УДК 004.7

ВОССТАНОВЛЕНИЕ МОМЕНТНЫХ ХАРАКТЕРИСТИК РАСПРЕДЕЛЕНИЯ ИНТЕРВАЛОВ МЕЖДУ ПАКЕТАМИ ВХОДЯЩЕГО ТРАФИКА

Горелов Г.А., Тарасов В.Н., Ушаков Ю.А.

В статье описывается подход к восстановлению моментных характеристик распределения интервалов между пакетами входящего трафика. Эти характеристики позволяют анализировать трафик методами теории массового обслуживания.

Ключевые слова: моментные характеристики, распределение трафика, анализатор трафика, программа Wireshar.

Введение

Как известно, теория массового обслуживания (ТМО) опирается на распределения интервалов между заявками входного потока и времени обслуживания. На практике распознавание закона распределения интервалов вызывает большие проблемы и к тому же трафик как случайный процесс имеет свойство постоянно меняться. Поэтому целесообразнее использование числовых характеристик распределения интервалов между пакетами. В данной работе для их определения предлагается использовать программу Wireshark.

Описание программы Wireshark

Wireshark (ранее Ethereal) – программа-анализатор трафика для компьютерных сетей технологии Ethernet и некоторых других, имеющий графический пользовательский интерфейс. В июне 2006 года проект был переименован в Wireshark из-за проблем с торговой маркой [1].

Функциональность, которую предоставляет Wireshark, очень схожа с возможностями программы tcpdump, однако Wireshark имеет графический пользовательский интерфейс и гораздо больше возможностей по сортировке и фильтрации информации. Программа позволяет пользователю просматривать весь проходящий по сети трафик в режиме реального времени, переводя сетевую карту в неразборчивый режим (от англ. promiscuous mode – см. рис. 1).

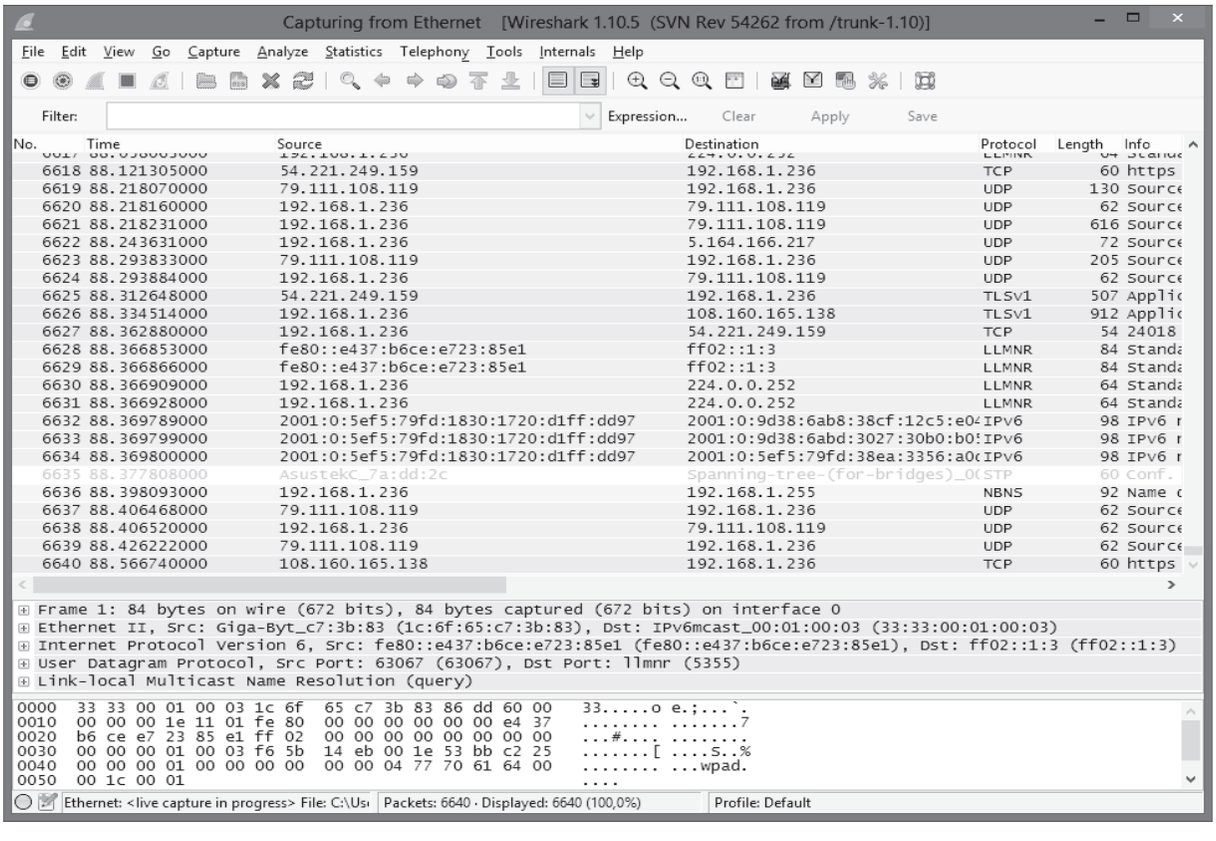

Рис. 1. Захват сетевого трафика программой Wireshark

Wireshark – это приложение, которое «видит» структуру самых различных сетевых протоколов и поэтому позволяет разобрать сетевой пакет, отображая значение каждого поля протокола любого уровня. Поскольку для захвата пакетов используется библиотека Pcap, существует возможность захвата данных только из тех сетей, которые поддерживаются этой библиотекой. Тем не менее программа Wireshark умеет работать с множеством форматов входных данных, соответственно, можно открывать файлы данных, захваченных другими программами, что расширяет возможности захвата.

Возможности программы включают:

- глубокий анализ сотни протоколов с регулярным добавлением новых;
- захват сетевого трафика в реальном времени с последующим анализом в любое удобное время;
- standard three-pane packet browser (стандартный пакетный браузер с тремя областями);
- кроссплатформенность: существуют версии для большинства типов UNIX, в том числе Linux, Solaris, FreeBSD, NetBSD, OpenBSD, Mac OS X, а также для Windows;
- захваченная информация по сети может быть просмотрена с помощью графического интерфейса пользователя или с помощью TTY-режима утилиты TShark;
- самые мощные возможности по сортировке и фильтрации информации в отрасли;
- обширные возможности по VoIP анализу;
- чтение/запись большого количества форматов файлов захвата: tcpdump (libpcap), Pcap NG, Catapult DCT2000, Cisco Secure IDS iplog, Microsoft Network Monitor, Network General Sniffer® (compressed and uncompressed), Sniffer® Pro, and NetXray®, Network Instruments Observer, NetScreen snoop, Novell LANalyzer, RAD-COM WAN/LAN Analyzer, Shomiti/Finisar Surveyor, Tektronix K12xx, Visual Net-works Visual UpTime, WildPackets Ether-Peek/TokenPeek/AiroPeek и многие другие;
- файлы захвата, сжатые с помощью gzip, могут быть распакованы сразу;
- захват данных в реальном времени может быть произведен с Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI и других (в зависимости от платформы);
- поддержка расшифровки для многих протоколов, включая IPsec, ISAKMP, Kerberos, SNMPv3, SSL/TLS, WEP, and WPA/WPA2;
- правила выделения цветом могут быть применены для списка пакетов для быстрого, интуитивного анализа;
- выходные данные могут быть экспортированы в XML, PostScript®, CSV или обычный текст.

Одним из форматов экспорта данных, удобных для просмотра, является CSV (см. рис. 2). Такой файл можно открыть в любом текстовом редакторе или редакторе табличных данных для последующего анализа и расчета характеристик.

Тем не менее трафик может быть настолько интенсивным, что обрабатывать его данные даже в табличных редакторах становится проблематично, не говоря уже о том, что и сами данные по трафику могут храниться не в одном файле. В данной статье нами рассмотрено программное решение для расчета моментных характеристик интервалов поступления пакетов. Главным преимуществом данного анализатора является его работа в малых масштабах времени (микросекунды), в отличие от той же программы NetFlow Analyzer, которая фиксирует пакеты с поминутной дискретизацией.

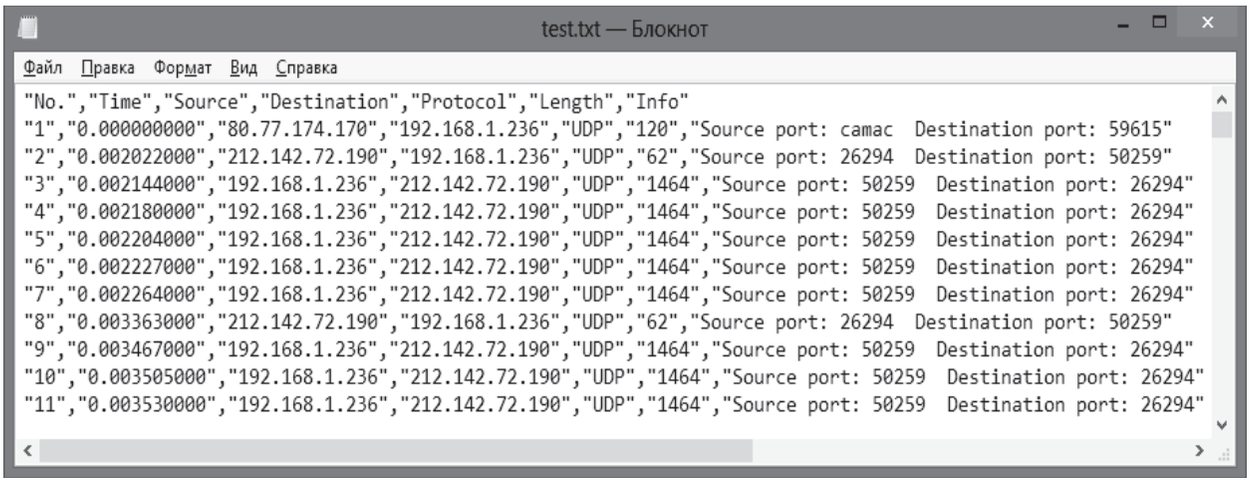

Рис. 2. Пример данных, экспортированных в формат CSV.

Определение моментных характеристик интервалов поступления пакетов

Разработанная программа в дополнение к анализатору позволяет осуществлять выборку моментов времен поступления пакетов, вычленив входящий трафик из всего набора данных, полученных программой Wireshark. Далее с использованием известных формул математической статистики определяются моментные характеристики распределения временных интервалов. В работе использованы статистики до третьего порядка, которые позволяют судить о характере распределения интервалов. Например, коэффициент вариации показывает отличие трафика от пуассоновского потока и совместно с асимметрией позволяет судить о степени весомости хвостов распределений.

Среднее значение интервала между соседними пакетами $\bar{\tau} = \frac{1}{N}\sum_{k=0}^{N}\left(t_{k+1} - t_k\right)$, где t_k – моменты времени поступления пакетов, N – число анализируемых интервалов. Выборочная дисперсия $D_в = \overline{t^2} - \bar{\tau}^2$, где $\overline{t^2} = \frac{1}{N}\sum_{k=0}^{N}\left(t_{k+1} - t_k\right)^2$ – второй начальный момент. Коэффициент вариации $c = \sigma_в / \bar{\tau}$, где $\sigma_в = \sqrt{D_в}$. Асимметрия $A_s = \overline{t^3} / \sigma_в^3$ где $\overline{t^3} = \frac{1}{N}\sum_{k=0}^{N}\left(t_{k+1} - t_k\right)^3$.

Если большой массив данных разбит на несколько блоков, то по указанным формулам определяются групповые средние, а затем их средние арифметические значения.

Программа анализа временных данных

Для расчета моментных характеристик была разработана программа, которая из всего входного файла с данными о захвате сетевого трафика выбирает только данные, относящиеся к поступившим пакетам, и для них производит расчет интервалов и их моментных характеристик.

Возможности программы включают:

- выборку данных о времени поступления пакетов на указанный хост;
- расчет интервалов времени между поступившими пакетами;
- расчет моментных характеристик для интервалов поступивших пакетов;
- сохранение данных о времени поступления пакетов в двоичном и текстовом формате;
- сохранение данных об интервалах между поступлениями пакетов в двоичном и текстовом формате;
- вывод и сохранение моментных характеристик в текстовом формате;

Программа обрабатывает текстовые файлы, содержащие данные в виде, представленном на рис. 2, или аналогичном. В программе разработаны два класса (в терминах объектно ориентированного программирования):

- TrafficLogParams – хранит данные о временах приходов пакетов, их интервалов и рассчитывает моментные характеристики. Также предоставляет методы сохранения данных в файлы и их загрузки из файлов;
- LogParser – статический класс, производящий анализ входного файла и добавление данных в класс TrafficLogParams.

Главный метод класса LogParser получает в качестве входных данных имя файла и IP-адрес хоста. Каждая строка исходного файла обрабатывается и из нее выбираются данные о времени и два IP-адреса – адрес отправителя и адрес получателя. Если поле получателя совпадает с IP-адресом хоста, тогда время поступления пакета добавляется в массив времен поступления пакетов класса TrafficLogParams.

```
public static TrafficLogParams TextFilePars-
er(string fileName, string ip, bool isIncom-
ing)
{
TrafficLogParams log = new TrafficLogPa-
rams();
StreamReader file = new StreamReader (file-
Name);
string[] currentLine;
int lineNumber = 0;
int ipIndex;
if (isIncoming)
       ipIndex = 2;
else
       ipIndex = 1;
while (!file.EndOfStream)
{
       currentLine = GetDataArray
       (file.ReadLine().Trim());
       lineNumber++;
       try
       {
       if (MinimizeIp (currentLine[ipIndex])
== MinimizeIp (ip))
       {
       log.AddTime(ParseDouble(currentLine
       [0]));
       }
       }
       catch (FormatException ex)
       {
       MessageBox.Show(string.Format("{0}\nC
трока = {1}", ex.Message, lineNumber));
```

```
        }
}
file.Close();
return log;
}
```

Второй по важности метод класса LogParser разбивает входную строку на составные элементы, проверяя каждый элемент на принадлежность формату времени или IP-адресу, и возвращает их в виде массива.

```
private static string[] GetDataArray(string
input)
{
string[] data = new string[3];
string currentValue = "";
int symbolIndex = 0;
int valueIndex = 0;
while (symbolIndex < input.Length && va-
lueIndex < 3)
{
       while (symbolIndex < input.Length &&
(char.IsDigit(input[symbolIndex]) || IsSepa-
rator(input[symbolIndex])))
       {
       currentValue += input[symbolIndex];
       symbolIndex++;
       }
       if (currentValue != "")
       {
              if ((IsDouble(currentValue) ||
              IsIp(currentValue)))
       {
       data[valueIndex] = currentValue;
       valueIndex++;
       }
       currentValue = "";
       if (valueIndex >= 3)
       {
       symbolIndex = input.Length;
       }
       }
       while (symbolIndex < input.Length &&
!char.IsDigit(input[symbolIndex]) &&
       !IsSeparator(input[symbolIndex]))
       {
       symbolIndex++;
       }
}
return data;
}
```

Метод проверяет входной символ на принадлежность символу-разделителю: «.» или «,». Такая проверка важна только для данных о времени, так как в некоторых странах дробная часть отделяется запятой (например в России), а не точкой. Именно по этой причине при преобразовании строкового представления числа в эквивалентное ему вещественное число, обозначающее время, используется не стандартный метод языка программирования, а его модификация, определяющая региональные параметры.

```
private static double ParseDouble(string
value)
{
if (CultureInfo.CurrentCulture
.NumberFormat.NumberDecimalSeparator == ".")
{
       value = value.Replace(',', '.');
       }
       else
       {
       value = value.Replace('.', ',');
       }
return double.Parse(value);
}
```

При сопоставлении IP-адреса хоста с IP-адресом в текущей строке лога-файла производится минимизация IP-адресов до общего вида. Другими словами, IP-адрес 010.014.000.011 будет равен 10.14.0.11.

Результаты

Авторской программой был проанализирован файл с данными о трафике, поступающем на прокси-сервер вуза почти за час съема. Входной файл содержал более 2150000 строк, обработка вручную которых не представляется возможным. Были получены следующие результаты (рис. 3):

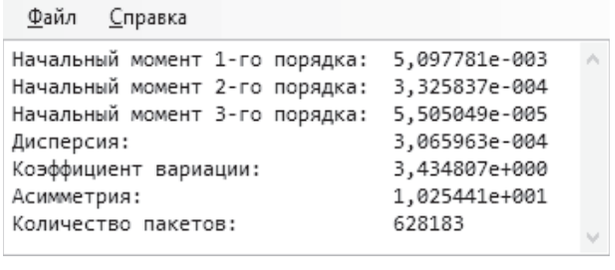

Рис. 3. Результат работы программы анализа лог-файлов

Заключение

Полученные данные свидетельствуют о том, что анализируемый трафик сильно отличается от пуассоновского (коэффициент вариации c = 3,43 вместо 1), значение асимметрии равной A_s = 10,25 говорит о том, что распределение интервалов между пакетами трафика относится к распределениям с тяжелыми хвостами. Например, у экспоненциального закона A_s = 2. Для расчета характеристик такого трафика требуется соответствующий математический аппарат.

Литература

1. Wireshark official web-site URL: http://www.wireshark.org/ [02.02.2014]

RESTORING MOMENT DISTRIBUTION CHARACTERISTICS INTERVAL BETWEEN PACKETS OF INCOMING TRAFFIC

Gorelov G.A., Tarasov V.N., Ushakov Y.A.

The paper describes an approach to restore torque characteristics of the distribution of intervals between bursts of incoming traffic. These characteristics allow the methods to analyze the traffic queuing theory.

Keywords: *torque characteristics, traffic, traffic analyzer, the program Wireshar.*

Тарасов Вениамин Николаевич, д.т.н., профессор, заведующий Кафедрой программного обеспечения и управления в технических системах (ПОУТС) Поволжского государственного университета телекоммуникаций и информатики (ПГУТИ). Тел. (8-846) 228-00-13; 8-960-827-22-33. E-mail: vt@ist.psati.ru.

Ушаков Юрий Александрович, к.т.н., доцент Кафедры системного анализа и управления Оренбургского государственного университета. Тел. 8-922-536-40-81. E-mail: unpk@mail.ru.

Горелов Глеб Александрович, ассистент Кафедры ПОУТС ПГУТИ. Тел. (8-846) 228-00-13, 8-927-721-89-79. E-mail: gleb_fox@bk.ru.

УДК 004.728.3

МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ МАРШРУТИЗАТОРА В СЕТИ В УСЛОВИЯХ ОГРАНИЧЕННОЙ НАДЕЖНОСТИ КАНАЛОВ СВЯЗИ

Макаренко С.И., Михайлов Р.Л.

В статье представлена модель процесса функционирования маршрутизатора, произведена оценка влияния отказов каналов связи на его надежность по показателю вероятности нахождения в работоспособном состоянии. Исследованы показатели надежности сети с различными протоколами маршрутизации в зависимости от количества каналов связи с ограниченной надежностью и интенсивности их отказов.

Ключевые слова: модель маршрутизатора, устойчивость сети, надежность сети, протоколы маршрутизации.

Введение

С развитием телекоммуникационных технологий и увеличением структурной сложности сетей связи актуализируются вопросы обеспечения устойчивости связи к отказам оборудования, а также к различного рода деструктивным воздействиям (ДВ), что подтверждается исследованиями [1-2]. На сетевом уровне модели взаимодействия открытых систем данную задачу решают маршрутизаторы – устройства, организующие передачу информационных пакетов по сети связи в соответствии с протоколами маршрутизации.

Анализ публикаций, находящихся в открытом доступе, показал, что вопросы разработки моделей функционирования маршрутизаторов решались в [3-19]. В [3-8] разработаны модели маршрутизатора, позволяющие оценить влияние структуры трафика на процесс его функционирования. Исследованию модели маршрутизатора как системы массового обслуживания (СМО) и оценке влияния параметров его буфера на процесс функционирования посвящены работы [9-13]. Вопросы влияния алгоритмов распределения информационных потоков рассмотрены в работах [14-17].

Схожая модель маршрутизатора рассмотрена в [18-19], однако авторами этих работ исследовались вопросы разработки метода повышения непрерывности функционирования сети связи за счет рациональной маршрутизации служебной информации, без учета временных параметров работы протоколов маршрутизации. При моделировании рассматривались условия скачкообразного изменения интенсивности потоков информации, но не учитывалась возможность отказов каналов связи (КС).

Таким образом, оригинальность предлагаемой модели функционирования маршрутизатора в сети определяется новыми рамками исследования – условием ограниченной надежности каналов связи.

Модель процесса функционирования маршрутизатора

Модель процесса функционирования маршрутизатора можно представить в виде процесса перехода между различными состояниями, параметры которого определяются вероятностью отказа $P_{отк}(t)$ и веро-