Лабораторная работа Wireshark: Введение

«*Скажи мне, и я забуду. Покажи мне, и я запомню. Дай попробовать самому, и я пойму*».
Китайская пословица

Значительно углубить понимание сетевых протоколов можно, если увидеть их в действии, пронаблюдав за последовательностью сообщений, которыми обмениваются два элемента протокола, если вникнуть в детали работы протокола, заставив его выполнять определенные действия и наблюдать за этими действиями и их результатами. Такое можно осуществить либо с помощью моделируемых сценариев, либо в реальной сетевой среде, такой, как Интернет. В лабораторных работах этого курса вы, используя программу Wireshark, будете запускать сетевые приложения с различными сценариями на вашем компьютере (или на компьютере, одолженном у друзей; сообщите нам, если у вас нет компьютера, где можно запустить Wireshark). Вы будете наблюдать, как сетевые протоколы вашего компьютера взаимодействуют и обмениваются сообщениями с объектами протокола, исполняющегося в другом месте сети Интернет. Таким образом, вы и ваш компьютер будете являться неотъемлемой частью этих «живых» лабораторных работ. Вы будете наблюдать и учиться на собственном опыте.
В этой первой лабораторной работе вы познакомитесь с программой Wireshark и выполните несколько простых действий по захвату пакетов и наблюдению за ними. Основной инструмент для наблюдения за сообщениями, которыми обмениваются элементы исполняемого протокола, называется **анализатор пакетов** (или **сниффер**). Как следует из названия, он анализирует (перехватывает) сообщения, которые отправляются или получаются вашим компьютером; он также обычно сохраняет и/или отображает содержимое различных полей протокола этих перехваченных сообщений. Анализатор пакетов является пассивной программой. Он только следит за сообщениями, отправленными и полученными приложениями и протоколами, запущенными на вашем компьютере, но сам никогда не отправляет пакеты. Полученные пакеты тоже никогда явно не адресуются анализатору. Он просто получает *копию* этих пакетов.
На рис. 1 показана структура анализатора пакетов. В правой части рис.1 находятся протоколы (в данном случае, Интернет-протоколы) и приложения (например, веб-браузер или FTP-клиент), которые обычно работают на вашем компьютере. Анализатор пакетов (в пунктирном прямоугольнике) является дополнением к обычному программному обеспечению вашего компьютера и состоит из двух частей.
Библиотека захвата пакетов получает копию каждого кадра канального уровня, который отправляется или получается компьютером. Вспомним из обсуждения в разделе 1.5 (рис. 1.24), что сообщения, которыми обмениваются протоколы более высокого уровня, такие как HTTP, FTP, TCP, UDP, DNS или IP, в конечном счете, заключены в кадры канального уровня, которые передаются через физический носитель, такой, как кабель Ethernet. На рис. 1 показано предположение, что физическим носителем является Ethernet, и поэтому все протоколы верхних уровней, в конечном счете, инкапсулируются в кадр Ethernet. Захват всех кадров канального уровня, таким образом, дает все сообщения, отправленные/полученные всеми протоколами и приложениями, выполняющимися на вашем компьютере.

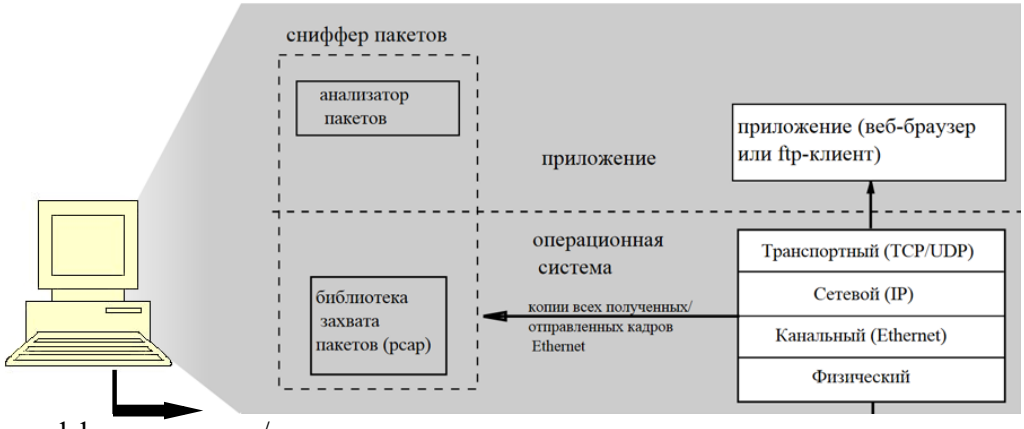

Рис. 1. Структура анализатора пакетов

Вторым компонентом является **анализатор пакетов**, который отображает содержимое всех полей в протокольном сообщении. Чтобы сделать это, анализатор пакетов должен «понимать» структуру всех сообщений, которыми обмениваются протоколы. Например, предположим, что мы хотим отобразить различные поля в сообщениях, которыми обменивается протокол HTTP на Рис. 1. Анализатор пакетов понимает формат Ethernet-кадров, и поэтому может идентифицировать IP-дейтаграммы внутри кадра Ethernet. Он также понимает формат IP-дейтаграммы, так что он может извлечь сегмент TCP из IP-дейтаграммы. И, наконец, он понимает структуру сегмента TCP, поэтому он может извлечь сообщение HTTP, содержащееся в сегменте TCP. Наконец, он понимает протокол HTTP и поэтому, например, знает, что первые байты сообщения HTTP будут содержать строку GET, POST или HEAD, как показано на рис. 2.8 в тексте.
Мы будем использовать анализатор пакетов Wireshark [**wireshark.org**] в этих лабораторных работах, который позволит нам отображать содержимое сообщений, переданных/полученных протоколами на разных уровнях стека протоколов. (С технической точки зрения, Wireshark – это анализатор пакетов, который использует библиотеку захвата пакетов в вашем компьютере). Это бесплатная программа, которая поддерживает работу в операционных системах Windows, Linux/Unix и OS X. Это идеальный анализатор для наших лабораторных – он стабилен, имеет большую базу пользователей и хорошо документированную поддержку, которая включает в себя руководство пользователя (**wireshark.org/docs/wsug_html_chunked/**), страницы электронного руководства (**wireshark.org/docs/man-pages/**) и подробный список часто задаваемых вопросов (**wireshark.org/faq.html**), богатый функционал, который включает в себя возможность анализировать сотни протоколов, и хорошо продуманный пользовательский интерфейс. Он работает в компьютерах, используя протоколы Ethernet, PPP и SLIP, 802,11 и многие другие технологии канального уровня (если среда, в которой он работает, позволяет Wireshark это делать).

Загрузка Wireshark

Чтобы запустить Wireshark, вам нужен компьютер, который поддерживает как Wireshark, так и одну из библиотек – libpcap или WinPCap. Библиотека libpcap, если она еще не присутствует в вашей операционной системе, устанавливается вместе с Wireshark. Список поддерживаемых операционных систем представлен на странице загрузки **wireshark.org/download.html**.

Для загрузки и установки Wireshark:

1. Перейдите по ссылке **wireshark.org/download.html**.
2. Загрузите установочный файл для вашей системы и установите Wireshark на компьютер.

Если у вас возникают сложности с установкой и запуском Wireshark, обратитесь к разделу Wireshark FAQ и вы найдете много полезной информации.

Запуск Wireshark

При запуске программы Wireshark, вы увидите главное окно, как показано ниже:

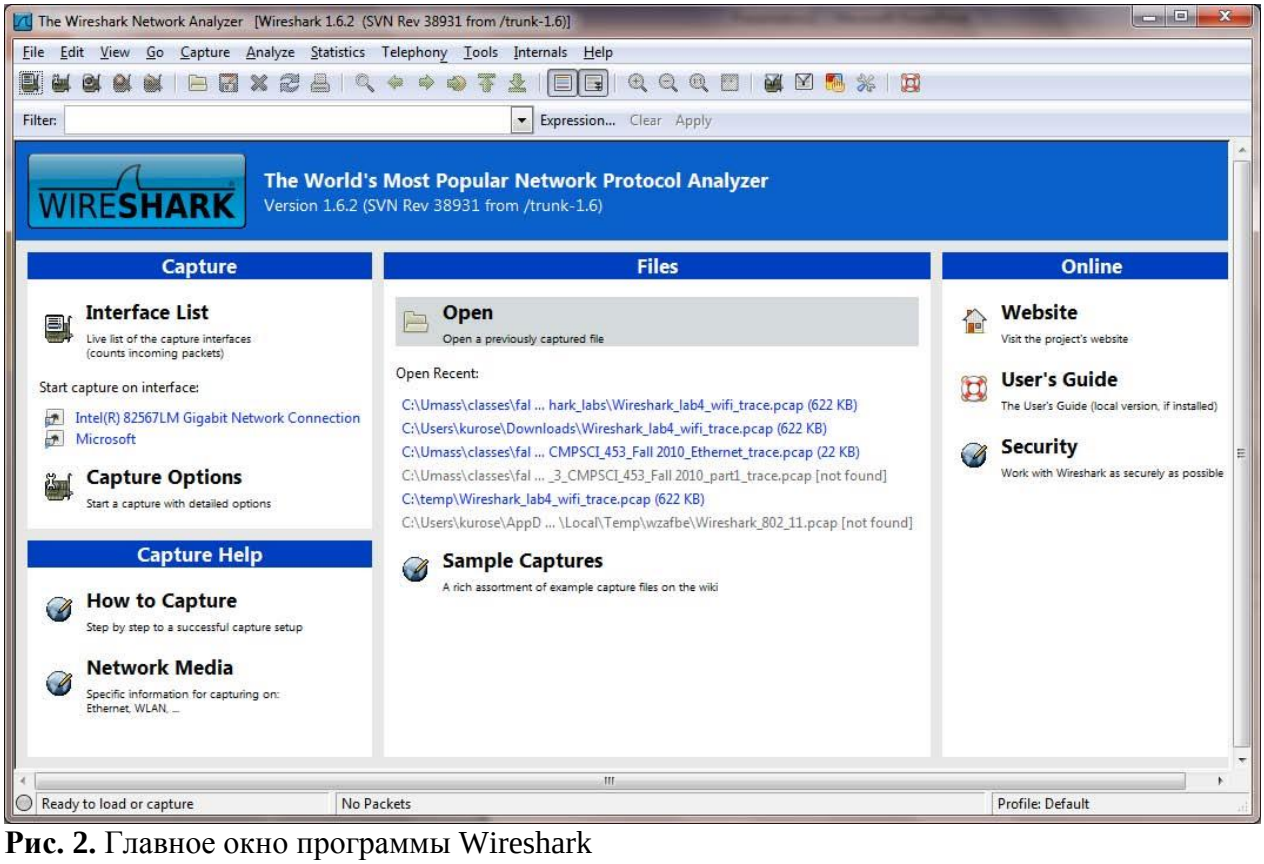

Рис. 2. Главное окно программы Wireshark

В левой верхней части окна вы увидите список интерфейсов (**Interface list**), в котором представлены все имеющиеся на вашем компьютере сетевые интерфейсы. После того, как вы выберете интерфейс, Wireshark будет перехватывать все пакеты, проходящие через него. В примере выше мы видим два интерфейса: Ethernet-интерфейс (**Gigabit network Connection**) и беспроводной интерфейс (**Microsoft**).

Если вы выберете один из интерфейсов, чтобы начать перехват пакетов (то есть дадите команду для Wireshark начать перехват пакетов на этом интерфейсе), появится окно (подобное тому, что вы видите ниже), показывающее информацию о перехваченных пакетах. Остановить захват пакетов вы можете, используя команду **Stop** (Стоп) в меню **Capture** (Захват).

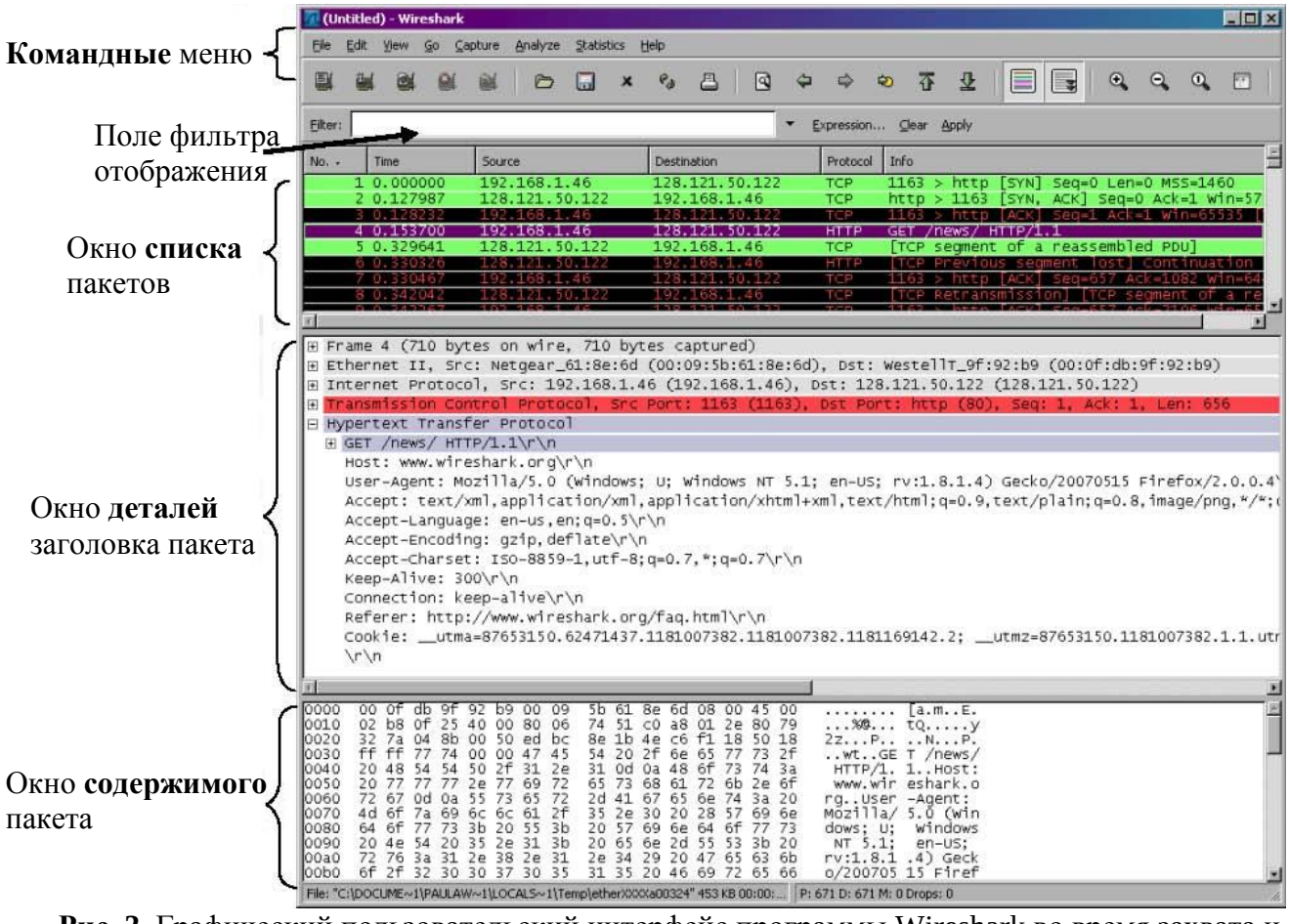

Рис. 3. Графический пользовательский интерфейс программы Wireshark во время захвата и анализа пакетов

Интерфейс Wireshark содержит пять основных областей:

- **Командные меню** представляет собой стандартные раскрывающиеся меню, расположенные вверху окна. Сейчас нас интересуют меню **File** (Файл) и **Capture** (Захват). Меню **File** (Файл) предназначено для сохранения захваченных пакетов, для открытия файла с уже сохраненными данными пакетов, а также для выхода из программы. Команды в меню **Capture** (Захват) позволяют начать захват пакетов.
- **Окно списка пакетов** отображает построчно информацию по каждому захваченному пакету, включая номер пакета (присваивается здесь в программе, а не содержится ни в каком заголовке) время, когда пакет был перехвачен, адреса источника и приемника, тип протокола а также специальную информацию, относящуюся к протоколу. Список пакетов можно отсортировать по любому из этих полей простым нажатием на имя соответствующего столбца. В поле тип протокола отображается самый верхний уровень протокола, то есть протокол, являющийся либо исходным, либо конечным для конкретного пакета.
- В **окне деталей заголовка пакета** отображается подробная информация о пакете, выбранном в предыдущем окне (строка с эти пакетом подсвечена). (Чтобы выбрать пакет в окне списка, просто наведите указатель мыши на соответствующую строку и нажмите левую кнопку мыши). Сюда включена информация о кадре Ethernet (полагаем, что пакет проходил через интерфейс Ethernet) и IP-дейтаграмме, содержащейся в пакете. Объем отображаемой информации в этом окне можно

уменьшать или увеличивать, сворачивая или разворачивая группу строк, используя значки плюс минус слева в строке.

- **Окно содержимого пакета** отображает все, что содержится в захваченном пакете, в шестнадцатеричном формате и в формате ASCII.
- Вверху графического окна пользователя, непосредственно под командным меню находится **поле фильтра отображения**, в которое может быть введено имя протокола или что-то еще, чтобы отфильтровать информацию, отображаемую в окне списка пакетов (и, следовательно, в двух следующих за ним окнах). В приведенном ниже примере, мы будем использовать это поле, чтобы Wireshark скрыл (не отображал) все пакеты, кроме тех, которые соответствуют сообщениям протокола HTTP.

Пробный запуск Wireshark

Лучший способ для изучения нового программного обеспечения – попробовать его в деле! Мы будем считать, что ваш компьютер подключен к Интернету через проводной интерфейс Ethernet. Мы рекомендуем вам для первой лабораторной работы использовать именно на Ethernet-соединение, а не беспроводную связь. Выполните следующее:

1. Запустите ваш любимый браузер, и в нем откроется домашняя страница.
2. Запустите программу Wireshark. Вы увидите начальное окно, показанное на рис.2. Программа еще не начала захватывать пакеты.
   Чтобы начать работу, выберите в меню **Capture** (Захват) команду **Interfaces** (Интерфейсы). Откроется окно **Wireshark: Capture Interfaces** (Wireshark: Интерфейсы для захвата), показанное на рис. 4.

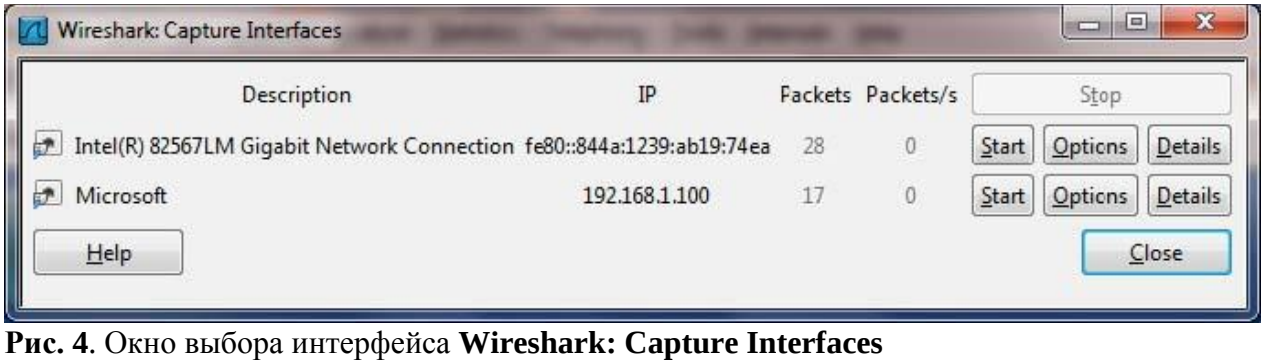

Рис. 4. Окно выбора интерфейса **Wireshark: Capture Interfaces**

Вы увидите список всех интерфейсов вашего компьютера, а также текущее число прошедших через интерфейсы пакетов. Нажмите кнопку **Start** (Запуск) рядом с тем интерфейсом, который хотите анализировать (в нашем случае Gigabit Network Connection). Начнется захват пакетов – программа Wireshark теперь перехватывает все пакеты, полученные или отправленные вашим компьютером!
Как только вы начнете захват пакетов, появится окно, подобное показанному на рис. 3. В нем отображаются перехваченные пакеты. Выбрав в меню **Capture** (Захват) команду **Stop** (Стоп), вы можете остановить захват пакетов. Но не останавливайте пока процесс. Давайте перехватим что-нибудь интересное. Чтобы сделать это, мы должны будем воспроизвести сетевой трафик. Воспользуемся веб-браузером, который использует протокол HTTP, который мы будем детально изучать, чтобы загрузить контент с веб-сайта.

Не завершая работу Wireshark, введите в браузере адрес **http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html**.
После того, как ваш браузер отобразил страницу INTRO-wireshark-file1.html (строка с поздравлением), остановите захват пакетов, выбрав в меню **Capture** (Захват) команду **Stop** (Стоп). Окно Wireshark теперь должно выглядеть так же, как показано на рис. 3. Теперь у вас есть реальные данные по пакетам, которыми обменивался ваш компьютер с другим объектом сети! HTTP-сообщения обмена с веб-сервером gaia.cs.umass.edu должны быть где-то в списке захваченных пакетов. Но там присутствует также множество других типов пакетов (видите различные типы в поле **Protocol** (Протокол) на рис. 3). Даже если кроме загрузки веб-страницы вы больше ничего не делали, все равно на вашем компьютере работает множество других протоколов, скрытых с глаз. Мы поговорим о них позднее, а пока нужно просто помнить, что в сети происходит всегда гораздо больше событий, чем заметно наглядно!
Для того чтобы отобразить страницу, ваш браузер связывается с HTTP-сервером по адресу gaia.cs.umass.edu и обменивается HTTP-сообщениями с сервером, чтобы загрузить эту страницу, как описано в разделе 2.2. Кадры Ethernet, содержащие эти HTTP-сообщения (а также все другие кадры, проходящие через адаптер Ethernet) будут перехвачены программой Wireshark.

3. Укажите значение **http** (все имена протоколов в Wireshark пишутся в нижнем регистре) в поле фильтра отображения. Затем нажмите кнопку **Apply** (Применить) (справа от этого поля). Это приведет к тому, что в окне списка пакетов будут отображаться только HTTP-сообщения.
4. Найдите сообщение GET протокола HTTP, отправленное с вашего компьютера на HTTP-сервер gaia.cs.umass.edu (ищите его в окне списка захваченных пакетов (см. рис. 3)), содержащее также введенный вами адрес gaia.cs.umass.edu. Когда вы выделите найденную строку с сообщением HTTP GET[1], то в окне деталей заголовков появится информация по заголовкам кадра Ethernet, IP-дейтаграммы, сегмента TCP и сообщения HTTP 2. Пользуясь кнопками + и - в левой части окна, вы можете по желанию сворачивать или разворачивать строки. Сверните, например, информацию о кадре и протоколах Ethernet, IP и TCP, а развернутой оставьте ту, что относится к протоколу HTTP. Теперь окно вашей программы Wireshark должно выглядеть примерно так, как показано на рис. 5.
5. Завершите работу Wireshark.

Поздравляем! Ваша первая лабораторная работа завершена.

[1] Напомним, что GET-сообщение протокола HTTP, которое передается на веб-сервер gaia.cs.umass.edu, содержится в сегменте TCP, который, в свою очередь, находится (инкапсулирован) в IP-дейтаграмме, которая инкапсулируется в кадр Ethernet. Еще раз воспроизвести в памяти процесс инкапсуляции, вы можете, обратившись к разделу 1.5 книги.

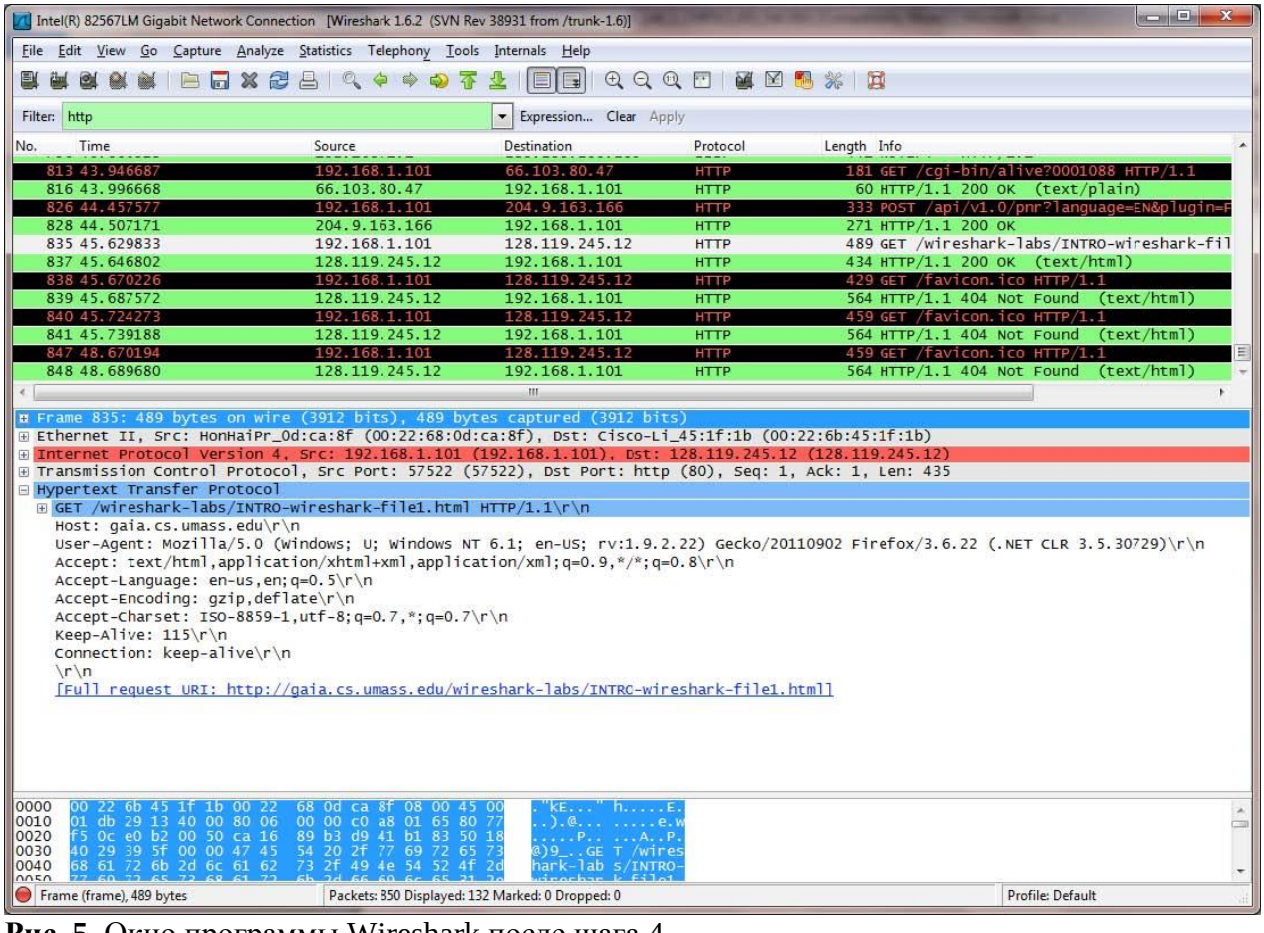

Рис. 5. Окно программы Wireshark после шага 4

Дополнительные задания

Цель этой первой лабораторной работы, в первую очередь, заключалась в том, чтобы познакомить вас с программой Wireshark. После ответа на следующие вопросы можно сделать вывод, что вы успешно скачали, установили и запустили Wireshark и исследовали некоторые из его возможностей. Ответьте на вопросы на основании ваших экспериментов с программой:

1. Перечислите любые 3 протокола, которые могут быть отображены в столбце **Protocol** (Протокол) при отключенном фильтре пакетов и показанном на рис. 3. Сколько времени прошло от момента отправки сообщения GET протокола HTTP до получения ответного сообщения OK? (По умолчанию, значения поля **Time** (Время) в окне списка представляет собой время в секундах от начала трассировки. Вы можете поменять вид этого поля по вашему желанию, выбрав в меню **View** (Вид) пункт **Time Display Format** (Формат отображения времени) и затем указав подходящее представление времени.)
2. Какой IP-адрес у сервера gaia.cs.umass.edu (также известного как wwwnet.cs.umass.edu)? Каков адрес вашего компьютера?
3. Распечатайте сообщения протокола HTTP (GET и OK), полученные вами при ответе на предыдущий вопрос. Для этого выберите команду меню File ⇒ Print (Файл ⇒ Печать), установите переключатели в положение **Selected Packet Only** (Только выбранный пакет) и **Print as displayed** (Печатать в формате отображения), соответственно, и затем нажмите кнопку **OK**.