Введение

Сегодня компьютерные сети (КС) широко используются для передачи различной информации, контроля и управления различными сервисами в реальном времени, просмотра телепередач, онлайн-покупок и т. д. В связи с увеличением новых классов телекоммуникационных устройств и соответствующих сервисов быстрыми темпами увеличиваются объемы информации, передаваемой через сеть Интернет (Интернет-трафик). Например, по данным Cisco Visual Networking Index (наглядные показатели Сети) объемы передаваемого Интернет-трафика увеличились со 100 Гбайт/сутки в 1992 г. до 16000 Гбайт/с в 2014 г. [84]*. При этом существенно усложнилась структура передаваемой информации, которая создается и используется многочисленными пользователями персональных компьютеров, смартфонов, планшетов, телевизоров, бытовой техникой (Интернет вещей) и др.

В этой ситуации закономерно возрастают требования к гибкости и масштабируемости современных КС, свойства которых оказываются существенно отличными от свойств КС с классической архитектурой (по сути статических). Например, традиционные архитектуры/дизайны КС оказываются неэффективными в динамических средах. При этом классические подходы, ориентированные на распределенное управление устройствами традиционных КС (например, виртуальные сети (VLAN)), не соответствуют современному уровню развития виртуализации серверов и систем хранения данных, а также требованиям крупного бизнеса и сервис-провайдеров (например, AT&T, Verizon, Google, Facebook, Microsoft и др.). Сложившаяся ситуация в телекоммуникационной отрасли подтверждается в том числе данными, содержащимися в аналитическом отчете за 2016 г. компании KPMG [10], где введены понятия «разрушающий трафик» (т. е. трафик таких объемов, с которыми не справляется используемое сетевое оборудование) и «разрушительные технологии», которые создают разрушающий трафик (виртуальная реальность, облачные сервисы, искусственный интеллект, анализ данных в реальном времени и др.).

---

* По последним данным Cisco в 2020 году объем Интернет-трафика составил примерно 40000 Гбайт/с. — *Прим. ред.*

Для эффективного решения проблем «разрушающего трафика», вопросов проектирования оборудования отвечающего потребностям современных КС, а также проектирования КС нового поколения, в том числе и виртуальных программно-конфигурируемых сетей (SDN, Software-defined Networking), необходимо понимать особенности информационных потоков, предаваемых в современных КС, механизмы их взаимодействия друг с другом и влияния на загрузку канала.

Анализ состояния современной теории телетрафика показывает, что имеет место определенный разрыв между современным уровнем развития телекоммуникационных технологий и математическими описаниями информационных процессов в КС, который пытаются восполнить большим количеством результатов проведенных экспериментальных исследований особенностей информационных потоков в КС, в особенности высокоскоростных магистральных Интернет-каналах (см. работы О.И. Шелухина [100, 101], В.В. Петрова [75, 76], Н.Г. Треногина [96], Е.В. Никульчева [105], M. Soysal, K. Fukuda, W. Leland, W. Willinger, D. Wilson и др.)

Однако объективный анализ этих работ показал, что проводимые экспериментальные исследования информационных потоков в магистральных Интернет-каналах зачастую имеют бессистемный характер. Это проявляется в отсутствии общепринятой методики исследований Интернет-трафика, а также в их направленности не на проверку, а на подтверждение тех или иных популярных математических моделей Интернет-трафика (в первую очередь, самоподобных). Кроме того:

- при проведении исследований используются устаревшие дампы Интернет-трафика, полученные в 90-х годах XX в. [75], на основе анализа которых далее идентифицируются параметры моделей Интернет-трафика, уже утративших свою актуальность [69, 57] (в том числе: искусственный трафик, передаваемый в тех или иных модельных локальных вычислительных сетях (ЛВС) [64, 96, 101], трафик, синтезированный с помощью соответствующих программных инструментов, в которых реализованы классические математические модели: системы массового обслуживания (СМО) (SPSS [22]), on-off-модель [68], жидкостная модель (ЖМ) [60] и гибридная жидкостная модель (ГЖМ) [61], а также программные генераторы трафика (NS-2, NS-3 [41]), свойства которых, как очевидно, существенно отличаются от свойств реальных информационных процессов,

протекающих в корпоративных ЛВС или провайдерских магистралях);

- зачастую проводится раздельное изучение свойств информационных потоков, созданных различными типами источников Интернет-трафика, без учета воздействия данных потоков друг на друга (см., например, [58]), что не позволяет достичь полного представления о процессах, протекающих в КС;
- в большинстве случаев анализ собранной экспериментальной информации проводится в соответствии со следующей схемой: выбор, зачастую субъективный, той или иной известной математической модели Интернет-трафика и далее идентификация ее параметров без проверки адекватности выбранной математической модели изучаемым информационным процессам, что подтверждает анализ многочисленных работ, посвященных исследованию самоподобных свойств Интернет-трафиках, число которых, по нашим оценкам, сегодня уже превысило тысячу;
- не обсуждаются используемые технологии получения Интернет-трафика и инструменты его анализа;
- отсутствуют общепринятые методики и общедоступные инструменты для анализа сетевого трафика.

Отметим, что Интернет-трафик является сложным многомерным объектом, который можно изучать в различных измерениях (например, число и размеры передаваемых пакетов, объемы передаваемой информации и т. д.). В этой связи очевидно, что первым и неотъемлемым этапом количественного анализа данных дампов является этап семантического анализа (парсинг) дампов Интернет-трафика (pcap-файлов), на котором из данных файлов извлекается необходимая количественная информация, используемая далее для получения количественных оценок характеристик трафика.

Таким образом, экспериментальные исследования свойств информационных потоков в высокоскоростных магистральных Интернет-каналах на основе системного подхода являются актуальным.

Объектом проведенного исследования были выбраны информационные потоки в высокоскоростных магистральных Интернет-каналах, а предметом исследования — свойства информационных потоков в высокоскоростных магистральных Интернет-каналах.

Цель исследования состояла в разработке и применении математического и алгоритмического обеспечения для анализа характеристик информационных потоков в высокоскоростных магистральных Интернет-каналах.

Для достижения поставленной цели были поставлены и решены следующие основные задачи исследования:

- анализ методов исследования информационных потоков в КС с точки зрения их применимости для исследования трафика в высокоскоростных магистральных Интернет-каналах;
- разработка программного инструмента, обеспечивающего автоматическое извлечение информации из pcap-файлов в выбранном измерении;
- разработка методики анализа первичной информации, извлеченной в соответствующем измерении из pcap-файлов и обеспечивающей получение количественных характеристик информационных потоков, переданных в магистральном высокоскоростном Интернет-канале;
- изучение особенностей информационных потоков в магистральном Интернет-канале, создаваемых выбранными классами пользователей («Слоны», «Мулы», «Мыши»), и их взаимного влияния друг на друга.

К новым научным результатам, полученным в проведенном научном исследовании особенностей информационных потоков в магистральном Интернет-канале, следует отнести:

- результаты анализа известных подходов к изучению данного типа Интернет-трафика, свидетельствующие об их недостатках;
- методологию анализа характеристик информационных потоков в высокоскоростных магистральных Интернет-каналах, а также соответствующее математическое и алгоритмическое обеспечение и их программные реализации;
- экспериментальные результаты, подтверждающие взаимное влияние информационных потоков, создаваемых в магистральном Интернет-канале выбранными классами пользователей («Слоны», «Мулы», «Мыши»), и наличие связей между объемами информации, переданной в магистральном Интернет-канале каждым из выбранных классов пользователей;
- разработанный алгоритм управления загрузкой канала передачи информационных потоков, позволяющий за счет отслеживания глобального показателя Херста накопленных сумм случайных последовательностей объема переданной информации класса «Мыши» и соответствующего проактивного ограничения скоростей потоков класса «Слоны» и «Мыши» минимизировать количество сбросов скользящего окна протокола TCP-IP для каждого потока, обеспечивая тем самым использование пропускной способности канала, близкой к максимальной.

К результатам исследования, имеющим практическую значимость, следует отнести:

- созданный программно-аппаратный комплекс, обеспечивающий анализ дампов Интернет-трафика, размещенных в pcap-файлов, адаптированный для использования на суперкомпьютере «Уран» Института математики и механики им. академика Н.Н. Красовского УрО РАН;
- разработанную методику анализа Интернет-трафика и созданный программно-аппаратный комплекс, работоспособность которых подтверждена результатами анализа дампов Интернет-трафика, зарегистрированного в магистральном Интернет-канале, проложенным между США и Японией, результаты которого подтвердили ее работоспособность;
- предложенный механизм балансировки объемов передаваемой информации каждым из выделенных классов пользователей («Слоны», «Мулы», «Мыши»), устанавливающий скорость передачи информации для каждого класса пользователей, исходя из значений показателей Херста накопленных сумм зависимостей мгновенного числа переданных пакетов, мгновенного объема переданной информации, мгновенного объема информации, переданной одним пакетом, от времени. (Здесь и далее под мгновенными значениями понимаются значения соответствующих параметров, подсчитанные в течение некоторого конечного временного интервала.)

Достоверность полученных результатов научных положений и выводов, изложенных в монографии, подтверждается использованием адекватных методов анализа первичной информации и выбранных количественных показателей процесса передачи данных в высокоскоростных магистральных Интернет-каналах и согласованностью полученных результатов с моделью OSI (Open Systems Interconnection — Модель взаимодействия открытых систем), технологией Ethernet (технология организации сетей), на базе которых построено большинство современных сетей, а также с результатами математического моделирования фрактального броуновского движения.

1 Анализ состояния предметной области. Постановка задач исследования

Большинство современных КС можно условно отнести к одному из следующих классов (рис. 1.1):

- локальные вычислительные сети коммерческих и образовательных учреждений;

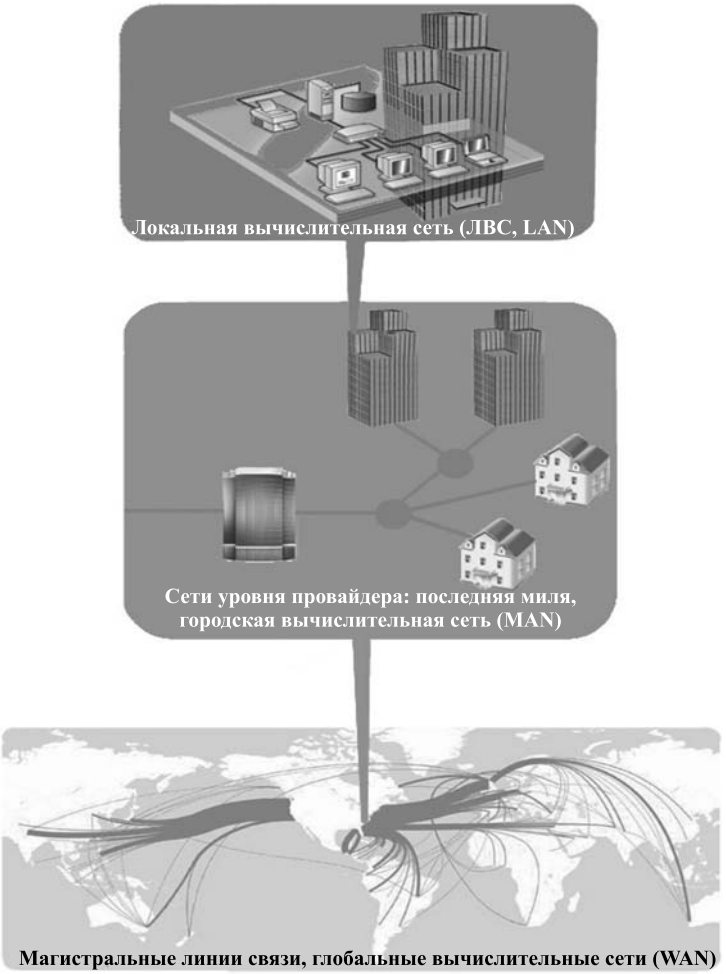

Рис. 1.1. Основные классы КС

- сети уровня провайдера: последняя миля, городская сеть;
- магистральные каналы, глобальные вычислительные сети.

Однако также существует множество виртуальных, анонимных и других смешанных КС, которые могут быть одновременно отнесены к двум или более из перечисленных выше классов КС.

1.1. Модели и протоколы, регламентирующие передачу данных в компьютерных сетях

Для обеспечения взаимодействия КС различных уровней разработан и используется ряд коммуникационных моделей, обсуждаемых в настоящем разделе.

1.1.1. Эталонная модель OSI

Методологической основой сетевых телекоммуникаций является модель взаимодействия открытых систем (Open System Interconnection — OSI), разработанная Международной организацией по стандартизации (International Organization of Standardization — ISO) [18]. Модель OSI, разработанная в 1984 г., представляет набор стандартов, которые обеспечивают совместимость и эффективное взаимодействие различных сетевых технологий и сетевого оборудования. В модели OSI выделены семь уровней процессов передачи информации (табл. 1.1).

Из табл. 1.1 видна основная особенность описания информационных потоков в КС — на разных уровнях модели OSI используются различные единицы измерения информации. Соответственно, при описании процессов передачи информации на каждом из

Таблица 1.1

Уровни модели OSI и их назначение

Уровень (layer)	Назначение	Единица информации
7. Прикладной (application)	Доступ к сетевым службам	Объем данных
6. Представления (presentation)	Представление и шифрование данных	Поток
5. Сеансовый (session)	Управление сеансом связи	Сеансы
4. Транспортный (transport)	Прямая связь между конечными пунктами и надежность	Сегменты / Дейтаграммы
3. Сетевой (network)	Определение маршрута и логическая адресация	Пакеты
2. Канальный (data link)	Физическая адресация	Кадры
7. Физический (physical)	Работа со средой передачи, сигналами и двоичными данными	Биты

уровней рассмотрения информационных потоков в КС используются собственные физические механизмы их описания. Отмеченная особенность описания процессов передачи информации в КС свидетельствует о том, что с научной точки зрения трафик в КС относится к категории «система», а его исследование к задачам системного анализа. Для обоснованного выбора уровня рассмотрения информационных потоков в высокоскоростных магистральных Интернет-каналах рассмотрим каждый из уровней модели OSI более подробно.

Уровень 7 (уровень приложений). Уровень приложений — наиболее близкий к пользователю уровень модели OSI. На данном уровне обслуживаются прикладные программы пользователей, находящихся вне пределов модели OSI, поэтому какие-либо услуги на данном уровне модели OSI не предоставляются. При этом, однако, идентифицируются источники и приемники передаваемой информации (партнеры), устанавливается их доступность для связи, синхронизируются совместно работающие прикладные программы, а также устанавливается договоренность о процедурах восстановления после обнаружения и устранения ошибок и контроля целостности данных. Также на данном уровне определяется степень достаточности ресурсов для осуществления предполагаемого обмена информацией между партнерами.

Уровень 6 (уровень представления). На уровне представления обеспечивается согласование синтаксиса передачи данных с уровнем приложений другой системы. При необходимости на данном уровне форматы данных приложений конвертируются в более общие форматы представления информации.

Уровень 5 (сеансовый). На сеансовом уровне обеспечивается установление, управление и завершение сеансов взаимодействия приложений. Сеансы состоят из диалога между двумя или более партнерами уровня представлений, в ходе которого реализуется управление обменом информации и обеспечивается синхронизация диалога между партнерами. На этом уровне также формируются отчеты об особых ситуациях, возникающих на сеансовом уровне, а также на уровнях приложений и представлений.

Уровень 4 (транспортный). На транспортном уровне данные сегментируются и повторно собираются в единый поток. Здесь решаются вопросы надежной транспортировки данных в КС со сложной топологией. На транспортном уровне функционируют механизмы установки, поддержания и упорядоченного завершения действия виртуальных каналов, обнаружения и устранения неисправностей транспортировки, а также управления информационным потоком с

целью предотвращения перегрузи одной системы данными, передаваемыми другой системой.

Уровень 3 (сетевой). На сетевом уровне функционируют механизмы, обеспечивающие выбор маршрута между пользователями, которые могут находиться в разных подсетях КС, и их соединение.

Уровень 2 (канальный). На канальном уровне решаются вопросы физической адресации, топологии сети, дисциплины в канале связи, уведомления об ошибках, упорядоченной доставки кадров, а также вопросы управления потоком данных. Этим обеспечивается надежный транзит данных через физический канал.

Уровень 1 (физический). На физическом уровне процесс передачи информации описывается в терминах теории электрических цепей (уровни напряжений, временные параметры изменения напряжений, скорости физической передачи данных, максимальные расстояния передачи информации, физические разъемы и т. д.). На физическом уровне задаются электротехнические, механические, процедурные и функциональные характеристики активизации, поддержания и деактивации физического канала между конечными системами.

Отметим, что на практике при выборе архитектуры сети, сетевого оборудования, его настройке и модернизации используется такая информация, как физическая адресация, маршруты, проверка доставки и др. [17], которая в соответствии с моделью OSI относится к канальному и сетевому уровням.

Одним из основных протоколов, регламентирующим правила передачи информации в КС, является стек протоколов TCP/IP. На практике производители аппаратного и программного обеспечения для описания и моделирования КС используют как уровни модели OSI, так и стек протоколов TCP/IP. В этой связи целесообразно рассмотреть, как соотносятся уровни рассмотрения информационных потоков в соответствии с моделью OSI и стеком протоколов TCP/IP.

1.1.2. Стек протоколов TCP/IP

Стек протоколов TCP/IP регламентирует процесс передачи информации между конечными точками. В нем определены формат данных, правила адресации, маршрутизации и процесса обмена. Стек TCP/IP включает в себя множество коммуникационных протоколов, важнейшими из которых являются TCP и IP. Функции стека протокола TCP/IP, как и в эталонной модели OSI, разделены на несколько уровней (рис. 1.2).

К канальному уровню протокола TCP/IP отнесены процессы, которые реализуются на двух нижних уровнях модели OSI — ка-

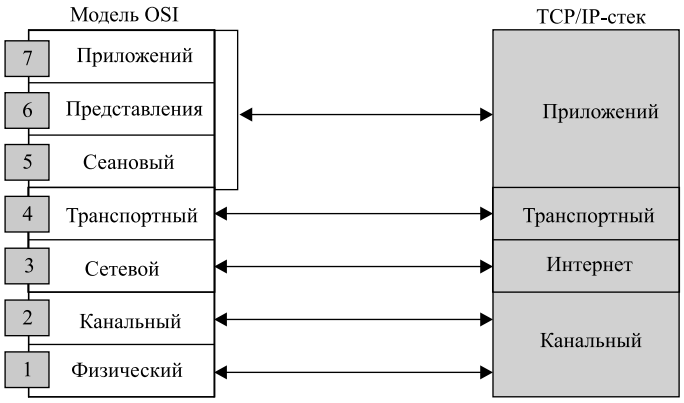

Рис. 1.2. Соответствие между уровнями модели OSI и стеком TCP/IP

нальном и физическом. На канальном уровне протокола TCP/IP задаются физические характеристики соединения и контролируется доступ к передаваемым данным и формат их передачи.

На Интернет-уровне решается задача маршрутизации данных от источника до места назначения за счет идентификации пакетов, в том числе переданных удаленными хостами, а также перемещения данных между канальным и транспортным уровнями, фрагментации и сборки пакетов данных.

Ядром стека протоколов TCP/IP является транспортный уровень. На данном уровне предоставляются услуги связи прикладным процессам, которые запущены на сетевых хостах.

На прикладном уровне рассматриваются приложения, обеспечивающие передачу файлов, устранение неполадок КС и доступа пользователей к сети Интернет, а также реализуется поддержка интерфейса программирования приложений (Application Program Interface — API), который обеспечивает доступ к КС программам, разработанным для различных операционных систем.

С научной точки зрения основной интерес представляют сетевой и канальный уровень модели OSI и соответственно транспортный и Интернет-уровни TCP/IP-стека. На данных уровнях работают TCP-, UDP- и IP-протоколы, которые являются основой функционирования вычислительных сетей. Данные между уровнями TCP/IP-стека (так же как и между уровнями модели OSI) передаются методом инкапсуляции.

1.1.3. Инкапсуляция

Данные, передаваемые прикладными программами в соответствии с протоколом TCP, прежде чем они превратятся в поток би-

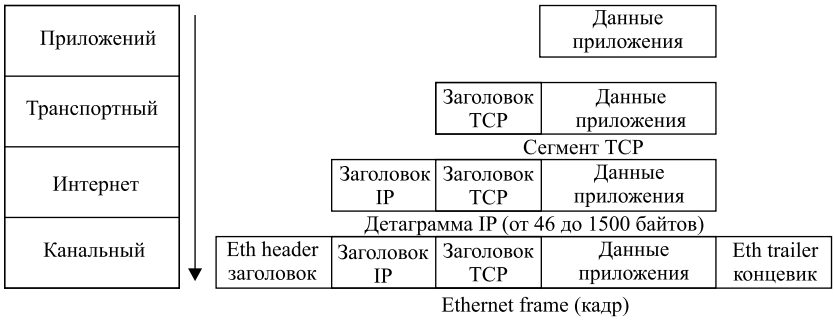

Рис. 1.3. Схема процесса инкапсуляции данных

тов, передаваемых в линию, проходят сверху вниз через все уровни протокольного стека (рис. 1.3).

Из рис. 1.3 видно, что на каждом слое к информации, полученной с верхнего уровня, добавляется дополнительная информация в виде заголовка (header), а на канальном уровне — дополнительного блока (концевик, trailer). Затем сегмент TCP (TCP segment), состоящий из заголовка TCP и данных приложения, передается IP-модулю. Порция данных, которую IP отдает драйверу интерфейса, называется IP-дейтаграммой (IP datagram). Пачка битов, передаваемых по кабелю Ethernet, образует кадр (frame).

Длина кадра в локальной сети Ethernet находится в диапазоне от 46 до 1500 байтов, что обусловлено физическими ограничениями на размер поля данных.

Передача данных в соответствии с протоколом UDP происходит аналогично. Здесь с транспортного уровня на Интернет-уровень передается UDP-дейтаграмма, состоящая из 8-байтного UDP-заголовка и данных приложения. Отметим, что подобным образом организована передача данных между рассматриваемыми уровнями и для других протоколов стека протоколов TCP/IP, например ICMP и IGMP. В этой связи в IP заголовке предусмотрено 8-битовое поле protocol, в котором указывается код протокола, по которому осуществлена передача данных с транспортного уровня на Интернет-уровень: ICMP — 1, IGMP — 2, TCP — 6, UDP — 17.

Аналогично, несколько различных приложений могут одновременно использовать протокол TCP (или UDP). В этой связи в TCP- и UDP-заголовках, добавляемых к данным на транспортном уровне, предусмотрены специальные поля (16-разрядный номер порта port numbers), в которые записывается условный код, позволяющий идентифицировать приложение-источник данных и приложение-получатель данных.

На канальном уровне к данным Интернет-уровня добавляется Ethernet-заголовок, в котором имеется 16-разрядное поле type, предназначенное для идентификации типа протокола, использованного для передачи данных между обсуждаемыми уровнями (IP, ARP или RARP).

Так как протоколы TCP, UDP и IP являются сегодня основными протоколами, в соответствии с которыми реализуется передачи данных в КС, далее эти протоколы рассматриваются более подробно.

1.1.4. Протокол TCP

Протокол TCP, являющийся надежной потоковой службой, соответствует транспортным уровням стека TCP/IP и эталонной модели OSI. По сути данный протокол — это независимый протокол общего назначения, который можно адаптировать для использования с любыми средствами доставки [4].

Единицей передачи данных между двумя хостами в протоколе TCP является сегмент. Сегменты обеспечивают установление соединений, передачу данных, отправку сигналов подтверждения приема, анонсирование размеров окон передачи данных и закрытие соединения. Поскольку протокол TCP является дуплексным (т. е. потоки данных могут одновременно передаваться в двух направлениях) сигналы подтверждения, посланные от хоста А к хосту Б, передаются в тех же сегментах, что и потоки данных от хоста А к хосту Б, несмотря на то что сигналы подтверждения приема относятся к потокам данных, текущих от хоста Б к хосту А. Формат TCP сегмента представлен на рис. 1.4.

Из рис. 1.4 видно, что каждый сегмент состоит из двух частей — заголовка и блока данных. В заголовке, называемом TCP-заголовком, находятся идентификационные данные и управляющая информация. В первых двух полях заголовка располагаются номера

<table>
<tr><td colspan="16">0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td><td colspan="16">16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31</td></tr>
<tr><td colspan="16">Номер порта отправителя</td><td colspan="16">Номер порта получателя</td></tr>
<tr><td colspan="32">Порядковый номер</td></tr>
<tr><td colspan="32">Номер сигнала подтверждения</td></tr>
<tr><td colspan="4">Длина заголовка</td><td colspan="6">Резерв</td><td colspan="6">Код сегмента</td><td colspan="16">Размер окна</td></tr>
<tr><td colspan="16">Контрольная сумма</td><td colspan="16">Указатель срочных данных</td></tr>
<tr><td colspan="24">Параметры протокола TCP (при наличии)</td><td colspan="8">Выравнивание</td></tr>
<tr><td colspan="32">Область данных</td></tr>
</table>

Рис. 1.4. Формат TCP-сегмента

TCP-портов отправителя и получателя, которые идентифицируют прикладные программы по обе стороны соединения. В поле «Порядковый номер» заносится текущее положение текущего сегмента в потоке данных отправителя. В поле «Номер сигнала подтверждения» указывается номер октета, который отправитель ожидает в дальнейшем получить обратно. Значение поля «Порядковый номер» относится к потоку данных, направленному в том же направлении, что и передаваемый сегмент, номер сигнала подтверждения — к потоку данных, направление которого противоположно направлению передаваемого сегмента.

В поле «Длина заголовка» указывается длина заголовка сегмента, выраженное в блоках длиной 32 бита. Данное поле включено в заголовок, поскольку поле параметров протокола имеет переменную длину, зависящую от того, какие параметры включены в сегмент. Таким образом, размер TCP-заголовка зависит от того, какие параметры в него включены. Поле «Резерв» (см. рис. 1.4) размером 6 битов зарезервировано для использования в будущих стандартах протокола.

Поскольку в протоколе TCP сегменты используются и для передачи данных, и для передачи сигналов подтверждения их приема, а также для передачи запросов на установку и закрытие соединения, в TCP-заголовок включено специальное 6-битовое поле «Код сегмента», которое определяет его формат и содержимое. Значения битов, записанных в данное поле, определяют правила интерпретации других полей TCP-заголовка (табл. 1.2).

В поле «Размер окна» отправляемого сегмента по протоколу TCP размещается количество октетов, которые получатель может принять (т. е. указывают размер своего приемного буфера). В это поле помещается 16-битовое беззнаковое целое число, определяющее стандартный сетевой порядок следования байтов. Анонсирование окна является примером дуплексной передачи данных, поскольку

Таблица 1.2

Значения битов кода сегмента TCP-заголовка

Название бита (слева направо)	Значение, если бит установлен в 1
URG	В заголовке присутствует указатель срочных данных
ACK	В заголовке указано поле подтверждения приема
PSH	В данном сегменте указан запрос на немедленную отправку данных (push)
RST	Сброс соединения
SYN	Синхронизация порядковых номеров
FIN	Отправитель достиг конца потока данных

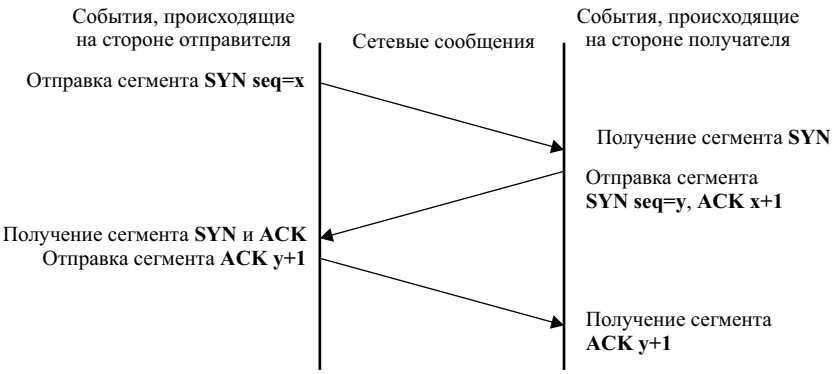

Рис. 1.5. Процесс квитирования протокола TCP

оно выполняется для всех сегментов, включая сегменты, в которых передаются данные, и сегменты с сигналами подтверждения приема.

Для установки соединения в протоколе TCP используется трехэтапный метод квитирования (three-way handshake). Пример простейшего процесса квитирование представлен на рис. 1.5.

Из рис. 1.5 видно, что в первом сегменте, посылаемым в процессе квитировании отправителем, в поле кода сегмента его заголовка установлен бит SYN. В заголовке второго сегмента, отправляемого получателем, установлены два бита: SYN и ACK. Для отправителя это означает, что с получателем успешно установлено соединение и он готов к получению данных. Далее отправитель передает получателю сегмент ACK, который подтверждает, что обе стороны уведомлены об установке соединения.

Трехэтапный метод квитирования выполняет две важные функции. Во-первых, он гарантирует, что обе стороны соединения готовы к приему данных и что о готовности одной стороны знает другая сторона. Во-вторых, с его помощью выполняется процесс согласования начального значения поля «Порядковый номер». Во время квитирования обе стороны соединения обмениваются начальными порядковыми номерами и ожидают подтверждения их приема. Порядковые номера используются для идентификации потоков данных, посылаемых каждой из сторон открытого соединения. Они выбираются сторонами самостоятельно (обычно случайным образом) во время открытия соединения, однако они никогда не могут начинаться с одного и того же значения. Это позволяет не допустить совпадения номеров октетов, указываемых в сигналах подтверждения приема, и номеров, используемых в заголовках при передаче сегментов данных.

Отметим, что в протоколе TCP невозможна передача данных вместе с начальным порядковым номером непосредственно в сегментах квитирования. При возникновении подобных случаев модуль протокола TCP блокирует данные до завершения процесса квитирования, а по его завершению доставляет ожидающему их приложению.

Метод скользящего окна. Метод скользящего окна в сравнении с описанным выше вариантом квитирования является более сложной технологией, в которой реализовано подтверждение приема с повторной передачей. Он позволяет эффективно использовать полосу пропускания — максимально возможную скорость передачи информации в КС, соединяющей компьютер пользователя с Интернет через оператора услуг связи (количество битов данных, которое можно передать по каналу в каждую секунду; Кбит/с (1024 бита в секунду), Мбит/с, Гбит/с) [8], поскольку отправитель может послать несколько пакетов сразу, не дожидаясь подтверждения приема каждого пакета. При использовании данного метода выбирается некоторое окно фиксированного размера, которое в процессе передачи сдвигается вдоль пронумерованной последовательности пакетов, предназначенных для передачи от источника к приемнику (рис. 1.6).

Рис. 1.6. К объяснению метода «скользящего окна» (размер окна — 8 пакетов)

Источник передает в КС пакеты № 1–8. После получения подтверждения о приеме пакета № 1 скользящее окно сдвигается на один пакет вправо и в сеть отправляется пакет № 9 (рис. 1.6,*б*). Если для какого-либо пакета, переданного при данном положении окна, не будет получен сигнал подтверждения его получения, то выполняется повторная передача данного пакета. Пакет считается непринятым (unacknowledged), если он был послан в КС, но подтверждение о его приеме не было получено. Формально количество непринятых пакетов не может превышать размеров окна и на практике оказывается относительно небольшим целым числом.

Иллюстрация метода скользящего окна на примере окна, состоящего из трех пакетов, представлена на рис. 1.7.

Из рис. 1.7 видно, что отправитель успевает послать все три пакета в КС до того, как будет получено хотя бы одно сообщение о подтверждении их приема. Таким образом, при правильном вы-