Министерство образования и науки, молодежи и спорта Украины
Севастопольский национальный технический университет

ИССЛЕДОВАНИЕ СЕТЕВЫХ ПРОТОКОЛОВ ТРАНСПОРТНОГО УРОВНЯ В БЕСПРОВОДНЫХ ЛОКАЛЬНЫХ СЕТЯХ

Методические указания
к выполнению лабораторной работы
по дисциплине **“Компьютерные сети”**

для студентов, обучающихся по направлению
6.050101 — **“Компьютерные науки”**
дневной и заочной формы обучения

Севастополь
2012

УДК 004.732

Исследование сетевых протоколов транспортного уровня в беспроводных локальных сетях. Методические указания / Сост. В.С.Чернега — Севастополь: Изд-во СевНТУ, 2012— 24 с.

Цель указаний: Оказать помощь студентам в изучении стека протоколов сети Интернет и особенностей обмена данными на транспортном уровне в беспроводных локальных компьютерных сетях, работающих в инфраструктурном режиме.

Методические указания предназначены для выполнения лабораторной работы по дисциплине "Компьютерные сети" для студентов дневной и заочной формы обучения

Методические указания рассмотрены и утверждены на методическом семинаре и заседании кафедры информационных систем
(протокол № 6 от 24 февраля 2012 г)

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент: Кротов К.В., канд. техн. наук, доцент кафедры ИС

СОДЕРЖАНИЕ

Стр.

1. ЦЕЛЬ РАБОТЫ

Исследование структуры заголовков пакетов стека протоколов TCP/IP и алгоритмов обмена данными в беспроводных локальных сетях с базовой станцией на транспортном уровне. Приобретение практических навыков мониторинга беспроводных компьютерных сетей и анализа заголовков кадров и пакетов данных.

2. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

2.1. Характеристика протоколов транспортного уровня

Стек протоколов TCP/IP — это сокращенное название набора (стека) сетевых протоколов разных уровней, используемых в Интернет. Группа протоколов TCP/IP делится на 4 уровня: прикладной (*application*), транспортный (*transport*), межсетевой (*internet*) и уровень доступа к среде передачи (*network access*) [1, 3, 4].

При продвижении пакета данных по уровням модели OSI сверху вниз на каждом новом уровне к пакету добавляется своя служебная информация в виде заголовка. В некоторых случаях в конце может добавляться и хвостовик (трейлер). Эта процедура называется инкапсуляцией данных верхнего уровня в пакете нижнего уровня (рисунок 2.1).

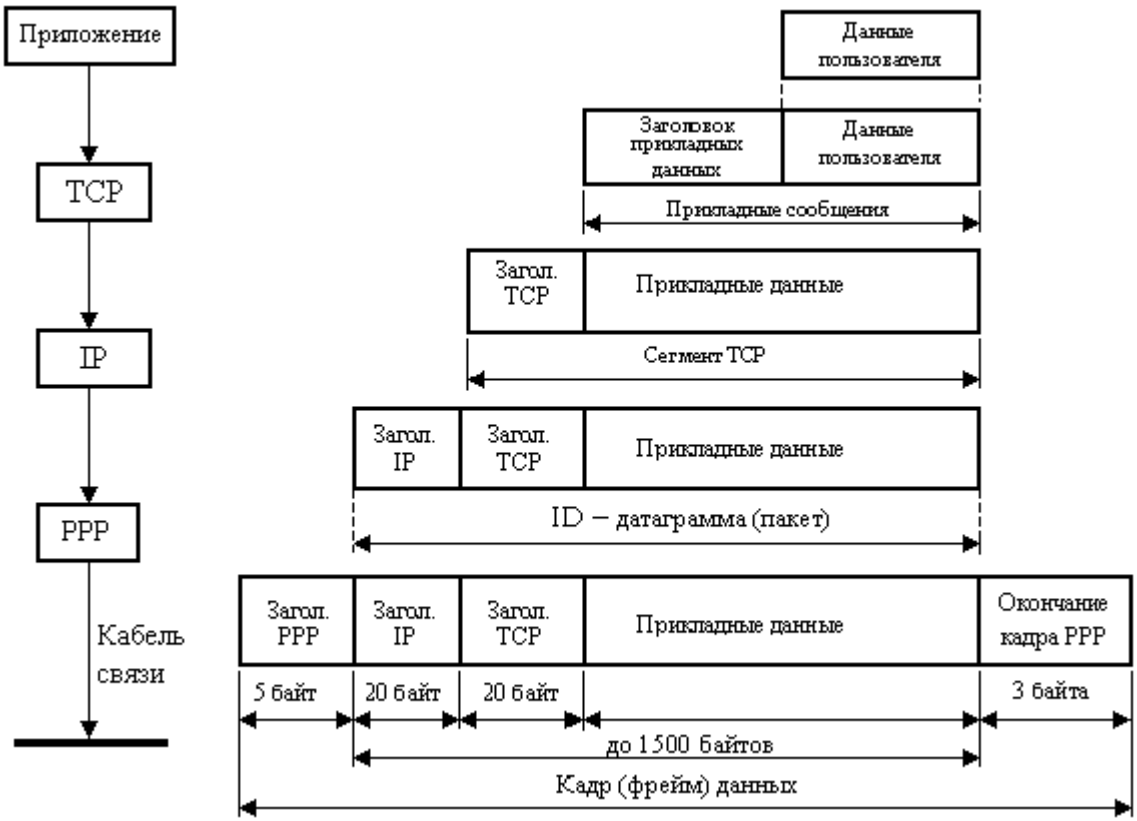

Рисунок 2.1 — Схема инкапсуляции данных на различных уровнях

Служебная информация предназначается для объекта того же уровня на удаленном компьютере, ее формат и интерпретация определяются протоколом данного уровня. На удаленной стороне при получении пакета от нижнего уровня он разделяется на заголовок, трейлер (при его необходимости) и данные. Служебная информация из заголовка анализируется и в соответствии с ней данные направляются одному из объектов верхнего уровня. Тот в свою очередь рассматривает эти данные как пакет со своей служебной информацией и данными для стоящего над ним уровня. Такая процедура повторяется, пока пользовательские данные, освобожденные от всей служебной информации, не достигнут прикладного процесса.

Протоколы транспортного уровня регламентируют прозрачную (сквозную) доставку данных между двумя прикладными процессами. Процесс, получающий или отправляющий данные с помощью транспортного уровня, идентифицируется на этом уровне номером, который называется номером порта. Таким образом, роль адреса отправителя и получателя на транспортном уровне выполняет номер порта. Заголовок транспортного уровня содержит также и другую служебную информацию. Его формат зависит от используемого транспортного протокола (UDP или TCP). Реализация протоколов транспортного уровня в компьютерной системе осуществляется соответствующими программными модулями, входящие в состав сетевой операционной системы.

TCP осуществляет надежную передачу данных между двумя компьютерами. Он обеспечивает деление данных, передающихся от одного приложения к другому, на сегменты подходящего для сетевого уровня размера, подтверждение станцией назначения принятых пакетов, установку тайм-аутов, в течение которых должно прийти подтверждение на пакет, и так далее.

UDP предоставляет более простой сервис для прикладного уровня. Он просто отсылает пакеты, которые называются дейтаграммами, от одного компьютера к другому. При этом нет никакой гарантии, что дейтаграмма дойдет до пункта назначения. За надежность передачи данных, при использовании дейтаграмм отвечает прикладной уровень. Для каждого транспортного протокола существуют различные приложения, которые их используют.

Формат заголовка TCP-сегмента транспортного уровня показан на рисунке 2.2. Поля заголовка имеют следующие значения.

Порты отправителя и получателя (*Source Port* и *Destination Port*) **—** номера портов процесса-отправителя и процесса-получателя соответственно. Оба поля занимают по 16 бит.

Номер последовательности (*Sequence Number, SN)*. Поле длиною 32 бита содержит порядковый номер первого октета в поле данных сегмента среди всех октетов потока данных для текущего соединения, то есть если в сегменте пересылаются октеты с 2001-го по 3000-й, то SN=2001. Если в заголовке сегмента установлен бит SYN (фаза установления соединения), то в поле SN записывается начальный номер ISN (*Initial Sequence Number*). Начальный номер выбирается из специального 32-разрядного счетчика, который при запуске TCP-модуля сбрасывается в нуль и затем инкрементируется каждые 4 мкс. Номер первого октета данных, посылаемых после завершения фазы установления соединения, равен ISN+1.

Номер подтверждения (*Acknowledgment Number*) —32 бита. Если установлен бит поля флагов ACK, то данное поле содержит порядковый номер октета (байта), который отправитель пакета подтверждения желает получить. Это означает, что все предыдущие октеты (с номерами от ISN+1 до ACK-1 включительно) были успешно получены.

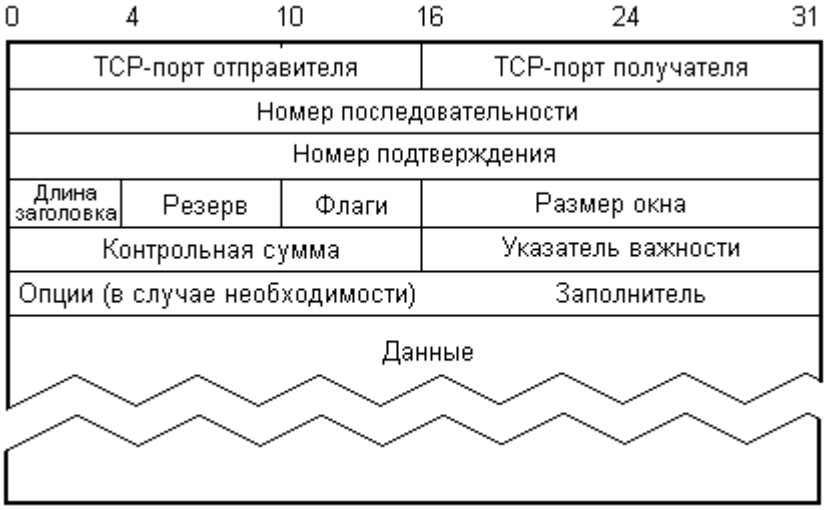

Рисунок 2.2 — Формат заголовка TCP-пакета транспортного уровня

Смещение заголовка (4 бита) — длина TCP-заголовка в 32-битных словах.

Резерв (6 бит) - зарезервировано; заполняется нулями.

Флаги (6 бит) — управляющие биты; активным является положение “бит установлен”. В одно и то же время могут быть установлены в единицу один или несколько флагов.

- **URG —** показывает на наличие в сегменте срочных данных, размер которых указан в поле срочного указателя (*Urgent Pointer*);
- **ACK —** бит подтверждения правильности приема; сигнализирует, что поле номера подтверждения (*Acknowledgment Number*) задействовано;
- **PSH —** информирует, что необходимо осуществить “проталкивание”; если модуль TCP получает сегмент с установленным флагом PSH, то он немедленно передает все данные из буфера приема процессу-получателю для обработки, даже если буфер не был заполнен;
- **SYN —** запрос на установление соединения;
- **FIN —** показывает, что у отправителя нет больше данных для передачи;
- **RST —** указывает на необходимость разрыва текущего соединения.

Размер окна (*Window*) — размер окна (16 бит) в октетах; показывает, сколько байтов может быть отправлено после байта, получившего подтверждение.

Контрольная сумма (*Checksum*) — поле длиною 16 бит. При вычислении контрольной суммы учитывается содержание заголовка сегмента и поля данных. Если поле данных содержит нечетное число октетов, то оно дополняется нулевым байтом. Контрольная сумма формируется путем сложения всех 16-разрядных слов в дополнительном коде и вычисления дополнения для полученной суммы.

Указатель срочности (*Urgent Pointer*) длиною 16 бит **—** используется для указания длины срочных данных, которые размещаются в начале поля данных сегмента. Указывает смещение октета, следующего за срочными данными, относительно первого октета в сегменте. Поле *Urgent Pointer* задействовано, если установлен флаг URG.

Опции — поле переменной длины; может отсутствовать или содержать одну опцию или список опций, реализующих дополнительные услуги протокола TCP. Опция состоит из октета "Тип опции", за которым могут следовать октет "Длина опции в октетах" и октеты с данными для опции.

Заполнитель — выравнивание заголовка по границе 32-битного слова путем добавления нулей, если список опций занимает нецелое число 32-битных слов.

Формат заголовка UDP-дейтаграммы показан на рисунке 2.3.

Рисунок 2.3 **—** Формат заголовка UDP-дейтаграммы транспортного уровня

Поля **UDP-порт источника** и **UDP-порт места назначения** (*Source Port* и *Destination Port*) **—** номера портов процессов отправителя и получателя соответственно.

Длина сообщения (*Length*) **—** длина UDP-пакета вместе с заголовком в октетах.

Контрольная сумма (*Checksum*). Вычисляется таким же образом, как и в TCP-заголовке; если UDP-пакет имеет нечетную длину, то при вычислении контрольной суммы к нему добавляется нулевой октет.

Для транспортировки пакетов транспортного уровня используются сетевые дейтаграммы. Заголовок IP-дейтаграммы (рисунок 2.4) состоит из 32-разрядных слов и имеет переменную длину, зависящую от размера поля “*Options*”, но всегда кратную 32 битам. За заголовком непосредственно следуют данные, передаваемые в дейтаграмме.

Версия (4 бита) **—** версия протокола IP, в настоящий момент используется версия 4, новые разработки имеют номера версий 6-8.

Длина заголовка (*Internet Header Length*) **—** 4 бита. Отображает длину заголовка в 32-битных словах; диапазон допустимых значений от 5 (минимальная длина заголовка, поле “*Options*” отсутствует) до 15 (т.е. может быть максимум 40 байт опций).

Тип сервиса (*Type Of Service, TOS*) **—** значение поля длиною 8 бит определяет приоритет дейтаграммы (3 бита). Следующие три бита (D, T и R) указывают

желаемый способ маршрутизации: D=1 — маршрут с минимальной задержкой; T=1 — маршрут с максимальной пропускной способностью; R=1 — маршрут с максимальной надежностью. Старшие два бита этого байта не используется.

Полная длина (*Total Length,* 16 бит) — длина всей дейтаграммы в октетах, включая заголовок и данные, максимальное значение 65535, минимальное — 21 (заголовок без опций и один октет в поле данных).

Рисунок 2.4 — Формат заголовка IP-дейтаграммы сетевого уровня

Идентификатор ID (*Identification*, 16 бит).

Флаги (*Flags*).3-битовое поле содержит признаки, связанные с фрагментацией: DF (*Do not Fragment*) — нельзя фрагментировать; MF (*More Fragments*) — это не последний фрагмент. Третий бит зарезервирован.

Смещение фрагмента (*Fragment Offset*) — поле длиной 13 бит используются для фрагментации и сборки дейтаграмм. Показывает смещение фрагмента в байтах от начала поля данных исходной дейтаграммы.

Время жизни (*TTL — Time To Live,* 8 бит) — устанавливается отправителем, измеряется в секундах. Каждый маршрутизатор, через который проходит дейтаграмма, переписывает значение TTL, предварительно вычтя из него время, потраченное на обработку дейтаграммы. Так как в настоящее время скорость обработки данных на маршрутизаторах велика, на одну дейтаграмму тратится обычно меньше секунды, поэтому фактически каждый маршрутизатор вычитает из TTL единицу. При достижении значения TTL=0 дейтаграмма уничтожается, при этом отправителю может быть послано соответствующее ICMP-сообщение. Контроль TTL предотвращает зацикливание дейтаграммы в сети.

Протокол (*Protocol*, 8 бит) — определяет программный модуль (вышестоящий протокол стека), которому должны быть переданы данные дейтаграммы для дальнейшей обработки. Коды протоколов TCP и UDP соответственно раны 4 и 7.

Контрольная сумма заголовка (*Header Checksum*, 16 бит) — представляет из себя 16 бит, дополняющие биты в сумме всех 16-битовых слов заголовка. Перед вычислением контрольной суммы значение поля “Header Checksum” обнуляется. Поскольку маршрутизаторы изменяют значения некоторых полей заголовка

при обработке дейтаграммы (как минимум, поля “TTL”), контрольная сумма каждым маршрутизатором пересчитывается заново. Если при проверке контрольной суммы обнаруживается ошибка, дейтаграмма уничтожается.

Адрес источника (*Source Address*, 32 бита) **—** IP-адрес отправителя.

Адрес места назначения (*Destination Address*, 32 бита) **—** IP-адрес получателя.

Опции (*Options*) **—** поле переменной длины. Опций может быть одна, несколько или ни одной. Опции определяют дополнительные услуги модуля IP по обработке дейтаграммы, в заголовок которой они включены

Заполнитель (*Padding*) **—** заполняется нулями для выравнивание заголовка по границе 32-битного слова, если список опций занимает нецелое число 32-битных слов.

2.2. Обмен данными на транспортном уровне

Протокол пользовательских дейтаграмм UDP (*User Datagram Protocol*,) не выполняет каких-либо особых функций дополнительно к функциям межсетевого уровня. Этот протокол используется либо при пересылке коротких сообщений, когда накладные расходы на установление сеанса и проверку успешной доставки данных оказываются выше расходов на повторную (в случае неудачи) пересылку сообщения, либо при передаче данных, к которым не предъявляются жесткие требования по безошибочной передаче (аудио-, видеопотоки).

Протокол UDP рассматривает данные приложения как целостное сообщение; он никогда не разбивает сообщение для передачи в нескольких пакетах и не объединяет несколько сообщений для пересылки в одном пакете. Если прикладной процесс *N* раз вызвал модуль UDP для отправки данных (т.е. запросил отправку *N* сообщений), то модулем UDP будет сформировано и отправлено *N* пакетов, и процесс-получатель будет должен *N* раз вызвать свой модуль UDP для получения всех сообщений. При получении пакета от межсетевого уровня модуль UDP проверяет контрольную сумму и передает содержимое сообщения прикладному процессу, чей номер порта указан в поле “*Destination Port*”.

В случае обнаружения ошибки или не существует процесса, подключенного к требуемому порту, принятый пакет отбрасывается. Если пакеты поступают быстрее, чем модуль UDP успевает их обрабатывать, то поступающие пакеты также игнорируются. Протокол UDP не имеет никаких средств подтверждения безошибочного приема данных или сообщения об ошибке. UDP не обеспечивает приход сообщений в порядке их отправки, не производит предварительного установления сеанса связи между прикладными процессами, поэтому он является ненадежным протоколом без установления соединения. Если приложение нуждается в подобного рода услугах, оно должно использовать на транспортном уровне протокол TCP.

TCP — это протокол, ориентированный на соединение. Перед тем как какая-либо сторона может послать данные другой, между ними должно быть установлено соединение. Процедура установления соединения происходит следующим образом.

1. Станция A, желающая установить соединение со станцией B, посылает SYN-сегмент (TCP-пакет), состоящий только из заголовка. В нем указан номер порта станции B, к которому она хочет подсоединиться, начальный номер своей последовательности ISN(A) (*Initial Sequence Number*) — число, начиная с которого станция A будет нумеровать отправляемые байты (октеты) и количество байтов, которая станция готова принять (Windows).
2. Станция B подтверждает приход сегмента SYN станции A передачей TCP-пакета (тоже состоящего только из заголовка) с установленным флагом ACK и с указанием ISN(A) станции A увеличенным на 1. Так как протокол TCP обеспечивает полнодуплексную передачу данных, то в этом же пакете станция B устанавливает флаг SYN и передает свой исходный номер ее последовательности ISN(B) и максимальный объем данных (размер окна), которые она разрешает передавать в свою сторону, если станция A еще не получила подтверждение (квитанцию) на предыдущую порцию данных.
3. Станция A должна подтвердить приход SYN от станции 2 путем передачи TCP пакета с установленным флагом ACK и ISN(B) станции B увеличенным на 1.

Соединение установлено, т.е. трех сегментов достаточно для установления соединения. Эту процедуру часто называют трехразовым рукопожатием (*three-way handshake*).

Если за установлением соединения сразу следует передача полезных данных, то пакет данных включается в сегмент, нумерация которого начинается с номера ISN(A)+1. Данные нумеруются по количеству отправленных октетов. В заголовке этого же сегмента станция A устанавливает бит ACK, подтверждающий установление связи B с A, что позволяет станции B включить в свой следующий сегмент полезные данные для A.

Для ускорения и оптимизации процесса передачи больших объемов данных протокол TCP определяет метод управления потоком, называемый методом скользящего окна, который позволяет отправителю посылать очередной сегмент, не дожидаясь подтверждения о получении в пункте назначения предшествующего сегмента.

В этом случае модуль TCP формирует подтверждения не для каждого конкретного успешно полученного пакета, а для всех данных от начала посылки до некоторого порядкового номера ACK SN (*Acknowledge Sequence Number*) исключительно. В качестве подтверждения успешного приема, например, первых 2000 байт, высылается ACK SN = 2001: это означает, что все данные в байтовом потоке под номерами от ISN+1=1 до данного ACK SN − 1 (2000) успешно получены.

Для разрыва соединения требуется отправить 4 сегмента. Это связано с тем, что TCP соединение полнодуплексное и поэтому каждое направление должно быть закрыто независимо от другого. Для этого каждая станция должна послать пакет с установленным флажком FIN и получить пакет с подтверждением его приема. Модуль TCP, приняв FIN, уведомляет приложение, что удаленная сторона разрывает соединение и прекращает передачу данных в этом направлении.

2.3. Особенность обмена данными в беспроводных локальных компьютерных сетях с базовой станцией

При передаче данных в сети с базовой станцией (точкой доступа) пакет, передаваемый от источника к получателю, сначала поступает на базовую станцию и буферизируется, а затем передается станции назначения [2, 4]. Если передача осуществляется по протоколу транспортного уровня TCP, то станция получатель обязана послать отправителю пакет подтверждения, который тоже буферизируется точкой доступа. Особенность обмена данными состоит в том, что и базовая станция, и станция-получатель не могут немедленно отправить пакеты транспортного уровня, так как для этого они должны выиграть в конкурентной борьбе за право доступа к среде. Это вносит в процесс обмена случайные задержки, которые в ряде случаев могут превышать допустимый интервал ожидания.

Формат кадра, передаваемого в беспроводной локальной сети на физическом уровне, показан на рисунке 2.5.

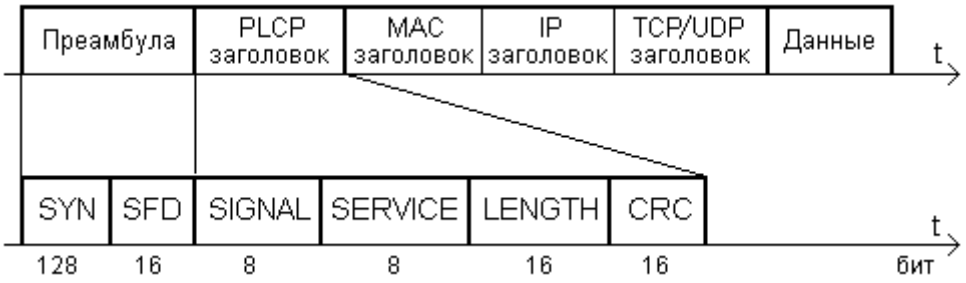

Рисунок 2.5 — Формат кадра физического уровня

Данные конкретного приложения с прикладного уровня поступают на транспортный уровень, где они делятся на сегменты и дополняются заголовком транспортного уровня TCP или UDP. Пакеты транспортного уровня на сетевом уровне дополняются IP-заголовком и передаются на канальный MAC-уровень, где к IP-дейтаграмме присоединяется заголовок MAC-уровня. Если размер пакетов превышает предельно допустимую длину кадра, то осуществляется их фрагментация.

Перед передачей MAC-кадра в канал к нему добавляется заголовок физического уровня, состоящий из преамбулы длиною 144 бита и собственно заголовка **PLCP** (*Physical Layer Convergence Protocol*) размером 48 битов. Преамбула служит для обеспечения тактовой и цикловой синхронизации передающей и приемной частей. Она состоит из синхронизирующей битовой последовательности **SYN** вида 1010…, которая завершается маркерной кодовой комбинацией **SFD** (*Start Frame Delimiter*) вида F3A0h, сигнализирующей о начале кадра.

PLCP-заголовок содержит поле SIGNAL с информацией о скорости передачи и способе модуляции, поле SERVICE с дополнительными сведениями о наличии вариантов расширений и поле LENGTH, в котором указана длина временного интервала в микросекундах, необходимого для передачи следующей за заголовком части кадра. Все поля заголовка кодируются помехоустойчивым циклическим

кодом с образующим полиномом 16-й степени. Результат кодирования помещается в поле контрольной последовательности CRC. Все биты заголовка передаются со скоростью 1 Мбит/с. Остальная часть кадра может передаваться с другой допустимой данным стандартом скоростью, которая указывается в полях SIGNAL и SERVICE.

На MAC-уровне беспроводных локальных сетей стандарта 802.11 применяются кадры трех типов:

- **служебные** (*Management Frames*) — обеспечивают соединения в сетях, аутентификацию, а также сообщают о состоянии оборудования;
- **информационные** (*Data Frames*) — выполняют транспортирование пользовательских данных;
- **управляющие** (*Control Frames*) — поддерживают передачу кадров данных при нормальном обмене информацией между станциями стандарта 802.11.

Форматы всех трех типов кадров похожи на основной формат, применяемый в сетях 802.11. Все они имеют заголовки с множеством полей, используемых подуровнем MAC. Названные три типа кадров расширяют и используют специфические участки кадра для своих целей.

Структура MAC-кадра основного формата показана на рисунке 2.3.

Управление кадром	Продол-житель-ность/ID	Адрес 1	Адрес 2	Адрес 3	Управле-ние оче-редностью	Адрес 4	Данные	КПК
2 байта	2	6	6	6	2	6	0 - 2312	4

Рисунок 2.6 — Основной формат MAC-кадра сети 802.11

Поле "Управление кадром" (*Frame Control*) занимает два байта и содержит 11 вложенных подполей, которые выполняют следующие функции:

- "Версия протокола" (2 бита) — указывает версию протокола MAC-подуровня. Подполе имеет значение 0 (используется только одна версия), остальные значения зарезервированы.
- "Тип" (2 бита) — указывает тип MAC-кадра: информационный, служебный или управляющий.
- "Подтип" (4 бита) — уточняет тип и подтип кадра (например, "запрос на соединение", "ответ на запрос", "зондирующий запрос", "аутентификация", кадры RTS или CTS, ACK и др.). Более подробное значение полей приведено в [11].

Далее располагаются однобитовые подполя, выполняющие следующие функции.

- Один бит "К DS" и один бит "От DS" — индицируют о направлении движения кадра: к междоменной системе распределения DS (например, Ethernet) или от нее. Координационная функция MAC присваивает биту "К DS" значение 1, если кадр предназначен распределительной системе, и единичное значение биту "От DS", если кадр исходит от распределительной системы. MF (*More Fragments*) — информирует о том, что далее следует еще один фрагмент.

• "Повтор" — отмечает повторно посылаемый фрагмент.

• "Управление питанием" — указывает на режим энергопотребления рабочей станции: "1" — режим пониженного потребления, "0" — активный режим. В кадрах точки доступа этот бит всегда сброшен в "0".

• "Продолжение" (*More Data*) — свидетельствует о том, что у отправителя имеются еще кадры для пересылки.

• W — является индикатором использования шифрования в поле данных кадра по алгоритму WEP (*Wired Equivalent Protocol*).

• O (*Order*) — информирует приемник о том, что кадры с этим битом должны обрабатываться строго по порядку.

Последующие поля кадра идентифицируются следующим образом.

• "Продолжительность / ID" (2 байта) — используется по-разному, в зависимости от того, получает ли доступ к среде рабочая станция, работающая в энергосберегающем режиме, находится ли среда в режиме сосредоточенной координации PCF периода свободного от конкуренции (CFP), или доступ к среде получает станция с распределенной функцией координации DCF (детальнее функции PCF и DCF рассматриваются в третьем разделе). В частности, для рабочей станции с DCF здесь указывается время передачи данных и ожидания ответа.

• "Адрес 1, 2" — содержат адреса рабочих станций отправителя и получателя соответственно, а поля "Адрес 3 и 4" входят в состав кадра только при соединении двух сетей через распределительную систему DS. В этом случае третий адрес обозначает адрес точки доступа сети отправителя, а четвертый — получателя.

• "Управление очередностью" — содержит порядковый номер и номер фрагмента кадра. Из 16 доступных бит 12 идентифицируют кадр, а 4 — фрагмент. "КПК" — контрольная 32-разрядная последовательность кадра (CRC), формируемая в результате помехоустойчивого циклического кодирования.

Процедура равноправного доступа компьютеров к среде на основе способа распределенной координации DCF иллюстрируется временной диаграммой, изображенной на рисунке 2.7 [2, 4].

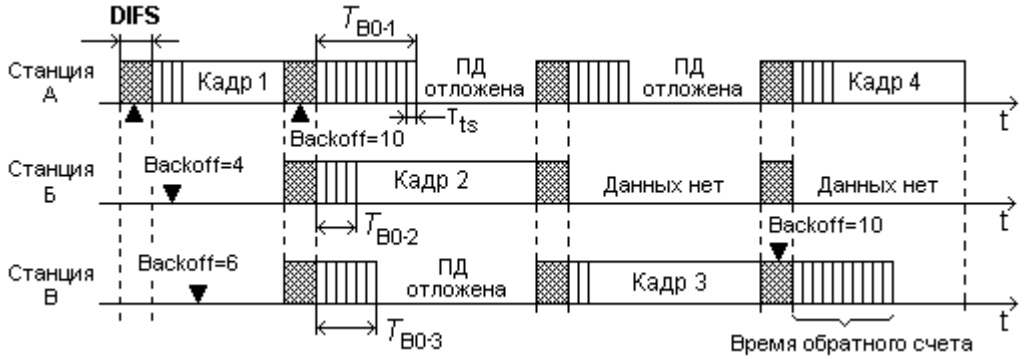

Рисунок 2.7 — Реализация равноправного доступа к среде передачи

Если узел сети пытается получить доступ к среде, то после обязательного промежутка ожидания DIFS запускается процедура обратного отсчета, т.е. включается обратный отсчет счетчика тайм-слотов, начиная от выбранного случайного

значения окна CW. Если в течение всего промежутка ожидания среда оставалась свободной, то узел начинает передачу. В данном примере диаграмма иллюстрирует асинхронный доступ к среде трех станций. Черными треугольниками на диаграмме отмечены моменты подачи запросов передачи данных от верхнего уровня.

Из рисунка видно, что запрос на передачу от станции А поступил в момент времени, когда среда не занята. Поэтому после истечения обязательного фиксированного межкадрового интервала ожидания **DIFS** станция А начинает передачу кадра. Во время его передачи поступили запросы от станций Б и В. Так как среда передачи занята, обе станции вычисляют случайное значение интервала обратного отсчета T_{BO} (*Backoff*), которое в нашем примере для станции Б равно 4 тайм-слота, а для станции В — 6. Передача следующего кадра начнется станцией Б лишь после завершения фиксированного интервала DIFS и уменьшении значения счетчика обратного счета до нуля.

Для минимизации коллизий после каждого успешного приема кадра принимающая сторона через короткий промежуток **SIFS** (*Short Interframe Space*) подтверждает прием, посылая ответную квитанцию — кадр подтверждения на канальном уровне ACK (рисунок 2.8). Другим станциям разрешается отправлять кадры лишь по окончании кадра ACK и после выдержки паузы DIFS.

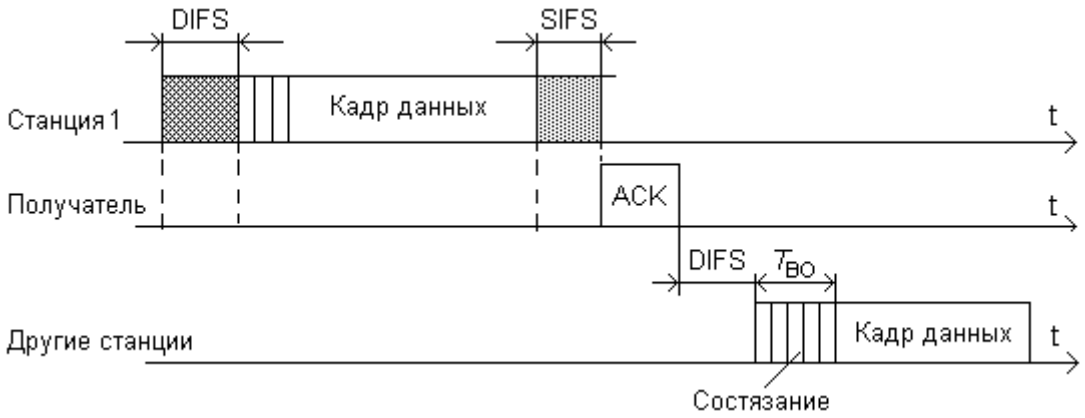

Рисунок 2.8 — Иллюстрация приоритетности передачи кадра ACK

Если в процессе передачи данных возникла коллизия, то передающая сторона не получает кадр ACK об успешном приеме. В этом случае размер CW-окна для передающего узла увеличивается почти вдвое.

3. Описание лабораторной установки

В состав лабораторной установки входят четыре рабочие станции, оснащенные радиоадаптерами стандарта IEEE 802.11b/g и точка доступа с функциями маршрутизатора (рисунок 3.1). Рабочие станции и точка доступа образуют локальную беспроводную компьютерную сеть, функционирующую в инфраструктурном режиме.

Для выполнения необходимых измерений сетевых параметров на рабочих станциях установлено специальное программное обеспечение в таком составе:

IxChariot (демоверсия);
CommView for WiFi (оценочная версия).

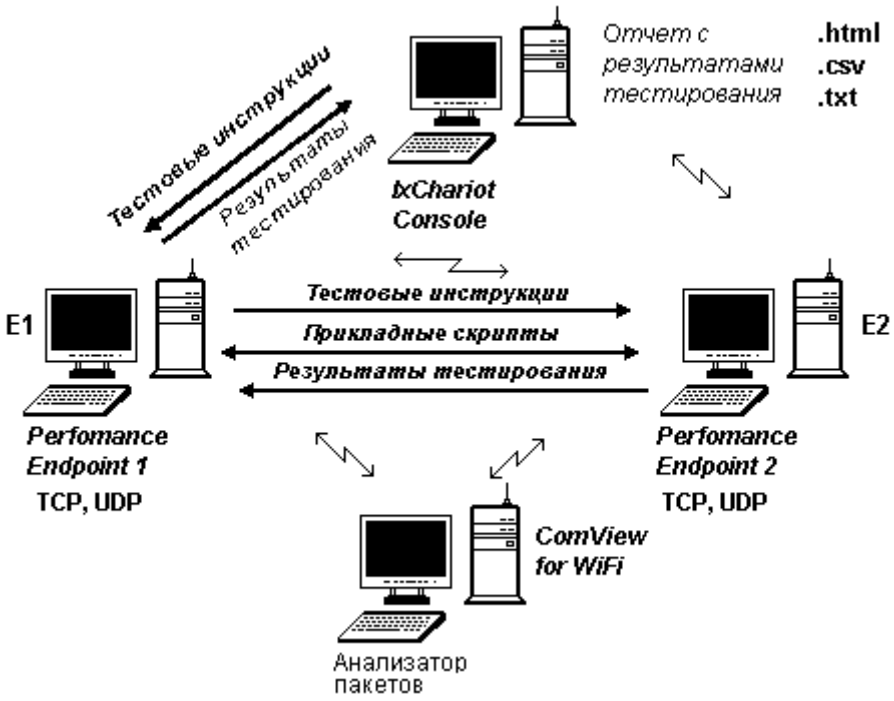

Рисунок 3.1 — Схема лабораторной установки

IxChariot — это инструмент на базе программного обеспечения для оценки параметров сетей, используемый для измерения ключевых функциональных характеристик, таких как пропускная способность, время задержки, потеря пакетов, разброс временных интервалов между пакетами (jitter), показатель MOS (*Mean Opinion Score*), используемый для субъективной оценки качества передаваемых аудио- и видеоданных для VoIP и коэффициент доставки мультимедийных данных MDI (*Media Delivery Index*) для видео в реальных условиях. Программное обеспечение состоит из консоли IxChariot и функциональных оконечных точек (Performance Endpoints).

Измерения рабочих характеристик проводятся путем передачи реальных потоков данных между устройствами, подключенными к сети. IxChariot эмулирует различные типы приложений, собирает и анализирует полученные результаты. Оконечные точки IxChariot генерируют трафик, используя те же методы, что и любое сетевое приложение, позволяя измерить каждый элемент в тракте передачи данных. Адреса отправителя и получателя и измеряемый параметр задается в окне Edit an Endpoint Pair (рисунок 3.2). В этом же окне задается тип транспортного протокола, измеряемый параметр и вид генерируемого трафика. Последние выбираются путем нажатия клавиши Select Script.

Изменение установленных по умолчанию параметров трафика осуществляется в окне Script Editor (рисунок 3.3). Для редактирования параметра необходимо сделать по соответствующей строке двойной щелчок правой кнопкой мышки.

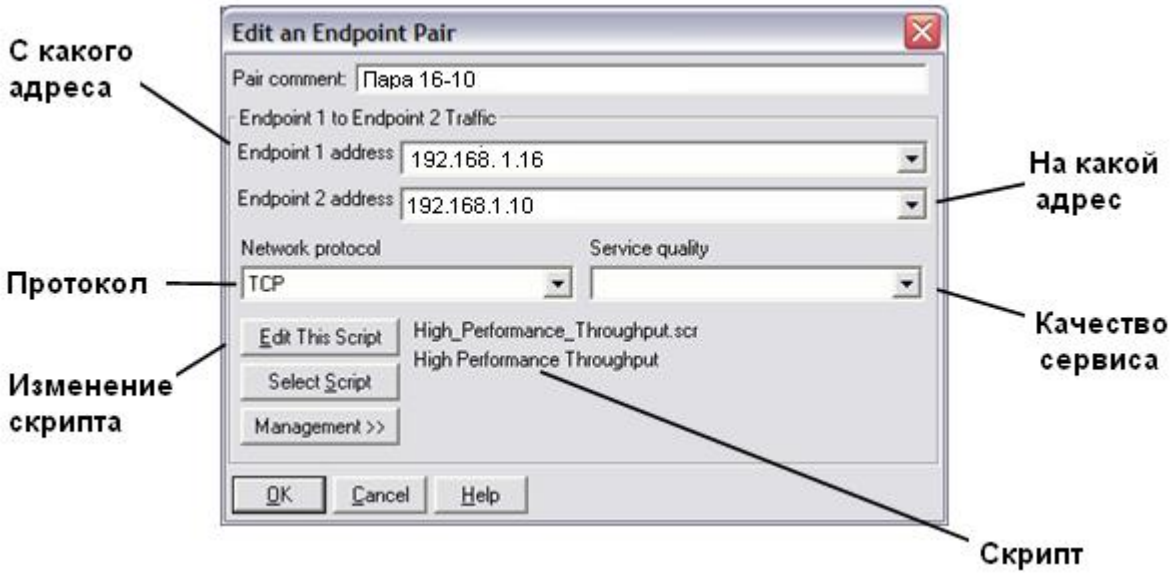

Рисунок 3.2 — Вид окна задания адресов станций и измеряемого параметра

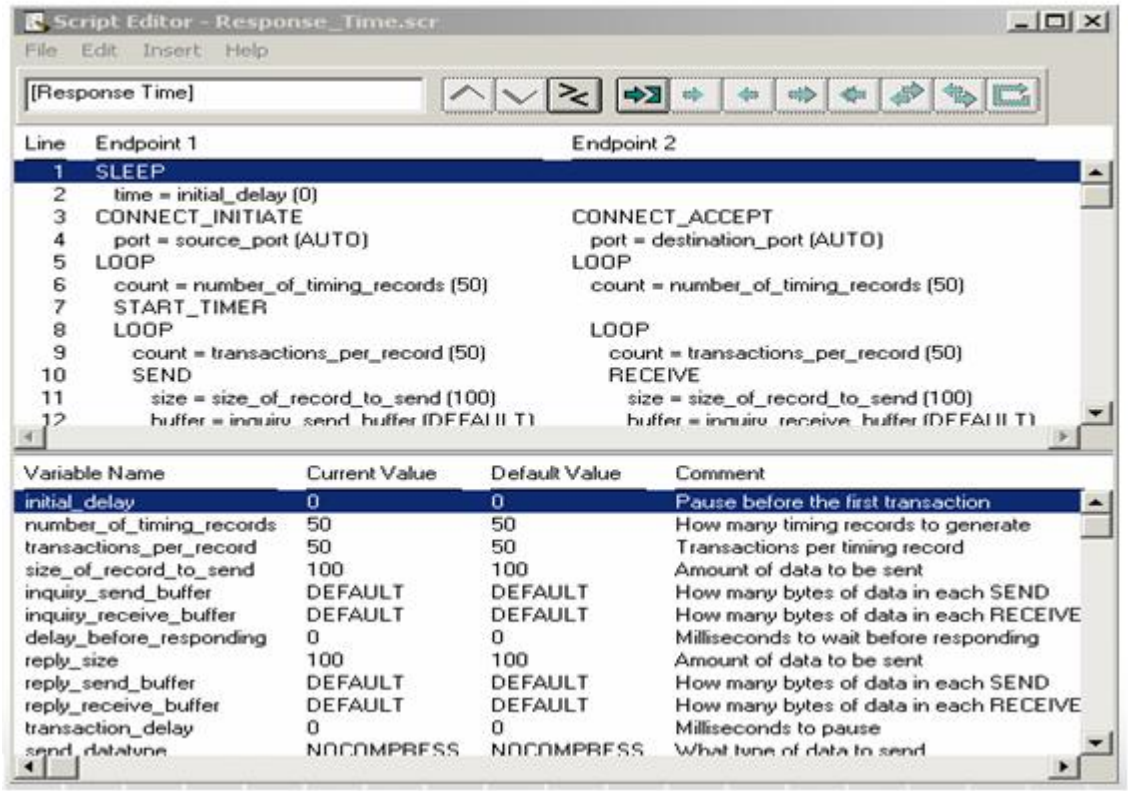

Рисунок 3.3 — Вид окна установки параметров передачи трафика

Запуск процесса измерения исследуемого параметра осуществляется нажатием кнопки Run. В результате открывается окно IxChariot Test (рисунок 3.4), на котором через несколько секунд начнет строиться график изменения измеряемого параметра во времени.

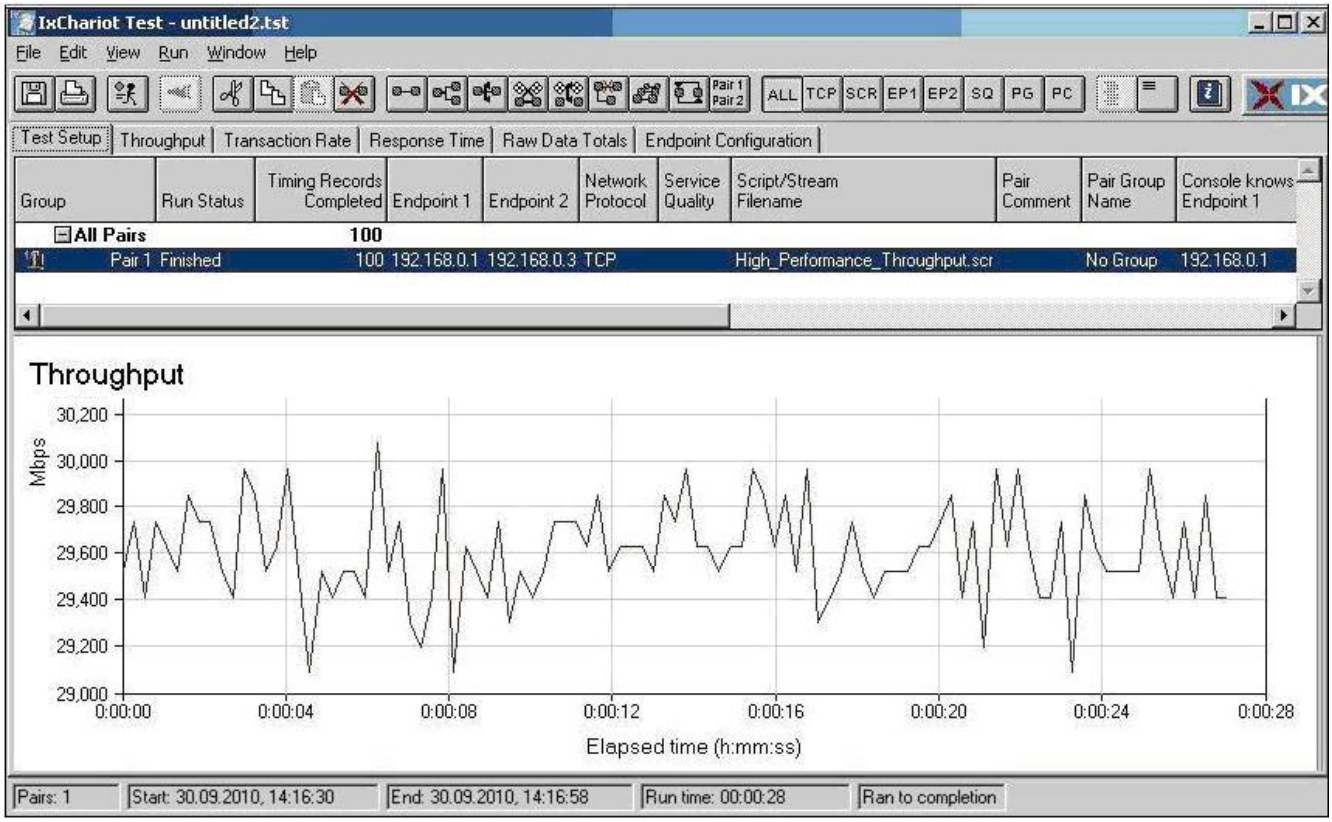

Рисунок 3.4 — Вид окна отображения измеряемой скорости передачи

Для анализа пакетов следует запустить программу CommView for WiFi. На мониторе компьютера появится главное окно (рисунок 3.5). Просканировать эфир на предмет сигналов обнаружения сигналов WiFi, а также выбрать канал для мониторинга можно в окне "Сканер". Сканирование начинается с нажатия кнопки "**Начать сканирование**". Процесс сканирования циклический, т. е. программа "слушает" сигналы на первом канале, потом переключается на следующий канал и т. д., пока не достигнет последнего канала, после чего начнется новый цикл сканирования. Процесс сканирования прекращается нажатием кнопки "**Остановить Сканирование**". Чтобы убрать всю собранную информацию следует нажать "**Сброс**". Сохранение отчет о сканировании в формате HTML происходит после нажатия "**Сохранить**". Если сбор пакетов нужно осуществлять на одном канале, то нужно в выпадающем списке "**Диапазон**" указать стандарт исследуемой сети (802.11b, 802.11b/g, или 802.11b/g/n), затем в выпадающем списке **Канал** выбрать желаемый канал и нажать "**Захват**".

Для указания типа перехватываемых пакетов следует выбрать в меню "Правила". Переключение в режим просмотра захваченных пакетов осуществляется выбором закладки "Пакеты".

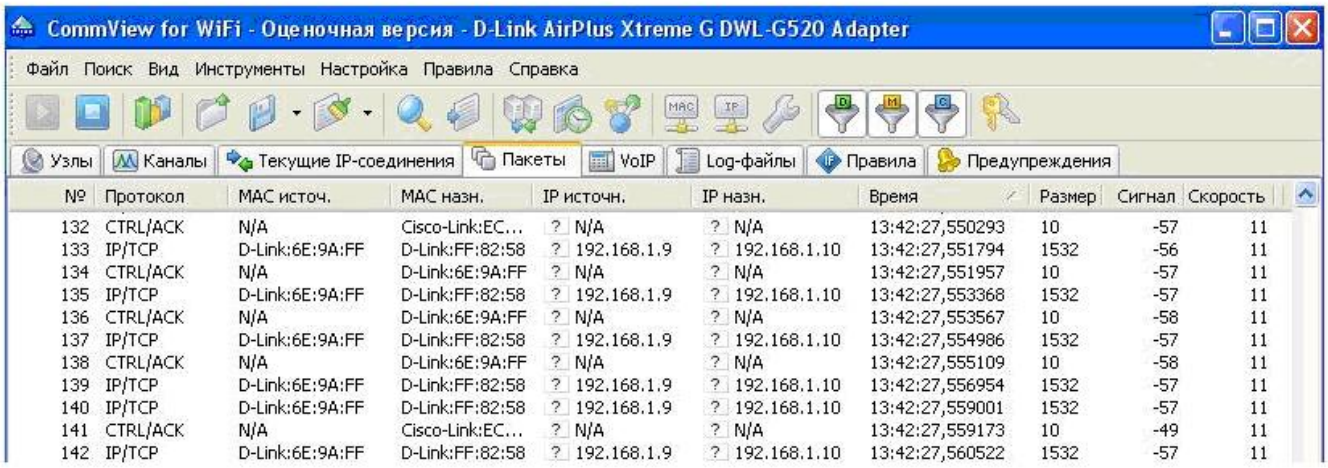

Рисунок 3.5 — Вид окна анализатора пакетов

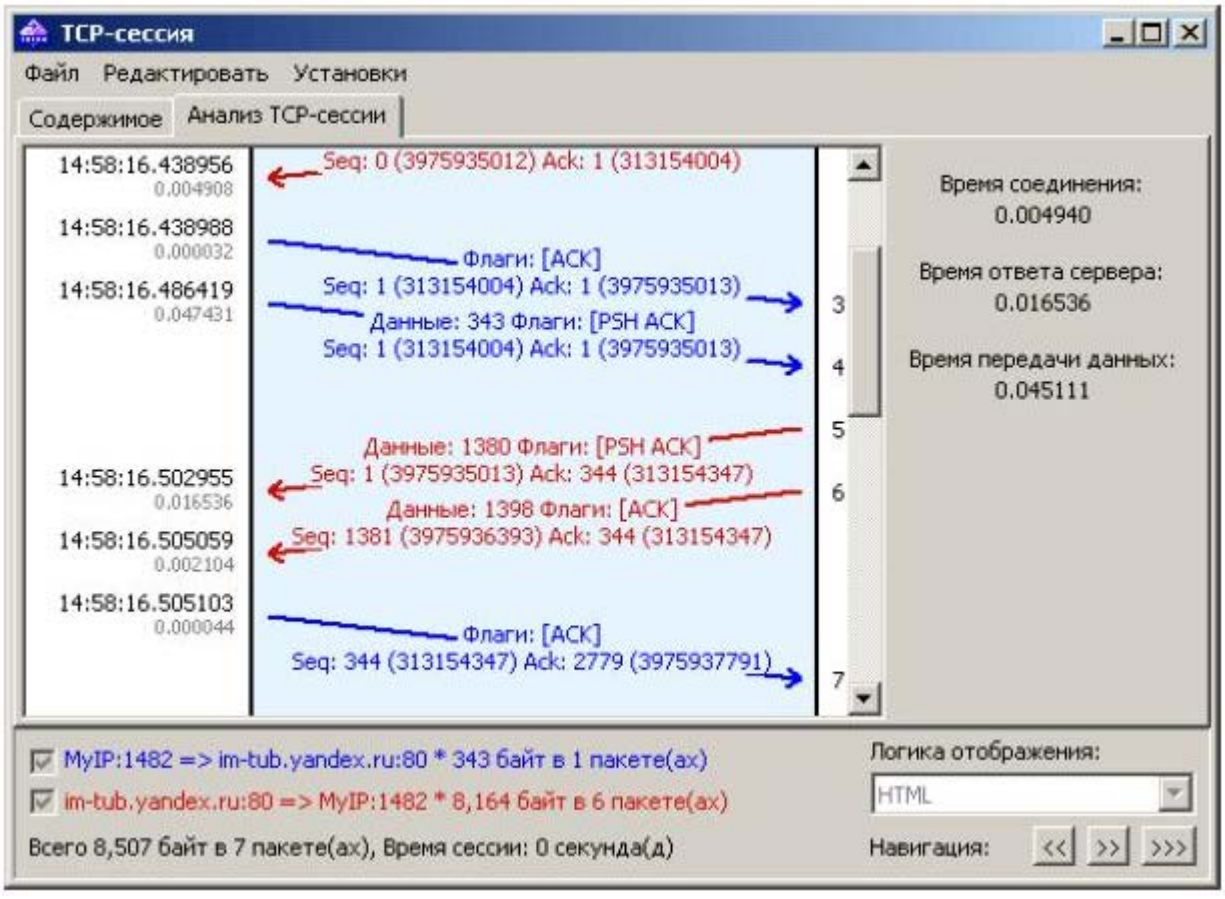

Рисунок 3.6 — Вид восстановленной сессии обмена данными
в окне анализатора пакетов

Тестирование сети происходит следующим образом.

1. Консоль генерирует и передает скрипт теста в Оконечную точку E1.
2. E1 передает копию скрипта в Оконечную точку E2.

3. E1 уведомляет Консоль, что оконечные точки готовы.
4. Консоль дает оконечным точкам указание начать тестирование.
5. Оконечные точки выполняют тест. E1 собирает результаты.
6. E1 возвращает результаты тестирования в Консоль.
7. Консоль отображает результаты в реальном времени (Рисунок 3.4).
8. Анализатор пакетов перехватывает все кадры, осуществляет их анализ и выводит результаты на экран дисплея (Рисунок 3.5).
9. Анализатор пакетов реконструирует TCP-сессию и выводит результат на экран дисплея (Рисунок 3.6).

4. Программа работы

4.1. Изучить стек протоколов TCP/IP, форматы заголовков пакетов и способы обмена данными на канальном, сетевом и транспортном уровнях (выполняется в процессе домашней подготовки).

4.2. Изучить способы доступа к среде в беспроводной локальной компьютерной сети с функцией распределенной координации DCF (выполняется в процессе домашней подготовки).

4.3. Измерить эффективную скорость передачи данных при использовании протоколов TCP и UDP при максимальном размере блока полезной информации.

4.4. Исследовать форматы заголовков кадров на канальном (MAC) уровне, определить тип и подтип кадров, общий размер кадров, направление передачи кадров и рассчитать отношение сигнал/шум и интервалы времени, затрачиваемые на различные типы кадров.

4.5. Исследовать форматы заголовков пакетов на сетевом (IP) уровне, определить длину заголовка, наличие фрагментации и время жизни пакета.

4.6. Исследовать форматы заголовков пакетов на транспортном уровне (TCP/UDP), определить длину заголовка и тип пакета, номера портов источника и назначения, ширину окна и длину поля данных.

4.7. Исследовать последовательность передачи пакетов в TCP-сессии и определить временные интервалы, требуемые для передачи различных типов пакетов.

4.8. Построить временные диаграммы обмена кадрами для фаз установления соединения, передачи данных и завершения соединения.

5. Методика исследований

5.1. Убедиться, что на тестируемых станциях установлены и активированы программные модули *Performance Endpoints*, на консольном компьютере *IxChariot*, а на четвертом компьютере — анализатор протокола *CommView for WiFi.*

5.2. Подать питание на точку доступа. Проверить IP-адреса на тестируемых компьютерах, настроить беспроводную сеть на канал, в котором отсутствуют сторонние станции. Проверить работоспособность беспроводной сети путем взаимного пингования станций и точки доступа.

5.3. Запустить на консольном компьютере программу IxChariot. Выбрать в меню «Создать». Появится окно IxChariot (рисунок 3.4) с чистым рабочим полем.

5.4. Запустить на анализаторе протокола CommView for WiFi. Откроется окно (рисунок 3.5). Выбрать канал сканирования (канал, на котором работают станции лабораторного стенда.

5.5. По п.4.3 программы: Нажать в окне IxChariot Test (рисунок 3.4) кнопку «Add Pair», в отрывшемся окне Edit an Endpoint Pair (рисунок 3.2) задать IP-адреса станции источника, станции назначения и тип протокола. Нажать клавишу Select Script и выбрать вид тестовой последовательности и измеряемый параметр, в частности, измерение пропускной способности (Throuhput).

5.6. Сделать двойной щелчок на клавише Edit This Script и в открывшемся окне Script Editor (рисунок 3.3) установить желаемые параметры передаваемой последовательности: size_of_record_to_send (размер передаваемого блока за запись байтах); transaction_per_record (количество повторений передачи блока за транзакцию) и number_of_timing_records (количество записей за сеанс связи). Эти параметры следует задавать таким образом, чтобы общее количество передаваемых байтов за сеанс связи было не менее 10^7.

5.7. Запустить одновременно захват пакетов на анализаторе протоколов CommView for WiFi и процесс измерения на IxChariot, нажав клавишу Run. Снять численные показания измеренной величины и скопировать для отчета график зависимости пропускной способности от времени измерения.

5.8. Остановить процесс захвата пакетов. Последовательно выделяя пакеты в окне анализатора пакетов (рисунок 3.5) проанализировать содержимое заголовков и произвести отсчет интервалов времени между пакетами в соответствии с пунктами программы исследований 4.3-4.6. Скопировать окна с заголовками для отчета.

5.9. Выбрать в меню анализатора протоколов «Реконструкция TCP сессии» и дождаться появления на экране изображения сессии (рисунок 3.6) и скопировать ее для анализа и отчета. На основании полученных диаграмм определить время, затрачиваемое на передачу пакетов различных типов с учетом межпакетных интервалов.

6. Содержание отчета

Отчет по лабораторной работе должен содержать:

6.1) титульный лист;
6.2) цель работы;
6.3) схему экспериментальной установки и программу работы;
6.4) методику проведения исследований;
6.5) вид окна с последовательностью пакетов, участвующих в обмене;
6.6) распечатки расшифровок заголовков всех типов кадров и пакетов;

6.7) вид фрагмента TCP-сессии;
6.8) численные результаты измерений
6.9) временные диаграммы обмена кадрами и пакетами между станциями и точкой доступа;
6.10) выводы по работе с раскрытием особенностей передачи данных в беспроводных сетях на транспортном уровне.

7. Контрольные вопросы

7.1. Из каких уровней состоит стек TCP/IP-протоколов и какие протоколы используются на различных уровнях?

7.2. Назовите поля заголовков кадров и пакетов MAC-, IP-, TCP- и UDP-протоколов и объясните их назначение.

7.3. С какой целью в начало заголовка кадра физического уровня введена преамбула и какова ее структура?

7.4. По какому признаку точка доступа определяет, что поступающий пакет принадлежит ей?

7.5. Для чего в заголовок MAC-кадра беспроводной сети введены флаги к DS и от DS?

7.6. В каком случае станция назначения, обнаружив в поле заголовка свой адрес, этот кадр не принимает, а в каком принимает?

7.7. Как осуществляется регулирование доступа к беспроводной среде при функции распределенной координации?

7.8. Каким образом задается приоритет доступа к среде?

7.9. Как осуществляется регулирование доступа к беспроводной среде при функции сосредоточенной координации?

7.10. Как происходит процесс установления соединения по протоколу TCP и какие параметры содержатся в передаваемых пакетах?

7.11. Объясните процесс установления соединения на восстановленной анализатором протоколов TCP-сессии.

7.12. Как пользуясь данными анализатора протокола можно определить время обратного отсчета $T_{\text{ВО}}$?

7.13. Как в системе IxChariot рассчитывается пропускная способность звена передачи данных сети?

7.14. Начертите временную диаграмму передачи кадров на канальном уровне от станции отправителя сообщения до станции получателя.

7.15. Начертите временную диаграмму передачи кадров на транспортном уровне с использованием протокола TCP от станции отправителя сообщения до станции получателя.

Библиографический список рекомендуемой литературы

1. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 2-е изд. / В.Г. Олифер, Н.А. Олифер. — СПб: Изд-во "Питер", 2005. — 864 с.
2. Рошан П. Основы построения беспроводных локальных сетей стандарта 802.11.: Пер. с англ. / П. Рошан, Дж. Лиэри. — М.: Издательский дом «Вильямс», 2004. — 304 с.
3. Таненбаум Э. Компьютерные сети: Пер. с англ. / Э.Таненбаум.— СПб.: Изд-во "Питер", 2005. — 672 с.
4. Чернега В.С. Компьютерные сети: Учебное пособие для вузов / В.С. Чернега, Б. Платтнер. — Севастополь: Изд-во СевНТУ, 2006. — 500 с.

Приложение

Типы кадров в сетях IEEE 802.11

Под тип кадра выделено 2 бита, подтип кадра **—** 4 бита.

00 - управление (*managemen*t)

- 0000 - запрос ассоциации от станции к точке доступа;
- 0001 - ответ точки доступа на запрос ассоциации;
- 0010 - запрос повторной ассоциации;
- 0011 - ответ на запрос повторной ассоциации;
- 0100 - пробный запрос;
- 0101 - ответ на пробный запрос;
- 1000 - сигнальный кадр, (beacon);
- 1001 - объявление наличия трафика;
- 1010 - разрыв ассоциации;
- 1011 – аутентификация;
- 1100 - отмена аутентификации.

01 - контроль (*control*, управление потоком)

- 1010 - опрос точки доступа на наличие буферизованного кадра;
- 1011 – RTS;
- 1100 – CTS;
- 1101 – ACK;
- 1110 - конец режима "без состязания" (CF) алгоритма PCF;
- 1111 - подтверждение конца режима CF.

10 - данные (*data*)

- 0000 - просто данные;
- 0001 - данные и подтверждение опроса (только в период CF);
- 0010 - данные и опрос (только в период CF);
- 0011 - данные, подтверждение, опрос (только в период CF);
- 0100 - нет данных (станция в извещает точку доступа о режиме энергосбережения);
- 0101 - подтверждение опроса (только в период CF);
- 0110 - опрос (только в период CF);
- 0111 - подтверждение, опрос (только в период CF).

11 – зарезервирован.

Заказ № _____ от «_____» _____ 2012 г. Тираж 50 экз.