

Александр Ватаманюк

# БЕСПРОВОДНАЯ СЕТЬ

своими руками

Убедитесь сами:  
без проводов проще!

## Из книги вы узнаете:

- о стандартах беспроводной связи
- о юридических ограничениях
- как выбрать, установить и настроить оборудование
- как настроить сеть
- как сделать работу в сети безопасной

 ПИТЕР®

Александр Ватаманюк

**Беспроводная сеть своими руками**

«Питер»

## **Ватаманюк А. И.**

Беспроводная сеть своими руками / А. И. Ватаманюк —  
«Питер»,

Вы хотите создать домашнюю или офисную сеть? Тогда вы должны быть готовы к тому, что вам придется провести немало времени с дрелью и перфоратором, глотать пыль в подвале или испытывать страх высоты на крыше, «воевать» с ЖЭСом за протянутый между домами провод, ругаться с родителями из-за испорченного ремонта – в общем, всех проблем и не перечислишь. Однако неприятностей можно избежать, построив сеть на основании беспроводных технологий. Прочитав эту книгу, вы будете знать все необходимое для того, чтобы суметь самостоятельно создать беспроводную сеть: от теоретических основ беспроводной связи до принципов настройки сети в различных версиях Windows. Книга написана так, что будет понятна пользователю, никогда ранее не занимавшемуся сетями. В то же время отдельные ее разделы откроют немало нового даже опытным системным администраторам.

© Ватаманюк А. И.

© Питер

# Содержание

Введение	5
От издательства	6
Глава 1	7
1.1. Модель ISO/OSI	8
Физический уровень	8
Канальный уровень	9
Сетевой уровень	10
Транспортный уровень	11
Сеансовый уровень	11
Уровень представления	11
Прикладной уровень	11
1.2. Понятие протокола	12
1.3. NetBIOS и NetBEUI	13
NetBIOS	13
NetBEUI	13
1.4. TCP, IP и UDP	14
TCP	14
IP	14
UDP	14
1.5. IPX и SPX	16
IPX	16
SPX	16
1.6. SMTP, POP3 и IMAP	17
SMTP	17
POP3	17
IMAP	17
Конец ознакомительного фрагмента.	18

# Александр Ватаманюк

## Беспроводная сеть своими руками

### Введение

С недавних пор человек и компьютер стали просто неразлучны. Это произошло, когда в один прекрасный день человек осознал, что компьютер чрезвычайно глубоко проник в его жизнь и отказаться от него практически невозможно.

Сначала компьютеры были каждый сам по себе, затем они объединились в локальную сеть, потом – в глобальную. Трудно представить себе, что будет дальше: компьютер, имплантированный в голову человека, телепатическая связь, виртуальные встречи...

Объединение отдельно стоящих компьютеров в группы позволило достичь невиданных высот как в технологическом плане, так и в сознании человека. Сеть предоставляет пользователям огромное количество разнообразнейших ресурсов, возможность общения и отдыха, серфинг в Интернете, бесплатные звонки в другие страны, участие в торгах на биржах, возможность неплохо зарабатывать и т. д. Чтобы все это получить, нужно начать с малого – создать сеть.

Существуют разные типы и способы построения компьютерных сетей. Наиболее «мощные» технические возможности предоставляет, конечно, проводная сеть. Однако все перспективнее становятся сети, построенные с помощью радиотехнологий, позволяющих приобрести максимальную мобильность и независимость. Пока технологии беспроводных сетей не такие «продвинутые», как технологии их проводных собратьев, но они только продолжают развиваться, и в них скрыт огромный потенциал.

В данной книге вы познакомитесь как с теоретическими, так и с практическими основами функционирования сетей. Используя полученные знания, вы сможете построить беспроводную сеть, пусть сначала и небольшую, но все равно это будет настоящая сеть, причем легко модифицируемая. Со временем ваша беспроводная сеть сможет превратиться в достаточно объемную ячейку с большими возможностями. Если взяться за это с умом, то вскоре вы сможете заработать на своей сети приличные деньги – можете не сомневаться!

## От издательства

Ваши замечания, предложения и вопросы отправляйте по адресу электронной почты [gurski@minsk.piter.com](mailto:gurski@minsk.piter.com) (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На сайте издательства <http://www.piter.com> вы найдете подробную информацию о наших книгах.

# Глава 1

## Сетевая модель и протоколы передачи данных

- Модель ISO/OSI.
- Понятие протокола.
- NetBIOS и NetBEUI.
- TCP, IP и UDP.
- IPX и SPX.
- SMTP, POP3 и IMAP.
- SLIP, PPP, HTTP и FTP.

Наш мир живет и функционирует, основываясь на многих законах и правилах. Конечно, все явления нельзя четко классифицировать или подогнать под общие правила, но, тем не менее, в основе всего сущего лежат законы физики, механики, экономики и, в конце концов, законы природы, которым подчиняются все процессы, происходящие на планете. Если бы ни они, Земля погрузилась бы в полный хаос.

Аналогично построен и компьютерный мир. Правда, к нему применимо скорее понятие стандарта, чем закона. Именно благодаря существованию определенных стандартов производители знают, какими параметрами должно обладать конкретное оборудование и какие функции выполнять, чтобы уметь работать в тех или иных условиях.

Подобные правила и стандарты существуют также в мире сетей, вне зависимости от того, какие они, – проводные или беспроводные. Основными стандартами здесь являются модель взаимодействий ISO/OSI и протоколы передачи данных.

## 1.1. Модель ISO/OSI

Пожалуй, ключевым понятием в стандартизации сетей и всего, что к ним относится, является модель взаимодействия открытых систем (Open System Interconnection, OSI), разработанная международной организацией по стандартам (International Standards Organization, ISO). На практике применяется название *модель ISO/OSI*.

Описываемая модель состоит из семи уровней. Каждый уровень отвечает за определенный круг задач, выполняя их с помощью специальных алгоритмов – стандартов. Основная задача – достичь глобальной цели, поэтому уровни модели связаны между собой. Таким образом, выполнив свою часть задачи, каждый уровень передает готовые данные следующему уровню. В результате прохождения такой цепочки данные полностью обрабатываются, и их можно использовать.

В зависимости от назначения уровни получили следующие названия: физический, канальный, сетевой, транспортный, сеансовый, уровень представления данных и прикладной (рис. 1.1).

Основные отличия между проводными (Ethernet 802.3) и беспроводными (IEEE 802.11) сетями кроются только в двух нижних уровнях – физическом и канальном. Остальные уровни работают абсолютно одинаково, без каких-либо отличий.

### Физический уровень

*Физический* уровень – первый, самый низкий, уровень. Фактически он представляет собой аппаратную часть сети и описывает способ передачи данных, используя для этого любой имеющийся «под руками» канал – проводной или беспроводной. В зависимости от выбранного канала передачи данных используют соответствующее сетевое оборудование. Параметры передачи данных следует настраивать с учетом особенностей канала: полос пропускания, защиты от помех, уровня сигнала, кодирования, скорости передачи данных в физической среде и т. п.



Рис. 1.1. Уровни модели ISO/OSI.

Фактически всю описанную работу выполняет сетевое оборудование: сетевая карта, мост, маршрутизатор и т. д.

Физический уровень – один из уровней, который отличает беспроводные сети от проводных. Как вы уже, несомненно, поняли, основное отличие между ними заключается в канале передачи данных. Для проводных сетей это радиоволны определенной частоты или инфракрасное излучение, для беспроводных – любая физическая линия, например коаксиал, витая пара или оптоволокно.

## Канальный уровень

Главная задача *канального* уровня – удостовериться, что канал готов к передаче данных и ничто не станет угрожать надежности этой операции и целостности передаваемых пакетов. В идеале протоколы канального уровня и сетевое оборудование должны проверить, свободен ли канал для передачи данных, не имеется ли коллизий передачи и т. п.

Такую проверку необходимо выполнять каждый раз, так как локальная сеть чаще всего состоит более, чем из двух компьютеров (хотя даже в таком случае канал может быть занят). Обнаружив, что канал свободен, элементы этого уровня делят данные, которые необходимо передать другому компьютеру, на более мелкие части – кадры. Затем каждый кадр снабжается контрольной суммой и отправляется. Приняв этот кадр, получатель проверяет контрольные суммы

и, если они совпадают, принимает его и посылает отправителю подтверждение о доставке. В противном случае получатель игнорирует кадр и фиксирует ошибку, после чего кадр передается заново. Так, кадр за кадром, передается необходимая информация.

На канальном уровне, как и на физическом, также существуют различия между проводными и беспроводными сетями. Это связано со спецификой сетевого оборудования. Так, доступное на данный момент беспроводное оборудование работает только в полудуплексном режиме: в один момент времени данные могут только приниматься или только передаваться. Этот недостаток резко уменьшает эффективность обнаружения коллизий в сети и, соответственно, понижает скорость передачи данных.

Поскольку модель ISO/OSI жестко регламентирует действия каждого уровня, то разработчикам пришлось немного модернизировать протоколы канального уровня для работы в беспроводных сетях. В частности, для беспроводной передачи данных используются протоколы CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance, многостанционный доступ к среде передачи с контролем несущей и избеганием коллизий) или DCF (Distributed Coordination Function, распространяемая координирующая функция).

Протокол CSMA/CA предназначен для обнаружения конфликтов при передаче данных. Он использует явное подтверждение доставки данных, сообщаящее о том, что передаваемый пакет доставлен и не поврежден.

Данный метод работает следующим образом. Когда один компьютер собирается передать информацию другому, он посылает всем компьютерам сети короткое сообщение – RTS (Ready To Send, готов к передаче), содержащее информацию о получателе и времени, необходимом для передачи данных. Получив такой пакет, все компьютеры прекращают на указанный промежуток времени передачу собственных данных. Компьютер, для которого предназначен пакет, отправляет отправителю сообщение CTS (Clear To Send, свободен для передачи) о готовности к приему данных. Получив такое сообщение, компьютер-отправитель пересылает первую порцию данных и ждет подтверждения доставки пакета. После такого подтверждения передача данных продолжается. Если сообщение об успешной доставке не пришло, то компьютер-отправитель повторно передает необходимый пакет.

Такой метод передачи гарантирует доставку пакетов данных, однако в то же время заметно снижает скорость передачи. Именно поэтому беспроводные сети намного медленнее проводных и останутся таковыми надолго, если не навсегда. Чтобы хоть как-то повысить скорость передачи данных по таким сетям, специальный протокол канального уровня фрагментирует пересылаемые пакеты (разбивает их на более мелкие части), что увеличивает шанс их удачной передачи.

## Сетевой уровень

Как и канальный уровень, *сетевой* отвечает за передачу данных между компьютерами. Для этого он использует сформированные данные и параметры двух предыдущих уровней – физического и канального. Главное отличие сетевого уровня от канального заключается в том, что он умеет передавать данные между сетями с разной топологией – комбинированными. Так, очень часто беспроводные и проводные сети используются в паре. Чаще всего это происходит, если по определенным причинам создать единую проводную сеть физически невозможно.

На сетевом уровне, как и на канальном, данные делятся на пакеты, что позволяет достичь качества и определенной скорости их передачи. Однако, в отличие от канального, сетевой уровень выбирает для передачи данных конкретный маршрут. Процесс выбора оптимального маршрута передачи данных называется *маршрутизацией*.

Как правило, информацию о выборе маршрута предоставляют специальные устройства, установленные в сети, – маршрутизаторы. Специальные таблицы маршрутизаторов содержат

информацию о скорости передачи данных между отдельными отрезками сети, трафике, среднем времени передачи и т. д. Основываясь на этой информации, протоколы сетевого уровня могут выбрать оптимальный путь прохождения данных.

## **Транспортный уровень**

Пожалуй, *транспортный* уровень можно отнести к более высоким. Это означает, что данным уровнем управляет программа, а не аппаратные средства.

Транспортный уровень отвечает за надежность передачи данных. Существует несколько способов передачи, которые отличаются друг от друга степенью защищенности и возможностью исправления ошибок. Естественно, это сказывается на времени и скорости передачи информации между конкретными точками.

Способ передачи данных выбирается автоматически, с помощью анализа информации маршрутизаторов сети. Если анализ показывает, что конфликты в сети минимальны, то используется самый простой (а значит, самый быстрый) способ. В противном случае выбирается способ передачи высокой степени надежности с возможностью исправления поврежденных пакетов (этот способ передачи, конечно, более медленный). Какой бы способ ни был выбран, в любом случае информация будет доставлена в целостном виде.

## **Сеансовый уровень**

*Сеансовый* уровень предназначен для контроля передачи пакетов между компьютерами. В процессе синхронизации принятых и отправленных пакетов протоколы сеансового уровня отслеживают недостающие данные и передают их заново. За счет работы только с недостающими пакетами достигается повышение скорости передачи данных.

## **Уровень представления**

На уровне *представления* данные приводятся к единому стандарту, что позволяет достичь договоренности при их приеме и передаче. Именно на этом уровне данные могут шифроваться, что повышает безопасность их передачи по сети. Кроме того, часто на уровне представления происходит компрессия информации, благодаря чему повышается скорость передачи данных.

Уровень представления реализуется программно, что позволяет использовать для шифрования данных новейшие достижения.

## **Прикладной уровень**

*Прикладной* уровень – самый верхний уровень модели ISO/OSI. Его задача – организация взаимодействия с прикладными программами. За это отвечает множество прикладных протоколов, с помощью которых операционная система и программы получают доступ к разнообразным ресурсам сети.

## 1.2. Понятие протокола

В предыдущем разделе мы кратко рассмотрели модель ISO/OSI, которая описывает работу любого сетевого оборудования и сети в целом. Однако это всего лишь модель, рисунок на бумаге. Для начала работы необходим механизм, реализующий описываемую модель. Таким механизмом является протокол передачи данных, включающий в себя множество протоколов.

Таким образом, *протокол* – набор правил, благодаря которым возможна передача данных между компьютерами. Эти правила работают в рамках модели ISO/OSI и не могут отступать от нее ни на шаг, поскольку это может повлечь за собой несовместимость оборудования и программного обеспечения.

Каждый уровень модели ISO/OSI обладает своими особенностями, и реализовать все особенности в рамках одного протокола невозможно. Мало того, это даже невыгодно, поскольку значительную часть логики можно разрабатывать на уровне аппаратного обеспечения, что приводит к ускорению работы с данными. Исходя из этих соображений, было разработано множество узконаправленных протоколов, каждый из которых с максимальной отдачей и быстродействием выполняет свою задачу.

Протоколы могут быть двух типов: низкоуровневые и высокоуровневые.

- Низкоуровневые протоколы появились достаточно давно и с тех пор не претерпели никаких кардинальных изменений. За длительное время использования таких протоколов в них были найдены и устранены все возможные «дыры» и ошибки.

Примечание.

Низкоуровневые протоколы реализуются на аппаратном уровне, что позволяет добиться их максимального быстродействия.

- Что касается высокоуровневых протоколов, то они постоянно разрабатываются и совершенствуются. В этом нет ничего плохого, даже наоборот: всегда существует возможность придумать новый, более эффективный, способ передачи данных.

Примечание.

Как правило, высокоуровневые протоколы реализуются в виде драйверов к сетевому оборудованию для работы в разных операционных системах.

Существует множество различных протоколов, каждый из которых имеет свои особенности. Одни протоколы узконаправленные, другие имеют более широкое применение. Каждая компания разрабатывает свой собственный *стек* (набор) протоколов. Хотя разные стеки протоколов изначально несовместимы, существуют дополнительные протоколы, представляющие собой «мосты» между стеками. Благодаря этому в одной операционной системе можно работать с несколькими несовместимыми между собой протоколами.

Стоит также упомянуть тот факт, что не все протоколы можно использовать в одинаковых условиях. Иногда применение одного протокола выгодно для небольшой группы компьютеров и крайне невыгодно для большого количества компьютеров с несколькими маршрутизаторами и подключением к Интернету.

В следующих разделах вы познакомитесь с наиболее распространенными протоколами и стеками протоколов.

## 1.3. NetBIOS и NetBEUI

### NetBIOS

*NetBIOS* (Network Basic Input Output System, базовая система сетевого ввода/ вывода) – протокол (скорее, интерфейс) прикладного программирования, разработанный в конце 1983 года для компьютеров IBM.

На самом деле NetBIOS не является полноценным протоколом, поскольку описывает только программную часть передачи данных – набор сетевых API-функций. Это означает, что с помощью этого протокола можно только подготовить данные для передачи. Физически же передача может осуществляться только с помощью любого транспортного протокола, например TCP.

Благодаря такой ситуации подготовка передачи данных не привязана к транспортному протоколу, что позволяет использовать для этих целей любой подходящий протокол. Кроме того, неоспоримым достоинством NetBIOS является быстрдействие.

Однако, к сожалению, для полноценной работы протокола NetBIOS требуется, чтобы на всех компьютерах сети использовался один транспортный протокол, иначе компьютеры не смогут синхронизироваться. Основным недостатком NetBIOS является то, что он не поддерживает маршрутизацию, без которой не обходится любая мало-мальски развернутая сеть.

### NetBEUI

*NetBEUI* (NetBIOS Extended User Interface, протокол расширенного пользовательского интерфейса базовой системы сетевого ввода/вывода) – протокол, дополняющий NetBIOS. Благодаря NetBEUI появилась возможность не только описывать программный уровень передачи данных, но и передавать их физически по сети, используя специальные встроенные механизмы этого протокола. Кроме того, значительно возросла надежность и скорость передачи данных.

Основной недостаток NetBEUI, как и NetBIOS, – отсутствие механизма маршрутизации, что делает этот протокол бесполезным в больших сетях. Если же ваша сеть состоит из нескольких компьютеров и не располагает маршрутизатором, то более быстрого протокола вы не найдете.

Итак, протокол NetBEUI не поддерживает маршрутизацию в сети, что не позволяет эффективно использовать его скорость в глобальных сетях. Тем не менее, этот протокол является одним из основных компонентов NT-систем, и его установка происходит автоматически.

## 1.4. TCP, IP и UDP

### TCP

*TCP* (Transmission Control Protocol, протокол управления передачей данных) – распространенный протокол, разработанный много лет назад. Он используется не только в локальных сетях, но и в сети Интернет, что однозначно характеризует TCP с хорошей стороны.

Главным достоинством протокола является его надежность, достигаемая путем использования подтверждающих пакетов, которые присылаются каждый раз в ответ на полученное сообщение. При этом в первую очередь устанавливается логическая связь между компьютером-отправителем и компьютером-получателем, что гарантирует успешную доставку пакетов.

Еще одним механизмом надежности передачи данных является механизм, отслеживающий время жизни пакета, – TTL (Time To Live, время жизни). Если по истечении заданного времени компьютер-получатель не пришлет подтверждение о доставке очередного пакета данных, то компьютер-отправитель перешлет эти данные повторно. Кроме того, данные будут повторно посланы, если пакет оказался поврежденным и компьютер-получатель его отклоняет, о чем сообщает отправителю.

### IP

*IP* (Internet Protocol, протокол межсетевое взаимодействия) – протокол, который обычно применяется вместе с протоколом TCP. Для работы он использует готовые данные маршрутизации, поэтому не контролирует доставку сообщений адресату. Располагая информацией о маршрутизации между выбранными компьютерами, этот протокол просто добавляет к пакету адрес отправителя и получателя<sup>1</sup> и пересылает его дальше. Дальнейшая судьба отправленных данных неизвестна, поэтому функцию контроля должен выполнять другой протокол, в частности TCP.

Чтобы хоть как-то повысить надежность, протокол IP вкладывает в пакет контрольную сумму, что позволяет компьютеру-получателю удостовериться в том, что пакет принят без ошибок или, в противном случае, отвергнуть его.

Преимуществом протокола является возможность фрагментации (разделения на компьютере-отправителе большого пакета на более мелкие) с последующей их дефрагментацией на компьютере-получателе.

### UDP

*UDP* (User Datagram Protocol, протокол пользовательских дейтаграмм) – один из самых быстрых, но не очень надежных протоколов, которые используют в сети для передачи данных. Он работает практически так же, как и протокол IP, однако после удачного приема пакета компьютер-получатель присылает соответствующее подтверждение. При этом логическое соединение между компьютерами не требуется, то есть пакет отсылается в надежде (или с уверенностью) на то, что нужный компьютер находится в сети и может его принять. Если подтверждение доставки не получено, значит, через некоторое время компьютер-отправитель повторно вышлет необходимый пакет данных.

---

<sup>1</sup> При работе протокол использует понятие дейтаграммы – пакета данных, снабженных служебной информацией о получателе и отправителе.

Как ни странно, протокол UDP применяется в сети достаточно часто. Благодарить за это нужно скорость, с которой он работает. Эта скорость достигается за счет отсутствия необходимости соединения с другими компьютерами, что позволяет использовать трафик сети в нужном направлении. Так, протокол UDP часто используется, например, в сетевых играх или для передачи звуковых данных с интернет-радио (когда надежность доставки пакетов не играет большой роли).

## 1.5. IPX и SPX

Протоколы IPX и SPX являются представителями стека протоколов, разработанных компанией Novell. В свое время эта компания являлась прямым конкурентом Microsoft в области сетевых операционных систем: противостояли операционная система Novell Netware и Windows NT. Соответственно, каждая операционная система использовала собственный набор протоколов.

Как известно, компания Novell сдала свои позиции, и первенство завоевали сетевые версии операционной системы Windows NT. Тем не менее протоколы, разработанные Novell, используются до сих пор и будут использоваться еще очень долго.

### IPX

*IPX* (Internetwork Packet eXchange, протокол межсетевого обмена объектами) – один из самых «ходовых» протоколов, используемых на сетевом уровне. Главной его задачей является определение оптимального маршрута между двумя выбранными компьютерами с использованием для этого данных других протоколов.

В первую очередь данный протокол вычисляет адрес компьютера, которому необходимо отправить пакет – дейтаграмму. Определив адрес нужного компьютера, он снабжает дейтаграмму служебной информацией (адресом отправителя и получателя) и отправляет «в путь» по выбранному маршруту.

Однако самостоятельно этот протокол работать не может, поскольку не способен устанавливать соединение между компьютерами. Соответственно, от IPX нельзя ожидать высокой степени надежности доставки пакетов.

### SPX

*SPX* (Sequenced Packet eXchange, последовательный обмен пакетами) – «родной брат» IPX, без которого его нельзя назвать полноценным протоколом. Эти протоколы используются вместе и имеют общее название – IPX/SPX.

Главная задача протокола SPX – установка логического соединения между выбранными компьютерами с последующей передачей подготовленных дейтаграмм.

## 1.6. SMTP, POP3 и IMAP

Без протоколов SMTP, POP3 и IMAP невозможна работа электронной почты. Надеюсь, что такое электронная почта и как без нее плохо, объяснять не нужно.

Особенностью этих протоколов является их узкая направленность. Это означает, что их принципиально невозможно использовать для других целей, что, к тому же, не имеет смысла. Задача SMTP, POP3 и IMAP – организация обмена электронными сообщениями, и они отлично с ней справляются.

Еще одной особенностью почтовых протоколов является однозадачность. Например, протокол, отсылающий сообщения, не способен их принимать, и наоборот. Именно поэтому такие протоколы работают парами.

### SMTP

*SMTP* (Simple Mail Transfer Protocol, упрощенный протокол пересылки почты) – протокол, основной задачей которого является отсылка подготовленных специальным образом сообщений. Перед тем, как это сделать, протокол устанавливает соединение между компьютерами, что гарантирует доставку сообщения.

Протокол SMTP очень простой и эффективный, однако эта эффективность не распространяется на все задачи и возможности. Так, он не обладает даже простейшим механизмом аутентификации и возможностью шифрования данных при передаче между почтовыми серверами. Самым большим недостатком SMTP является его неспособность к пересылке графики.

Чтобы не отказываться от этого довольно хорошего протокола, было принято решение расширить его несколькими полезными и необходимыми расширениями. Таким расширением, например, является *MIME* (Multipurpose Internet Mail Extensions, многоцелевое расширение почтовой службы в Интернете), благодаря которому существует возможность отсылать файлы любого формата и содержания. Кроме того, разработан стандарт *UUENCODE*, позволяющий передавать текстовые сообщения в разных кодировках.

### POP3

*POP3* (Post Office Protocol 3, почтовый протокол версии 3) – почтовый протокол, который используется для приема электронных сообщений с почтового сервера.

Обычно POP3 работает в паре с протоколом SMTP, что позволяет организовать эффективную систему отсылки и приема электронных сообщений.

Интерфейс протокола еще более простой, чем интерфейс SMTP, и с этим связаны определенные неудобства. Так, отсутствует возможность выборочного скачивания письма или просмотра содержимого письма непосредственно на почтовом сервере.

### IMAP

*IMAP* (Interactive Mail Access Protocol, протокол интерактивного доступа к электронной почте) – наиболее «продвинутый» почтовый протокол, предназначенный для приема электронных сообщений с почтового сервера. В большинстве случаев более удобным и эффективным является использование протокола IMAP, чем POP3.

## **Конец ознакомительного фрагмента.**

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.