АНАЛИЗ IP-ТРАФИКА МЕТОДАМИ DATA MINING
Проблема кластеризации

Н. Г. Щербакова

Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

УДК 681.324

Представлен подход к решению задачи классификации IP-трафика, основанный на методах интеллектуального анализа данных. С использованием статистических параметров потоков, извлекаемых из характеристик, не зависящих от полезной нагрузки IP-пакетов, проводится идентификация сетевых приложений. Для решения задачи применяется анализ данных, обеспечивающий автоматическое выявление скрытых закономерностей. Рассмотрен ряд алгоритмов кластеризации, проведен их сравнительный анализ.

Ключевые слова: классификация IP-трафика, P2P-трафик, методы машинного обучения, кластеризация, эффективность.

The methodology of IP traffic classification based on the intellectual data analysis is introduced. The identification of network applications is based on statistical flow characteristics derived from payload-independent features. Data mining techniques is used for automatic extraction of hidden patterns. The set of clustering algorithms are examined. The comparison of the algorithms is presented.

Key words: IP traffic classification, Peer-to-Peer traffic, machine learning technique, clustering, efficiency.

1. Проблема и исследуемые данные. Настоящая работа является продолжением работы [1], посвященной исследованию методов Data Mining [2] применительно к анализу трафика. Рассматриваются алгоритмы из области машинного обучения (machine learning) [3]. В рассматриваемой серии работ для анализа трафика применяются алгоритмы кластеризации. Задача кластеризации состоит в следующем. Имеется множество исследуемых объектов $X = x_1, x_2, \ldots, x_n$, каждый из которых характеризуется набором переменных $X_j = a_1, a_2, \ldots, a_m$. Каждая переменная a_i принимает значение из некоторого множества $A_i = \{a_{i1}, a_{i2}, \ldots\}$. Необходимо разбить множество X на группы (кластеры), т. е. построить множество C=$\{c_1,\ c_2,\ \ldots,c_k\}$, где c_i — кластер, содержащий сходные объекты из множества Xотносительно некоторой меры близости.

Исходными данными для исследования являются IP-пакеты (или заголовки пакетов), собранные в точках наблюдения. Единицей рассмотрения является поток — последовательность IP-пакетов, агрегируемых согласно определенным правилам, например двунаправленная или однонаправленная последовательность пакетов между двумя IP-адресами, полная TCP-сессия или однонаправленная последовательность IP-пакетов, определяемая на основе пяти полей заголовка

$<src\_ip,\ src\_port,\ dst\_ip,\ dst\_port,\ protocol>$

и правил формирования, по которым определяется завершение потока (обычно тайм-аут, когда стороны неактивны, или флаг "END" в заголовке пакета). Здесь *src_ip* — IP-адрес источника; *src_port* — порт источника; *dst_ip* — IP-адрес назначения; *dst_port* — порт назначения; *protocol* — транспортный протокол. В качестве протоколов транспортного уровня рассматриваются TCP и UDP. Предполагается, что для классификатора доступна только информация, содержащаяся в заголовках пакетов. Информация, содержащаяся в полезной нагрузке (payload), если она доступна, используется только с целью проверки правильности классификации.

Предположим, требуется определить категории трафика, каждая из которых соответствует приложению или набору приложений. Фиксируется множество переменных (атрибутов), основанных на статистических характеристиках, таких как размер пакетов или интервалы между пакетами, и характеристиках, извлекаемых из заголовков пакетов, таких как размер TCP-сегментов или количество повторных передач. Потоку ставится в соответствие набор значений атрибутов, согласно которым проводится кластеризация. В результате желательно получить кластеры, так чтобы каждому кластеру соответствовала доминантная категория. Можно предположить, что количество кластеров должно быть равно количеству категорий. Однако одно и то же приложение может иметь разные функции распределения параметров, в этом случае потоки попадут в разные кластеры. Таким образом можно получить более детальное представление о приложениях. Например, Web-трафик может использоваться для разных целей: передача больших объемов информации, интерактивный обмен, потоковая передача. Заметим, что чрезмерно большое количество кластеров истощает вычислительные ресурсы и плохо воспринимается. Количество кластеров обычно является параметром алгоритма кластеризации.

Как правило, для определения эффективности алгоритма используются следующие основные метрики: FP (false positive) — доля трафика, приписанного классу X, но не принадлежащего X, по отношению к мощности X; FN (false negative) — доля трафика, принадлежащего классу X, но не приписанного классу X; TP (true positive) — количество правильно классифицированных единиц класса. При определении эффективности единицами рассмотрения могут быть потоки, пакеты или байты.

При описании алгоритмов и статистических характеристик используются термины теории вероятностей и математической статистики, определения которых приведены в [4].

2. Кластеризация трафика на основе EM-алгоритма. Одной из первых работ, посвященных классификации трафика с применением техники машинного обучения, является работа [5], в которой приведена методология разделения потоков трафика на группы с одним типом поведения по отношению к нагрузке на сеть: передача больших объемов данных, одна транзакция в процессе обмена, несколько транзакций и т. д. Задача идентификации конкретных приложений не ставилась, так как в рамках одного приложения можно проследить различные типы поведения, и наоборот, разные протоколы, например HTTP и FTP, могут иметь подобные характеристики.

Исследуются полные двунаправленные потоки, ограниченные только временем наблюдения и сформированные на основе общедоступных наборов пакетов waikato Internet traffic storage (WITS) (www.wand.net.nz/wits/index.php). Рассматриваются следующие атрибуты потока: статистика размеров пакетов (минимальная, максимальная, квартили, отношение минимума к максимуму, первые пять мод); статистика временных интервалов между пакетами; количество байтов; длительность взаимодействия; количество транзакций между режимом транзакций и режимом передачи (более трех пакетов в одном направлении и ни

одного в обратном направлении); время бездействия как сумма интервалов длительностью более 2 с, когда ни в одном направлении не передаются пакеты для режимов транзакций или передачи больших объемов трафика.

Применен подход недетерминированной, так называемой мягкой, кластеризации: один и тот же поток может принадлежать нескольким кластерам с определенной долей вероятности. Такой статистический подход применяется в практических ситуациях, например в случае, когда тренировочных данных недостаточно для принятия точного решения. Ставится задача найти наиболее правдоподобное множество кластеров, имея набор тренировочных данных и априорное ожидание. В основе лежит модель, называемая конечной смесью (finite mixture). Смесь — это уникальное для каждого кластера множество распределений вероятностей, моделирующее значения атрибутов для членов кластера. Для оценки параметров, обеспечивающих максимальное правдоподобие (en.wikipedia.org/wiki/Maximum_likelihood) смешанной модели, используется алгоритм EM (expectation-maximization) (en.wikipedia.org/wiki/Expectation_maximization_algorithm), имеющий необходимый статистический базис.

Переменные, указывающие на принадлежность к кластеру, являются скрытыми, т. е. если рассматривается единица данных x_i, то скрытая переменная z_{ij} принимает значение 0 или 1 в зависимости от того, принадлежит ли единица данных x_i к кластеру c_j. EM-алгоритм стартует с первоначально угаданными параметрами моделей кластеров, т. е. предполагается знание функций распределения для атрибутов и делается предположение о значении параметров распределения, а также вероятностей попадания образца в кластер $p(c_j)$. Затем алгоритм итеративно приближается к максимально правдоподобному выбору параметров распределения. Предполагается, что все атрибуты независимы, а числовые атрибуты (например, среднее значение интервала между пакетами) имеют нормальное распределение с неизвестными параметрами μ и δ (среднее и стандартное отклонение). Каждая итерация включает два шага. На шаге ожидания (expectation) проводится "мягкое" назначение кластера для каждого образца данных, т. е. рассматриваются текущие значения параметров распределения и делается предположение о вероятностях попадания образцов в кластер согласно относительной плотности распределения для каждой модели. На шаге максимизации (maximization) эти плотности рассматриваются как веса и используются для вычисления новых взвешенных оценок для параметров каждой модели. Итерации продолжаются до тех пор, пока увеличивается логарифм максимального правдоподобия всей модели.

Рассмотрим случай, когда число классов равно двум, а атрибут один. Алгоритм начинает работу с предположительных значений вероятности попадания рассматриваемых экземпляров в классы $p(c_1) = P$, $p(c_2) = 1 - P$. Вероятность попадания в класс зависит не от атрибутов, а только от множества рассматриваемых образцов данных. Для каждого распределения атрибута внутри класса выбираются параметры μ и δ, а значит, для каждого x_i можно вычислить $P(x_i|c_1)$ и $P(x_i|c_2)$. Правдоподобие вычисляется по формуле

$$\prod_{i=1}^{n} (p(c_1)P(x_i|c_1) + p(c_2)P(x_i|c_2)),$$

соответственно логарифм правдоподобия представляется в виде двойной суммы. Логарифм правдоподобия можно рассмотреть как функцию, зависящую от параметров p, μ_1, δ_1, μ_2, δ_2. Вследствие линейности логарифмической функции правдоподобия максимизация функции проводится по каждому параметру отдельно. Обычно алгоритм запускается несколько раз с различными начальными значениями, чтобы приблизиться к глобальному максимуму.

Количество кластеров, являющееся параметром реализации EM-алгоритма, может определяться подбором, но в данной работе предложен метод перекрестной проверки (cross-validation), обеспечивающий автоматическое нахождение числа кластеров. Этот метод дает обобщенную оценку полученной модели, например исследуется вопрос о поведении модели на данных, отличных от тренировочных. Для визуализации кластеров предлагается использовать kiviat-графы (radar, star chart (en.wikipedia.org/wiki/Radar_chart)) — двумерные графики, на которых числовые атрибуты соответствуют осям, выходящим из одной точки.

Используется реализация алгоритма, предложенная группой машинного обучения университета Waikato (www.cs.waikato.ac.nz/ml/weka). В результате кластеризации получилось шесть кластеров. Один из кластеров содержал 59 % всех потоков и относился к протоколу HTTP со схемой поведения, характерной для извлечения объектов малого и среднего размера. Один кластер содержал в основном потоки, относящиеся к TCP DNS. Остальные кластеры содержали потоки, соответствующие разным протоколам.

В качестве одного из вариантов проверки достоверности рассматривались кластеры, полученные на половине тренировочных данных. Получилось такое же базисное количество кластеров. Однако не удалось получить кластеры, каждый из которых соотносится с доминантным приложением. По-видимому, путем выбора другого набора атрибутов можно решить эту проблему.

В настоящей работе не приводится формальное описание реализации алгоритма кластеризации и методики, позволяющей снизить размерность вектора атрибутов. Однако представляет интерес подход к выявлению групп приложений, имеющих сходные характеристики по отношению к нагрузке на сеть.

3. Кластеризация трафика на основе алгоритма AutoClass и повышение качества кластеров. В работах [6, 7] для классификации трафика используется классификатор AutoClass [8], являющийся реализацией EM-алгоритма. Алгоритм AutoClass предназначен для нахождения множества кластеров (классов в терминологии AutoClass), максимально правдоподобного по отношению к данным и модели. Количество кластеров может быть определено автоматически, если оно не задано. Для определения числа кластеров и наилучшего разделения данных по кластерам EM-алгоритм используется итеративно. Методика повышения качества кластеров позволила достигнуть хороших результатов по разделению различных приложений.

Исследовались три общедоступных набора данных Auckland-VI, NZIX-II (WITS) и Leipzig-II (pma.nlar.net/special), собранные в разное время в разных точках. Рассматриваются двунаправленные потоки, определяемые пятеркой $<src\_ip$, src_port, dst_ip, dst_port, $protocol>$ и тайм-аутом 60 с. В качестве атрибутов потока рассматриваются средние и отклонения для интервалов между пакетами и для размеров пакетов, размер потока в байтах и длительность потока. Все параметры, кроме длительности, вычисляются отдельно для каждого направления трафика. Атрибуты считаются независимыми и моделируются с помощью логарифмически нормального распределения. Предлагаемая методика классификации включает следующие этапы.

1. Преобразование входных данных: организация пакетов в потоки, вычисление характеристик потоков и предварительная классификация потоков с помощью свободно распространяемой системы для сетевых измерений NetMate (sourceforge.net/projects/netmate-meter/).

2. Использование полученных характеристик потоков и модели атрибутов для неконтролируемого обучения на тестовых данных с помощью AutoClass. Поскольку обучение — длительный процесс, в качестве образцов взяты 1000 случайно выбранных потоков с TCP/UDP-

портами, соответствующими FTP, Telnet, SMTP, DNS, AOL Messenger, Napster, Half-Life (из 8000 потоков, соответствующих этим приложениям).

3. Оценка комбинаций атрибутов и повышение качества кластеров. Для нахождения наиболее контрастной кластеризации осуществляется поиск наилучшей комбинации атрибутов. Процесс поиска — это итеративный процесс, состоящий из трех фаз: 1) выбор подмножества атрибутов; 2) изучение полученных кластеров; 3) оценка структуры кластеров. Для поиска наилучшего подмножества атрибутов используется техника sequential forward selection. Процесс начинается с одного атрибута, показавшие себя наилучшим образом атрибуты помещаются в множество SEL(1). Затем проверяется набор из двух атрибутов, один из которых находится в SEL(1), а другой — не в SEL(1). Показавшие себя с наилучшей стороны наборы из двух атрибутов помещаются в SEL(2). Процесс продолжается до тех пор, пока не будет выявляться улучшение в структуре кластеров.

В качестве меры качества кластеризации используется мера однородности внутри кластера H. Пусть C — количество кластеров, A — количество приложений, N_{ac} — количество потоков приложения a, попавших в кластер c, N_c — количество потоков в кластере c. Тогда H_c — однородность класса c — определяется по формуле $H_c = \max(N_{ac}/N_c)$ по всем a $(0 \le a \le A-1)$. Для каждой попытки кластеризации мера однородности H определяется как средняя мера однородности H_c по всем кластерам $(0 \le c \le C-1)$. Задача состоит в максимизации H. При вычислении H "абсолютная истина" определяется согласно номеру порта назначения, зарезервированному за приложением авторитетным источником (the Internet assigned numbers authority (IANA)) (www.iana.org/assignements/port-numbes), так как исследуемые наборы данных содержат только заголовки пакетов. Методика проверялась на трех наборах данных или с разделением одного набора на две части. Выяснилось, что для различных наборов данных лучшими относительно меры однородности H оказались различные наборы атрибутов, а их количество варьируется в пределах от 4 до 6. Для рассмотренных наборов данных наиболее эффективными оказались атрибуты, построенные на основе статистики величины отклонения размеров пакетов в потоках. Показатели по различным приложениям также оказались различными. Таким образом, определенный успех в разделении приложений достигается при правильном подборе атрибутов. Наиболее однородными оказались кластеры, соответствующие приложению Half-Life. В среднем для разных тестовых данных однородность H составляет 85 %. В качестве показателя неверной классификации для приложения рассматривается отношение неверно классифицированных потоков по отношению к потокам этого приложения, которым приписаны классы. Наименьший показатель распознаваемости имеют FTP-, Web- и Telnet-трафики.

Следует отметить, что при анализе не рассматривались потоки, содержащие менее трех пакетов, так как для таких пакетов невозможно вычислить некоторые характеристики. Если для TCP-потоков таким образом отбрасываются аномальные потоки (TCP предусматривает отправку не менее трех пакетов в каждом направлении), то для UDP-потоков при таком подходе может быть отброшен нормальный трафик, например DNS-запросы. В результате исследования трафика, относящегося к восьми приложениям, получилось около 50 кластеров. Неясно, какая интерпретация соответствует кластерам, для которых не определено доминантное приложение.

4. Сравнение трех алгоритмов кластеризации. В работе [9] проведено сравнение результатов кластеризации сетевого трафика с помощью алгоритмов K-Means [2], DBSCAN [10] и AutoClass [8]. При этом для первых двух алгоритмов при проведении исследования отсутствовала информация о применении к кластеризации сетевого трафика. Исследова-

ние проведено для данных алгоритмов, так как известно, что они работают быстрее, чем AutoClass. Алгоритмы сравнивались с точки зрения способности генерировать кластеры, которые в большей степени можно было бы отнести к одному приложению, что в итоге представляет наибольший интерес для исследователей, разрабатывающих эффективный и точный механизм классификации.

Исследования проводились на двух блоках данных: общедоступном суточном блоке Auckland IV (WITS) и собственном блоке данных Calgary, состоящем из трафика в точке выхода университета Calgary в глобальную сеть и содержащем полные пакеты. В данном случае поток означает двунаправленный трафик, соответствующий полной TCP-сессии начиная с установления соединения до разрыва или до обнаружения простоя в течение 90 с. Выбор атрибутов в значительной степени обусловлен работой [7], в которой приведен непротиворечивый и неизбыточный список атрибутов.

Рассматривались следующие атрибуты потоков: количество пакетов, средний размер пакета, средний размер полезной нагрузки пакета (в каждом направлении и общий) и средний интервал между пакетами. Выполняется логарифмическое преобразование значений атрибутов, поскольку многие характеристики имеют распределения с "тяжелыми хвостами" и в качестве метрики используется евклидово расстояние между векторами атрибутов [11, 12].

Категории трафика определялись на основе изучения номеров портов, так как блок Auckland IV содержит только заголовки пакетов. Блок представлен категориями DNS, FTP, HTTP, IRC, LIMEWARE (P2P), NNTP, POP3 и SOCKS. Второй блок исследовался путем поиска в полном содержании пакетов шаблонов, соответствующих исследуемым протоколам, и состоял из трафика по протоколам HTTP, P2P, SMTP и POP3. В обоих блоках данных доминирующим трафиком оказался HTTP. Поэтому блоки были преобразованы: в блоке Auckland IV случайным образом было выбрано по 1000 образцов каждой категории трафика, а в блоке Calgary — по 2000 образцов каждой категории.

В качестве меры близости векторов атрибутов x и y размерности n использовалось евклидово расстояние:

$$\operatorname{dist}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

В качестве меры эффективности алгоритмов используется полная правильность (от англ. overall accuracy) — суммарная правильность по всем кластерам, характеризующая способность построить кластеризацию, при которой каждый кластер соотносится только с одной категорией трафика. В данном случае кластер помечается категорией, если большинство потоков относится к этой категории. Полная правильность определяется как ΣTP — количество исследуемых потоков.

Кластеризация K-Means. Кластеризация предполагает предварительное задание числа кластеров K. Алгоритм проверялся для различных значений K начиная с 10 (ожидается, что для каждой категории получится хотя бы один кластер) с шагом 10. При малом количестве кластеров (10–20) для блока Auckland IV полная правильность составила в среднем 49 %, для блока Calgary — 67 %. При $K = 100$ правильность возросла в среднем до 79 и 84 % соответственно. При $K = 150$ правильность возросла только на 1 %, а затем даже уменьшалась. Большое количество кластеров приводит к увеличению вероятности того, что модель будет описывать случайную ошибку или шум (overfitting).

Кластеризация DBSCAN. Кластеризация с помощью алгоритма DBSCAN предполагает предварительное задание двух параметров: ε (максимальное расстояние между точками) и *MinPts* (минимальное количество близких точек). Параметр *MinPts* варьировался в интервале 3÷24, параметр ε — в интервале 0,005÷0,040. При $MinPts = 3$, как и ожидалось, результаты лучше, чем при $MinPts = 24$, так как в последнем случае получается очень малое количество кластеров. При $MinPts = 3$ и варьировании ε в пределах от 0,005 до 0,020 для блока Auckland IV полная правильность варьируется в диапазоне от 59,5 до 75,6 %, для блока Calgary — в диапазоне от 32 до 72 %. Оказалось, что при $\varepsilon > 0{,}020$ правильность уменьшается вследствие объединения в один кластер трафиков разных категорий.

Кластеризация Autoclass. Применение алгоритма AutoClass дало наибольшую правильность: 92,4 и 88,7 % для блоков Auckland IV (167 кластеров) и Calgary (247 кластеров) соответственно.

Проблемой кластеризации является интерпретация кластеров. При большом количестве кластеров трудно пометить все кластеры соответствующими приложениями. Задача упрощается, если небольшое количество кластеров включает большое количество единиц данных, в данном случае — потоков. Хорошей характеристикой обладает DBSCAN: для блока Auckland IV в результате работы DBSCAN ($\varepsilon = 0{,}03, MinPts = 3$) пять наибольших кластеров содержат 50 % потоков, при этом идентифицированы 75,4 % трафика категорий NNTP, POP3, SOCKS, DNS и IRC с полной правильностью 97,6 %. В то же время в результате работы K-Means ($K = 100$) 15 наибольших кластеров содержат 50 % потоков. Такие же качественные результаты получены при исследовании трафика блока Calgary. Алгоритм DBSCAN обладает способностью помечать часть данных как шум. С одной стороны, эти данные представляют собой выбросы, с другой — признанные шумом потоки считаются нераспознанными. Такие потоки были отсеяны, после чего алгоритмы сравнивались относительно меры точность (precision), в большинстве работ определяемой по формуле $TP/(TP+FP)$, а в данной работе — по соотношению TP/FP для трафика, представляющего собой кластер. Эта мера отражает способность правильно классифицировать категорию трафика. Интересно, что все три алгоритма обеспечивают практически одинаковую точность для категории P2P. Алгоритм DBSCAN достиг точности, приближенно равной 95 %, для семи из девяти категорий трафика. Несмотря на то что алгоритм DBSCAN имеет меньшую полную правильность по сравнению с двумя другими алгоритмами, он порождает точные кластеры. При этом среднее время, затраченное на распознавание, сравнимо со средним временем, затрачиваемым при работе K-Means. При кластеризации с помощью алгоритма AutoClass затрачивается существенно большее время. Из результатов сравнения K-Means и AutoClass следует, что K-Means обеспечивает несколько меньшую полную правильность, но при этом работает во много раз быстрее.

Исследования показали, что результаты существенно зависят от выбора параметров алгоритмов. В этом отношении алгоритм AutoClass, не имеющий параметров, предпочтителен. Остается неизвестной зависимость результатов исследования от выбранных образцов данных.

5. Кластеризация однонаправленных потоков трафика с применением алгоритма K-Means. Работа [13] является продолжением исследования алгоритмов кластеризации применительно к категоризации сетевого трафика. Как показано в п. **4**, алгоритм K-Means является быстрым и достаточно эффективным, поэтому он был выбран для того, чтобы выяснить, какие результаты покажет алгоритм в случае исследования однонаправленных потоков. Такая ситуация может возникнуть, например, вследствие несимметричной

маршрутизации. Вновь рассматривается только TCP-трафик, так как формат TCP-пакетов позволяет по трафику в одном направлении оценить такие атрибуты потока, как длительность, средний размер пакета и средний интервал между пакетами трафика в другом направлении, т. е. оценить параметры двунаправленного потока [13].

Исследовались три типа потоков: сервер → клиент, клиент → сервер и двунаправленные потоки. В качестве атрибутов выбраны общее количество пакетов, средний размер пакета, средний размер полезной нагрузки пакета, средний интервал между пакетами, общее количество байтов и длительность. Значения атрибутов подвергались логарифмической трансформации. В качестве меры близости потоков использовалось евклидово расстояние между векторами атрибутов. В качестве исследуемых данных рассматривались восемь одночасовых блоков. Минимум данных объясняется тем, что собирались полные пакеты, которые тщательно исследовались различными, в том числе сигнатурным, методами (аналогичный подход использовался в работах [14, 15]).

В результате предварительного анализа исходных данных установлено, что приблизительно 85 % пакетов и 90 % байтов соответствуют TCP-трафику. Выявлено 29 протоколов включая BitTorrent, eDonkey, KaZaA и другие P2P-протоколы (4 % потоков и 36 % байтов общего количества в блоке данных). Идентифицированный трафик был представлен категориями Web, EMAIL, DATABASE, P2P, CHAT, FTP, STREAMING и OTHER (трафик, соответствующий различным протоколам, каждый из которых представлен небольшим количеством потоков). Неидентифицированный трафик был разделен на три группы: UNKNOWN(NP) (трафик без полезной нагрузки); UNKNOWN(443) (трафик с использованием порта 443, соответствующий протоколу HTTPS); UNKNOWN (Other) (остальной неидентифицированный трафик). Web- и P2P-трафики лидируют по количеству байтов.

В качестве основной метрики, оценивающей кластеризацию, рассматривалась правильность (отношение количества правильно классифицированных единиц к общему количеству единиц) в байтах и потоках. Кроме того, рассматривались такие меры, как точность $TP/(TP + FP)$ и полнота $TP/(TP + FN)$.

Алгоритм K-Means предполагает предварительное задание количества k желаемых кластеров. Изучались результаты кластеризации для значений k в диапазоне от 25 до 400. При увеличении количества кластеров для всех трех рассматриваемых типов потоков (сервер → клиент, клиент → сервер и клиент-сервер) правильность в потоках и байтах увеличивается. Правильность в потоках увеличивается на 5–8 % для всех трех типов потоков. Правильность в байтах увеличивается с 59 до 80 % для потоков сервер → клиент, причем это происходит за счет корректности идентификации именно P2P-трафика. При $k = 25$ P2P-трафик почти всегда определяется как Web-трафик, причем весь трафик P2P попадает в один кластер. При $k = 400$ P2P распределяется в 12–15 кластерах, причем правильность в байтах составляет 80 %. При больших значениях параметра k выявляется большее количество характеристик приложений.

Таким образом, потоки сервер → клиент дают высшую правильность классификации — 95 и 79 % в байтах и потоках соответственно. Наилучшие результаты классификации получены для трафиков категорий DATABASE, EMAIL и Web. Наиболее сложна классификация P2P-трафика.

Разработан алгоритм оценки трафика сервер → клиент путем изучения трафика клиент → сервер. Несмотря на то что протокол TCP достаточно прогнозируем и позволяет делать выводы на основании полей в заголовках пакетов (номера последовательности, максимальный размер сегмента) и обмена пакетами с флагами ACK и RST, при реализации

сделан ряд предположений, например об отсутствии потерь пакетов и о стратегии обмена подтверждениями. На основе оценки атрибутов потоков трафика сервер → клиент по наблюдению за потоками клиент → сервер была проведена кластеризация потоков в двух направлениях. Результаты кластеризации показали, что правильность в потоках и байтах незначительно отличается от правильности, имевшей место при кластеризации потоков реального двунаправленного трафика.

Помимо влияния на результаты классификации направления исследовалось влияние размеров блока данных (количества пакетов). Алгоритм кластеризации был применен к блокам данных различных размеров, к трафику, собранному в различные дни. Это позволило сделать вывод о стабилизации результатов начиная с некоторого объема, что позволяет считать возможной классификацию в реальном времени.

6. Подходы к созданию систем классификации сетевого трафика. В работах [16, 17] приводится методология создания системы классификации сетевого трафика. Предлагается использовать метод частично контролируемой кластеризации, имеющий ряд преимуществ. Во-первых, требуется лишь небольшое количество помеченных потоков (принадлежность к классу), смешанное с большим количеством непомеченных потоков. Пометка образцов создает определенную трудность, а малое количество используемых при обучении образцов может привести к неверным результатам. Во-вторых, имеется возможность отслеживания новых приложений и нового поведения уже встречавшихся приложений, а в случае контролируемой классификации требуется сопоставить каждому типу потока определенный класс. В-третьих, подход может быть использован в рамках систем сбора статистики потоков, функционирующих как в точках выхода из локальной сети в глобальную сеть, так и в пределах опорной сети.

Под потоком понимается двунаправленная последовательность пакетов с учетом протокола транспортного уровня и номеров портов. Конец потока определяется по завершению связи или по тайм-ауту. Как и выше, рассматривается только наиболее распространенный TCP-трафик. Для каждого транспортного протокола требуется отдельный классификатор. В качестве атрибутов потока выбраны общее количество пакетов, средний размер пакета, общее количество байтов, общее количество байтов в заголовках, количество пакетов от активной стороны к пассивной, количество байтов от активной стороны, количество байтов полезной нагрузки от активной стороны, общее количество байтов в заголовках от активной стороны. Для выбора подмножества атрибутов использовалась технология backward greedy search [18].

При пометке тренировочных данных для установления “абсолютной” истины применялся комбинированный подход, основанный на автоматическом исследовании сигнатур; соотнесении трафика, содержащего зашифрованные данные, с трафиком, относящимся к тем же IP-адресам и содержащим открытый текст; в случае HTTPS (порт 443) проводилась вручную проверка наличия обращений к реальным web-серверам.

Предложенный метод, сочетающий контролируемый и неконтролируемый подходы, включает два этапа. Предположим, зафиксирован набор желаемых категорий приложений $Y = \{Y_1, \ldots, Y_q\}$. На первом этапе алгоритм кластеризации применяется к тренировочному набору помеченных и непомеченных потоков. В данном случае используется алгоритм K-Means. Затем помеченные потоки используются для того, чтобы поставить в соответствие кластерам известные категории. Кластер получает метку приложения, к которому принадлежит максимум помеченных потоков. Все непомеченные потоки, попавшие в данный кластер, получают метку приложения кластера. При этом некоторым кластерам невозможно поставить

в соответствие категорию, если в них не вошли помеченные потоки. Такие кластеры помечаются меткой Unknown.

Изучалась возможность проводить кластеризацию без использования помеченных потоков. В результате экспериментов установлено, что можно предварительно не помечать потоки. При достаточно большом числе кластеров ($k = 400$) можно сначала провести кластеризацию, затем пометить лишь несколько произвольно выбранных потоков из кластера и получить правильность, равную 94 %. Эксперимент проводился на блоке, состоящем из 64 000 потоков.

Следующая серия экспериментов выполнялась с целью определения количества помеченных потоков, необходимого для качественной кластеризации. Обнаружено, что при фиксированном количестве помеченных тренировочных потоков увеличение количества непомеченных потоков приводит к увеличению правильности. Это существенно, так как пометка потоков сложна, кроме того, следует учесть возможность ошибок.

Классификация в реальном времени предполагает максимально быстрое выявление категории, к которой принадлежит рассматриваемый поток. В отличие от автономной классификации, когда доступна вся информация о потоках, в реальном времени доступна лишь часть информации о статистических параметрах потоков. Для преодоления этой трудности была разработана многоуровневая (layered) система классификации. Уровни базируются на понятии packet milestone (мильный столб, граница). Граница достигается, когда количество отправленных или полученных пакетов достигает определенного значения (пакеты SYN/SYNACK включаются в рассмотрение). Каждый уровень — это независимая модель классификации. Модель получается путем обучения на образцах, достигших соответствующего размера, определяемого packet milestone. Можно определять слои на основе особенностей транспортного протокола. Первый слой определяет только транспортный протокол и порты. Для TCP-трафика поток переходит на второй уровень при получении первого пакета данных и т. д.

Многоуровневый подход позволяет потенциально улучшить классификацию. На каждом уровне рассматривается одно и то же множество атрибутов потоков. Статистика собирается по мере сбора пакетов данного потока. Когда достигается первая граница, неполный поток классифицируется согласно модели первого уровня. Когда достигается следующая граница, поток вновь классифицируется согласно модели данного уровня, т. е. происходит ревизия. Правильность реального времени вычисляется на каждом уровне относительно байтов — доля байтов, получивших правильную метку. Окончательная классификация потока происходит на уровне, где поток завершается. Такой многоуровневый подход реализован в классификаторе реального времени, используемом в системе защиты от несанкционированного доступа Bro [19].

При проверке многоуровневой методики исследовался блок данных, состоящий из 966 000 потоков. На каждом уровне модель строилась на основе 8000 потоков при $k = 400$. Рассматривалось 13 слоев, границы распределялись экспоненциально: 8, 16, 32,.... Одиннадцать и более слоев достигли только 5 % потоков размером более 4096 байтов. На первом этапе были правильно оценены 40 % байтов, на пятом — 50 %, на тринадцатом — 78 %. Последняя мета, полученная потоком, оказалась корректной в 82 % случаев. Замечено, что некоторые слои, не увеличивающие правильность, могут быть исключены.

Эксперименты показали, что рассмотренные подходы позволяют получить классификаторы, которые можно использовать в течение достаточно длительных периодов времени, переобучение необходимо только при значительных изменениях в сетевом поведении, например при появлении новых приложений.

7. Гибридный классификатор реального времени для выявления P2P-взаимодействия. В работе [20] для классификации в реальном времени используется гибридный классификатор, состоящий из аппаратного классификатора, действующего на уровне номеров портов и сигнатур прикладного уровня, и программного классификатора на базе нейронной сети специального типа.

Аппаратный классификатор строится на основе сетевого процессора IXP2400, обеспечивающего скорость, позволяющую обслуживать гигабитные линии, программируемость, возможность доступа к памяти, сетевые интерфейсы. Процессоры, в которых реализована архитектура высокоскоростной параллельной обработки, способны реализовывать сложные алгоритмы, в том числе исследование полного содержимого пакетов, управление трафиком и перенаправление трафика на высоких скоростях. В случае распознавания P2P-трафика пакет не передается на уровень программного классификатора, т. е. аппаратный классификатор работает как фильтр. Кроме того, в задачу аппаратного классификатора входит формирование значений атрибутов трафика, используемых программным классификатором.

Программный классификатор оперирует статистическими параметрами трафика. Для идентификации P2P-трафика изучаются поведенческие характеристики участников обмена на транспортном уровне, подобные приведенным в работе [21]: 1) одновременное использование протоколов TCP и UDP в трафике между двумя IP-адресами; 2) редкое обращение к службе доменных имен; 3) практически совпадающее количество исходных адресов и номеров портов при обращении к некоторому IP-адресу назначения (характерное при P2P-обмене в силу случайного выбора номеров портов); 4) использование небольшого количества номеров портов источника при обращении к разным адресам и портам назначения при UDP-обмене. Рассматриваются отражающие эти свойства нормализованные параметры, вычисляемые на интервале t.

Одновременное использование IP-адресом протоколов TCP и UDP учитывает влияние параметра f_{pro}:

$$f_{pro} = \frac{N_{pro} - N_{spec}}{N_{pro}}.$$

Здесь N_{pro} — количество IP-адресов, при взаимодействии с которыми используются одновременно оба транспортных протокола; N_{spec} — количество IP-адресов, при взаимодействии с которыми используются номера портов, входящие в список исключений. Эти номера портов соответствуют протоколам, отличным от P2P, для которых также свойственно одновременное использование на транспортном уровне TCP и UDP. В данном случае в список исключений входят следующие номера портов: {135 (Location Service), 137 (NETBIOS Name Service), 139 (NETBIOS Session Service), 445 (Microsoft-DS), 53 (DNS), 21 (FTP), 1433 (Microsoft SQL Server), 1434 (Microsoft SQL Monitor)}. Чем больше величина f_{pro}, тем больше вероятность того, что это P2P-взаимодействие.

Редкое обращение к службе доменных имен (DNS) учитывает влияние параметра f_{DNS}:

$$f_{DNS} = \frac{N_{\log}}{N_{all}}.$$

Здесь N_{all} — количество IP-адресов, с которыми взаимодействовал данный IP-адрес; N_{log} — количество адресов, зафиксированных в журнале DNS log. Чем меньше значение f_{DNS}, тем больше вероятность того, что IP-адрес участвует в P2P-обмене.

Практически совпадающее количество исходных адресов и номеров портов при обращении к рассматриваемому IP-адресу учитывает влияние параметра $f_{IP-Port}$:

$$f_{IP-Port} = \frac{N_{ip} - N_{Port}}{N_{Port}}.$$

Здесь N_{ip} — количество исходных IP-адресов, взаимодействующих с IP-адресом назначения; N_{Port} — количество разных исходных номеров портов, используемых при взаимодействии. Чем меньше значение $f_{IP-Port}$, тем более вероятно, что рассматриваемый IP-адрес назначения принимает участие в P2P-обмене.

Использование небольшого количества номеров портов источника при обращении к разным адресам и портам назначения при UDP-обмене учитывает влияние параметра

$$f_{u\mathrm{Port}} = \frac{m}{\sum\limits_{i=1}^{m} N_i}.$$

Здесь m — количество разных номеров портов назначения, используемых некоторым IP-адресом источника при UDP-обмене; N_i — количество различных IP-адресов назначения, с которыми рассматриваемый IP-адрес источника контактировал с использованием i-го UDP-порта. Чем меньше значение $f_{u\mathrm{Port}}$, тем больше вероятность того, что IP-адрес участвует в P2P-взаимодействии.

Программный классификатор P2P строится на основе нейронной сети специального вида, так называемого flexible neuron tree (FNT) [22]. Нейронная сеть имеет вид дерева с двумя скрытыми слоями. Особенностью сети является использование различных функций активации для разных узлов, необязательная связанность элементов слоя i со всеми элементами слоя $i+1$ и пересечение слоев, т. е. на вход слоя $i+1$ поступают не только элементы слоя i, но и элементы слоев, номера которых меньше i. В качестве входов x0, x1, x2, x3 используются атрибуты $f_{pro,}f_{IP-Port}$, f_{uPort}, f_{DNS}, причем они могут использоваться и в скрытых слоях, но на выходе, в корне, должен быть только один элемент y. По его значению определяется принадлежность к P2P-трафику. В качестве функции активации используется сигмоидная функция вида $f(a, b, x)$. При тестировании классификатора атрибуты для IP-адреса подсчитывались через каждые 5 мин. Выходной элемент y — это вещественное число между 0 и 1. Если значение y лежит в диапазоне [0; 0,5], то IP-адрес принимает участие в P2P-обмене с большой вероятностью, в противном случае это не P2P-трафик.

Возможности идентификации P2P-трафика рассмотрены на примере обучения нейронной сети на трафиках BitTottent и Edonkey, выявляемых с помощью аппаратного классификатора с последующей идентификацией гибридным классификатором трафика P2P, соответствующего протоколам BitTottent, Edonkey Kazaa, PPlive и Skype. Было исследовано примерно 1675 Мбайт, из них около 885 Мбайт соответствовали P2P-трафику. При этом правильность составила приблизительно 95 %.

Подход представляется перспективным. Во-первых, первоначальное обучение потребовало наличия лишь небольшого количества правил. Во-вторых, имеется возможность пополнять правила аппаратного классификатора путем изучения трафика, идентифицированного программным классификатором, а также возможность переобучения в реальном времени нейронного дерева на основе предшествующих результатов.

Заключение. В настоящей работе представлены методы анализа IP-трафика с применением алгоритмов кластеризации. Анализ проводился на основе информации, содержащейся

в заголовках IP-пакетов. Основными проблемами, возникающими при решении задачи методом кластеризации, являются выбор адекватных характеристик классифицируемых данных, выбор метода объединения в кластеры, выбор числа кластеров и интерпретация результатов.

Рассмотрены алгоритмы кластеризации различной сложности, предназначенные для пассивного анализа и анализа в реальном времени. Единицами рассмотрения являются потоки, как правило, сформированные на основе полных TCP-сессий. Для получения достоверных данных, необходимых на этапе тренировки для пометки кластеров и при определении эффективности классификатора, используются в основном два подхода. В случае если доступна вся информация о содержимом IP-пакетов, используется метод поиска шаблонов, характерных для изученных приложений. Если доступны только данные заголовков пакетов, приходится основываться на номерах портов, зарезервированных за приложениями, а также проверять данные вручную, например, действительно ли IP-адрес является адресом web-сервера.

Сравнение алгоритмов кластеризации, тестируемых в одном и том же сетевом окружении и на одном и том же множестве атрибутов, показало, что алгоритмы кластеризации различаются не столько по качеству, сколько по времени обучения и длительности процесса классификации. Несмотря на удовлетворительные оценки правильности в целом, существуют различия в эффективности классификации различных категорий трафиков. Наиболее сложным при распознавании является P2P-трафик.

Основными факторами, влияющими на качество анализа, являются следующие: 1) важность выбора набора непротиворечивых и неизбыточных атрибутов и зависимость этого набора от образцов данных; 2) зависимость от параметров алгоритмов, например от выбора порогов и количества кластеров; 3) восприимчивость предполагаемых параметров распределения значений атрибутов внутри класса или кластера к первому выбору; 4) возможность применения к однонаправленному трафику; 5) влияние размеров блоков образцов данных на качество алгоритма.

При изучении различных алгоритмов классификации и кластеризации и методик их применения выявляется сложность решения задачи классификации трафика даже при наличии ограничений и предположений, сделанных в ходе исследования. Трудность состоит прежде всего в необходимости выявления свойств, характеризующих приложения, но не зависящих от реализации, уникальных и позволяющих идентифицировать приложения в реальном времени. Для того чтобы решить проблемы реализации приложений, следует чаще проводить переобучение. Однако при этом возникает проблема сложности проверки: при небольших объемах данных проще установить “абсолютную” истину относительно представленных классов приложений, но параметры эффективности неубедительны, в то время как большие объемы труднее проанализировать.

Таким образом, несмотря на большое количество работ, проблема классификации IP-трафика не решена в полном объеме. Представляются перспективными гибридные классификаторы, применяющие одновременно несколько методик.

Список литературы

1. Щербакова Н. Г. Анализ IP-трафика методами Data Mining // Пробл. информатики. 2012. № 4. С. 30–46.

2. Барсегян А. А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, Olap / А. А. Барсегян, М. С. Куприянов, В. В. Степаненко, И. И. Холод. СПб.: БХВ-Петербург, 2007.

3. Mitchell T. Machine learning. N. Y.: McGraw-Hill, 1997.

4. Гмурман В. Е. Теория вероятностей и математическая статистика. М.: Юрайт, 2011.

5. McGregor A., Hall M., Lorier P., Brunskill J. Flow clustering using machine learning techniques // Passive and active measurement workshop (PAM 2004): Proc. Lecture Notes in Computer Science. V. 3015, Antibes Juan-les-Pins (France), Apr. 2004. N. Y.: Springer, 2004. P. 205–214.

6. Zander S., Nguyen T., Armitage G. Self-learning IP traffic classification based on statistical flow characteristics // Passive and active measurement workshop (PAM 2005): Proc. Lecture Notes in Computer Science. V. 3431, Boston, USA, March — Apr. 2005. N. Y.: Springer, 2005. P. 325–328.

7. Zander S., Nguyen T., Armitage G. Automated traffic classification and application identification using machine learning // 30th Annual IEEE conf. on local computer networks (LCN 2005): Proc. IEEE Computer Soc., Sydney (Australia), Nov. 2005. Washington: IEEE Computer Soc., 2005. P. 250–257.

8. Cheeseman P., Stutz J. Bayesian classification (AutoClass): theory and results // Advances in knowledge discovery and data mining. Palo Alto (CA, USA): AAAI/MIT Press, 1996. P. 61–68.

9. Erman J., Arlitt M., Mahanti A. Traffic classification using clustering algorithms // Special interest group on data communication (SIGCOMM) 2006 workshops: Proc. of the 2006 SIGCOMM workshop on Mining network data, Pisa (Italy), Sept. 11–15, 2006. N. Y.: ACM, 2006. P. 281–286.

10. Ester M., Kriegel H., Sander J., Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise // Proc. of the 2nd Intern. conf. on knowledge discovery and data mining (KDD-96), Portland (USA), 1996. Palo Alto: AAAI/MIT Press, 1996. P. 226–231.

11. Witten I. H. Data mining: practical machine learning tools and techniques / I. H. Witten, E. Frank. San Francisco: Morgan Kaufmann, 2005. P. 560.

12. Paxson V. Empirically-derived analytic models of wide-area TCP connections // IEEE/ACM Trans. Network. 1994. V. 2, N 4. P. 316–336.

13. Erman J., Mahanti A., Arlitt M., Williamson C. Identifying and discriminating between web and peer-to-peer traffic in the network core // Proc. of the 16th Intern. conf. on world wide web (WWW) 2007, Banff (Alberta, Canada), May 8–12, 2007. N. Y.: ACM, 2007. P. 883–892.

14. Sen S., Spatscheck O., Wang D. Accurate, scalable in-network identification of P2P traffic using application signatures // Proc. of the 13th Intern. world wide web conf., New York (USA), May 2004. N. Y.: ACM, 2004. P. 512–521.

15. Karagiannis T., Papagiannaki K., Faloutsos M. BLINK: multilevel traffic classification in the dark // ACM SIGCOMM Computer Comm. Rev. 2005. V. 35, iss. 4. P. 229–240.

16. Erman J., Mahanti A., Arlitt M., et al. Semi-supervised network traffic classification // Proc. of the Intern. conf. on measurement and modeling of computer systems (SIGMETRICS'07), San Diego (USA), June 12–16, 2007. N. Y.: ACM, 2007. P. 369–370.

17. Mahanti A., Arlitt M., Cohen I., Williamson C. Offline/realtime traffic classification using semi-supervised learning: Tech. rep. Univ. of Calgary. 2007. V. 64. P. 1194–1213.

18. Guyon I., Elisseeff A. An introduction to variable and feature selection // J. Machine Learn. Res. 2003. V. 3. P. 1157–1182.

19. Paxson V. Bro: a system for detecting network intruders in real-time // Computer Networks. 1999. V. 31, iss. 23/24. P. 2435–2463.

20. Chen Z., Yang B., Chen Y., et al. Online hybrid traffic classifier for Peer-to-Peer systems based on network processors // Appl. Soft Comput. 2009. V. 9, N 2. P. 685–694.

21. Karagiannis T., Broido A., Faloutsos M., Mc Claffy. Transport layer identification of P2P traffic // Proc. of the 4th ACM SIGCOMM conf. on Internet measurement (IMC'04), Taormina (Sicily, Italy), Oct. 25–27, 2004. N. Y.: ACM, 2004. P. 121–134.

22. Chen Y., Yang B., Dong J. Nonlinear systems modelling via optimal design of neural trees // Intern. J. Neural Systems. 2004. V. 14. P. 125–138.

Щербакова Наталья Григорьевна — ст. науч. сотр. Института вычислительной математики и математической геофизики СО РАН; e-mail: nata@nsc.ru

Дата поступления — 11.01.13