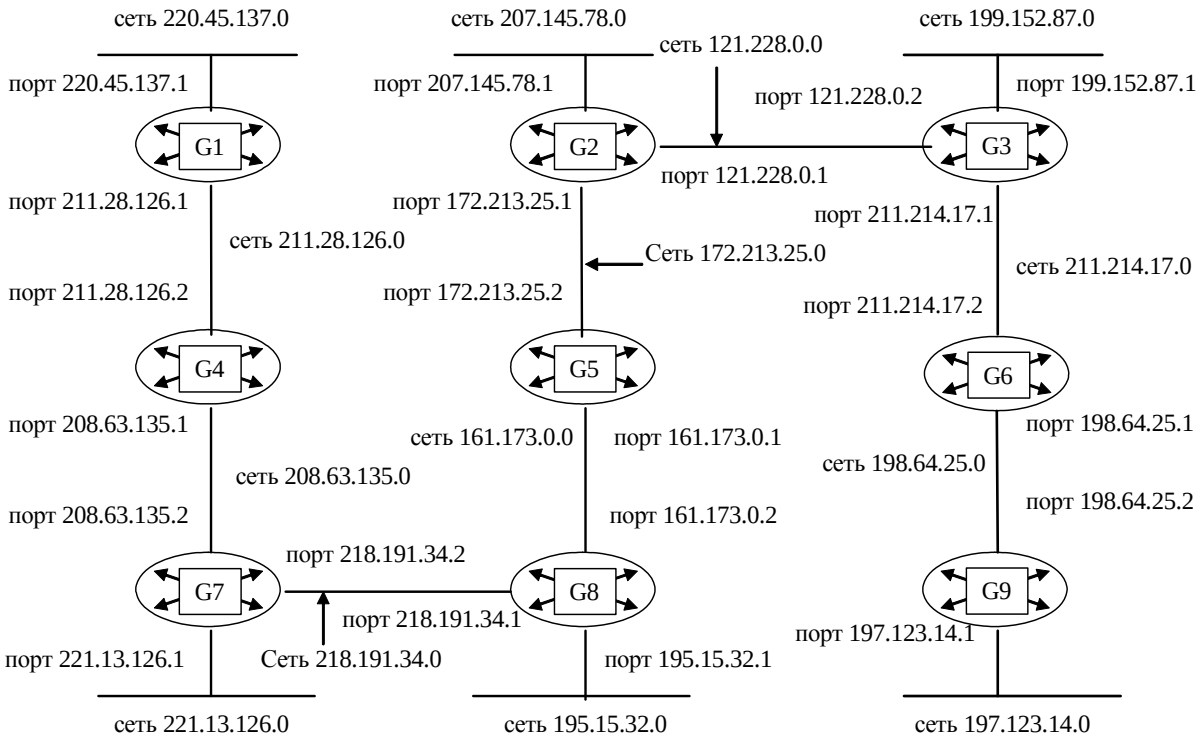
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x450001058B7A000089062B51AC137825A146C25D.

По принятой информации определить протокол верхнего уровня, использующий данный пакет, а также указать класс адресов сети, в которой расположен узел источника, и класс адресов сети, в которой расположен узел приемника.

### *Задание 2*

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G9, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G9;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 165.204.0.0 | – | 165.204.0.2 | 0 (подсоединена) |
| 193.48.97.0 | 219.42.153.1 | 219.42.153.2 | 1 |
| 194.76.187.0 | – | 194.76.187.2 | 0 (подсоединена) |
| 200.137.94.0 | 219.42.153.1 | 219.42.153.2 | 2 |
| 212.134.65.0 | 219.42.153.1 | 219.42.153.2 | 1 |
| 219.42.153.0 | – | 219.42.153.2 | 0 (подсоединена) |

## Вариант № 28

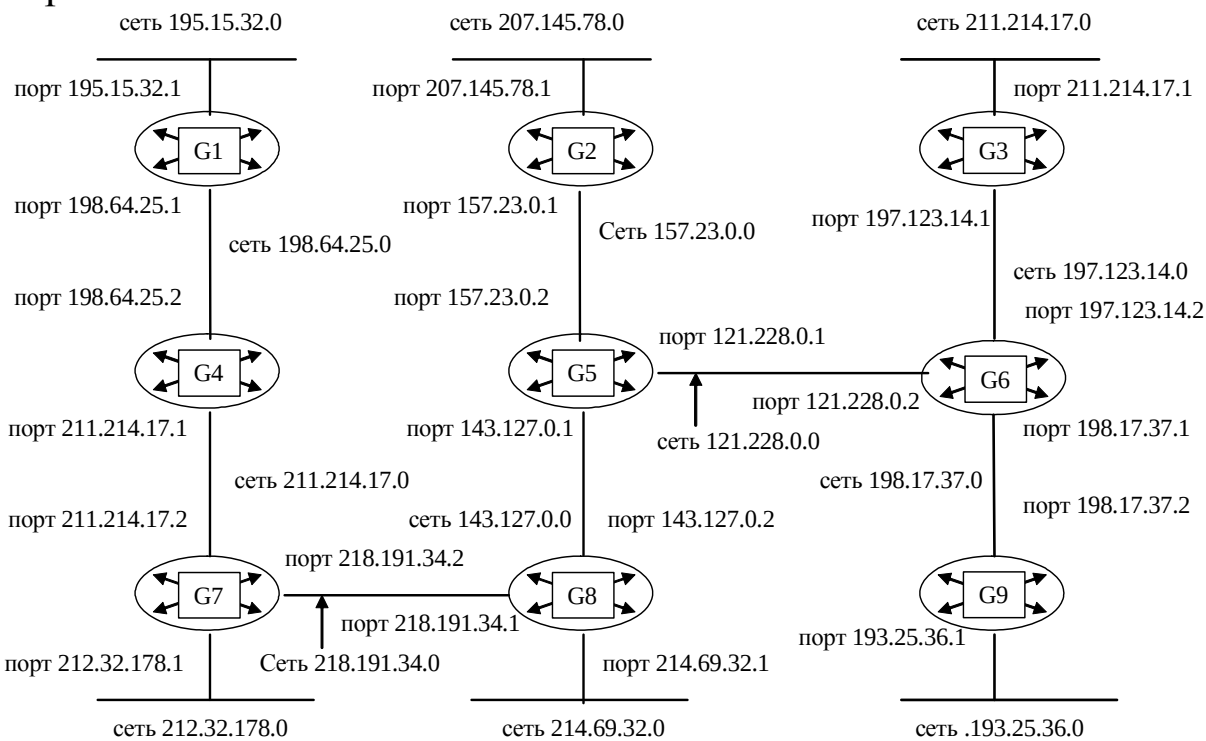
### *Задание 1*

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500004078250000 8909C25D20B5AC1320A7F146.

По принятой информации определить время жизни пакета и IP-адрес узла источника (в десятичной нотации).

### *Задание 2*

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G1, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G1;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 132.194.0.0 | – | 132.194.0.2 | 0 (подсоединена) |
| 195.164.138.0 | 196.47.135.1 | 196.47.135.2 | 1 |
| 196.47.135.0 | – | 196.47.135.2 | 0 (подсоединена) |
| 208.132.64.0 | 196.47.135.1 | 196.47.135.2 | 2 |
| 213.164.127.0 | 196.47.135.1 | 196.47.135.2 | 1 |
| 221.143.79.0 | – | 221.143.79.2 | 0 (подсоединена) |

**Вариант № 29**

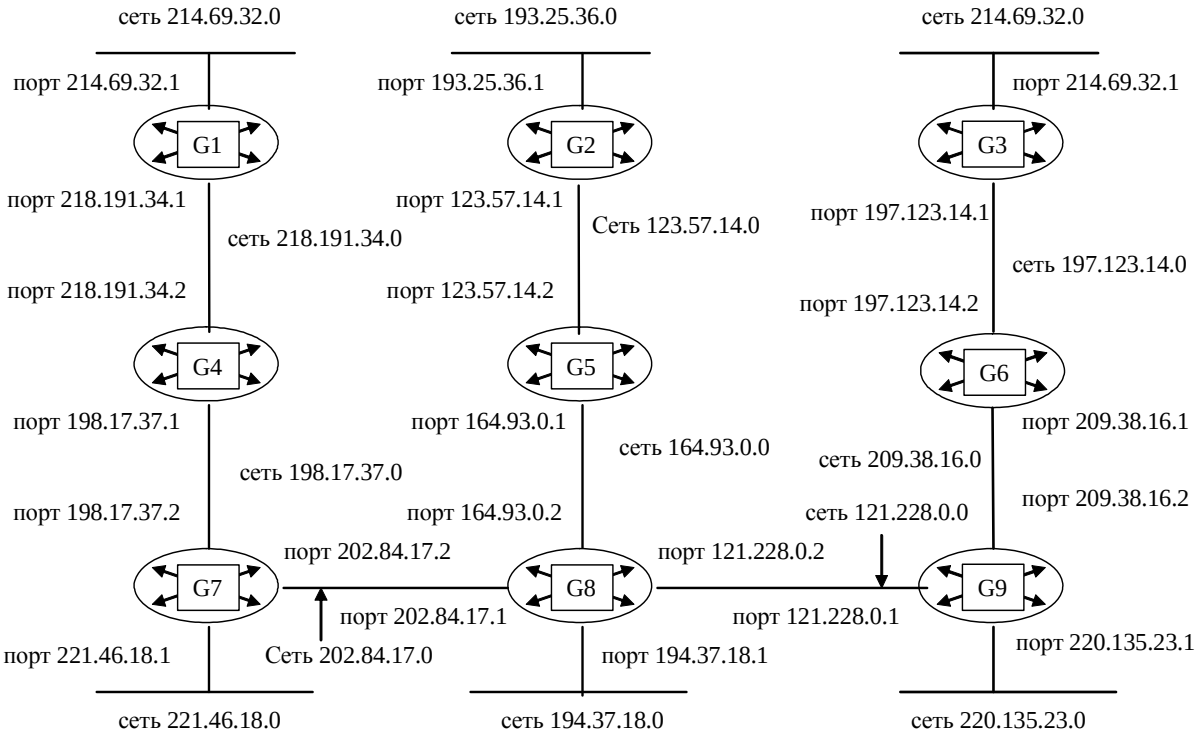
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x45000035AC1300003702F146C25D20B5C1A320A7.

По принятой информации определить общую длину дейтаграммы и IP-адрес узла назначения (в десятичной нотации).

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:




Составьте таблицу маршрутизации для маршрутизатора G2, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G2;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

### *Задание 3*

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 144.216.0.0 | – | 144.216.0.2 | 0 (подсоединена) |
| 195.84.132.0 | – | 195.84.132.2 | 0 (подсоединена) |
| 197.164.38.0 | 213.89.165.1 | 213.89.165.2 | 1 |

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 201.48.137.0 | 195.84.132.1 | 195.84.132.2 | 1 |
| 213.89.165.0 | – | 213.89.165.2 | 0 (подсоединена) |
| 217.68.213.0 | 195.84.132.1 | 195.84.132.2 | 1 |

**Вариант № 30**

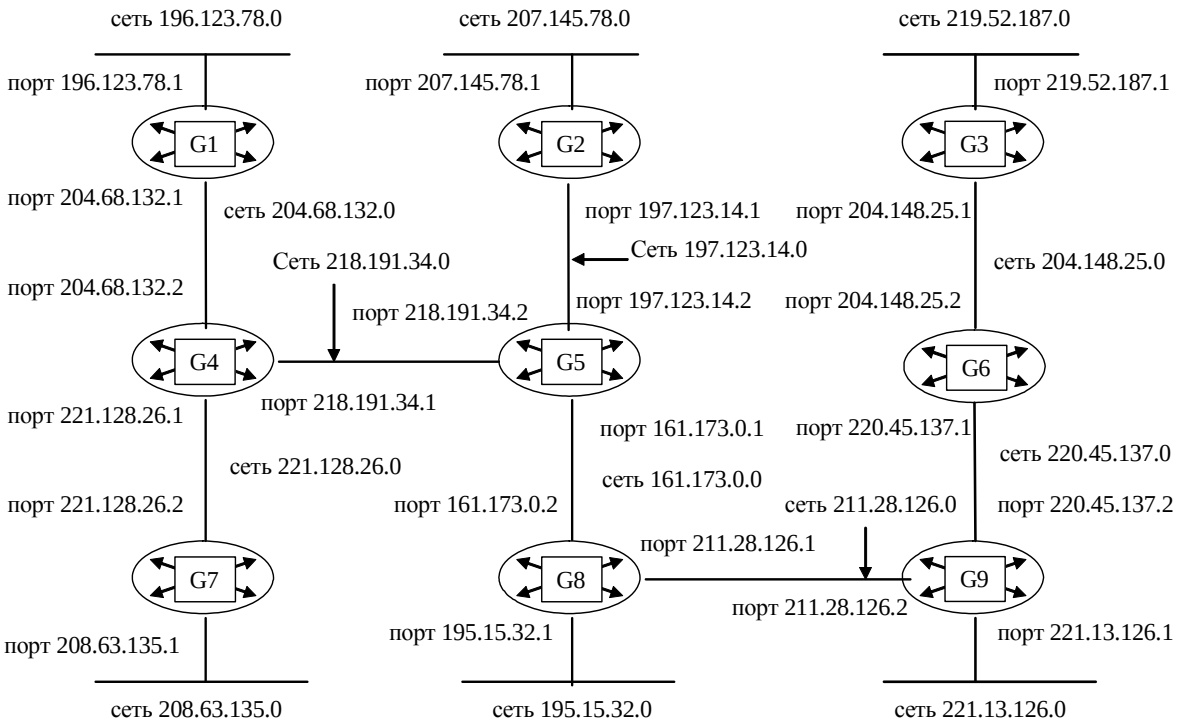
***Задание 1***

Заголовок IP-пакета представлен шестнадцатеричным кодом 0x4500003020A700004506C25DB147B520B2C1A315.

По принятой информации определить протокол верхнего уровня, использующий данный пакет, а также указать класс адресов сети, в которой расположен узел источника, и класс адресов сети, в которой расположен узел приемника.

***Задание 2***

Схема сети некоторой организации выглядит следующим образом:



Составьте таблицу маршрутизации для маршрутизатора G3, в которой укажите:

– адреса всех сетей, входящих в составную сеть;

– сетевой адрес следующего маршрутизатора, на который необходимо переслать пакет;

– сетевой адрес выходного порта маршрутизатора G3;

– расстояние до сети назначения (критерий выбора маршрута — количество пройденных в маршруте промежуточных маршрутизаторов).

***Задание 3***

Для структуризации составной сети используется 3 маршрутизатора. Составьте схему этой сети, если таблица маршрутизации одного из маршрутизаторов содержит следующие записи:

| Номер сети назначения | Сетевой адрес следующего маршрутизатора | Сетевой адрес выходного порта | Расстояние до сети назначения |
|---|---|---|---|
| 157.186.0.0 | – | 157.186.0.2 | 0 (подсоединена) |
| 193.178.93.0 | 217.154.67.1 | 217.154.67.2 | 1 |
| 195.78.141.0 | – | 195.78.141.2 | 0 (подсоединена) |
| 202.154.73.0 | 217.154.67.1 | 217.154.67.2 | 1 |
| 215.74.146.0 | 195.78.141.1 | 195.78.141.2 | 1 |
| 217.154.67.0 | – | 217.154.67.2 | 0 (подсоединена) |

# СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Олифер В.Г., Олифер Н.А. Компьютерные сети: принципы, технологии, протоколы: Учебник. — СПб.: Изд-во «Питер», 2002. — 672 с.: ил.

2. Таненбаум Э. Компьютерные сети. — СПб.: Изд-во «Питер», 2002. — 848 с.: ил.

3. Кульгин М.В. Коммутация и маршрутизация IP/IPX трафика. — М.: КомпьютерПресс, 1998. — 320 с.: ил.

4. Семенов Ю.А. Сети Интернет. Архитектура и протоколы. — М.: «Бликплюс», 1998. — 424 с.

5. Титтел Э., Хадсон К., Стюарт Дж.М. Networking Essentials. Сертификационный экзамен — экстерном (экзамен 70-058) — СПб.: ПитерКом, 1999. — 384 с.: ил.