УДК 004.7

Ф.И. АНДОН, А.П. ИГНАТЕНКО

МОДЕЛИРОВАНИЕ КОНФЛИКТНЫХ ПРОЦЕССОВ В ИНТЕРНЕТЕ

Ключевые слова: *моделирование сетей Интернет, управление переполнением, TCP-алгоритмы, атаки типа «отказ в обслуживании».*

ВВЕДЕНИЕ

Современные компьютерные сети буквально пронизывают жизнедеятельность человечества. Всего за несколько десятилетий Интернет стремительно проник в нашу жизнь, объединив около 2,4 миллиарда пользователей в наибольшую за историю планеты искусственную коммуникационную структуру. Однако непрерывное успешное развитие информационных сетей, сопровождающееся их усложнением, интеллектуализацией и распределенностью, которое сегодня выглядит естественным, было обусловлено решением ряда сложнейших задач, возникших во второй половине 80-х годов прошлого века. Успешное решение этих задач было неразрывно связано с разработкой протокола управления передачей данных [1] (Transmission control protocol — TCP). На текущий момент этот протокол является ключевым для развития и функционирования Интернета, с его помощью передается более 70 % всего трафика. Непосредственной предпосылкой создания механизмов Congestion control (управления переполнением) стало нарастание негативных явлений, получивших название коллапс переполнения [2]. Основной предпосылкой этих явлений был нерегулируемый и неограниченный доступ пользователей к сети (к тому времени она состояла из нескольких десятков узлов, расположенных преимущественно в научных институтах). Возрастание скорости передачи данных существенно ухудшило работу сети (в некоторых случаях 90 % пакетов терялось), поэтому возникла необходимость создания децентрализованного алгоритма управления. Главными составляющими нового алгоритма стали два механизма: управление окном (congestion window) и обратная связь на основе пакетов подтверждений (ACK-clocking). Рассмотрим систему, состоящую из источника и приемника, соединенных сетью. Окно — это количество пакетов, которые источник может отправить в сеть до возвращения подтверждения об успешном получении первого пакета. В идеальном случае окно должно равняться количеству пакетов источника, которые пребывают «в дороге» (на практике это выполняется очень редко). Время, необходимое на подтверждение всех пакетов окна, называется временем обращения (Round Trip Time — RTT).

Согласно В. Джекобсону [1] идея первого варианта протокола состояла в формулировке общего принципа, которому должны подчиняться TCP-потоки и который гарантировал бы устойчивую работу системы в окрестности точки равновесия. Этот принцип заключается в том, что каждое TCP-соединение должно управлять своим окном, исходя из доступной ему информации о состоянии сети. Естественным (но не единственным) источником такой информации являются пакеты подтверждения (ACK-пакеты). Отослав пакет приемнику, источник может рассчитывать на два возможных исхода: либо пакет успешно получен, либо он потерян (поврежден, отброшен вследствие переполнения буфера или сбоя оборудования). В первом случае следует увеличивать окно аддитивно, т.е. отсылать в сеть

пакет взамен каждого доставленного плюс один, т.е. окно увеличивается вдвое после каждого RTT. Если считать RTT постоянным, то окно изменяется по экспоненциальному закону. Во втором случае сеть, возможно, оказалась под угрозой переполнения, и источнику следует уменьшить окно мультипликативно (в оригинальном алгоритме вдвое). В результате возникает ситуация, когда источник ожидает новых подтверждений, увеличивая окно, пока достаточное количество его пакетов не покинет сеть и ему будет разрешено продолжить передачу. Описанная идея составила ядро предложенного в [1] алгоритма TCP Tahoe, который впервые решил проблему коллапса переполнения. Данный алгоритм является реализацией AIMD (additive increase, multiplicative decrease) схемы, впервые исследованной в контексте компьютерных сетей [3]. Из возможных вариантов реакции системы с бинарной обратной связью эта схема единственная устойчиво стремится к оптимальному состоянию (при выполнении определенных условий). Дальнейшее развитие алгоритмов TCP-семейства подробно описано в [4] и связано с попытками решения новых проблем, регулярно возникающих с развитием Интернета. Для решения этих проблем (например, связанных со случайными потерями в беспроводных сетях или поведением потоков в высокопродуктивных линиях связи) предложено более 50 экспериментальных протоколов, однако только около 10 принято как стандарт в современных операционных системах. Как отмечают авторы в работе [4], ситуация такова, что ни один из существующих вариантов TCP не может подходить всем возможным состояниям, которые встречаются на практике. Например, потеря пакета в обычной проводной сети, как правило, указывает на переполнение — вероятность потери пакета при пересылке намного меньше 1 %. В беспроводных сетях в зависимости от различных внешних факторов вероятность пересылочной потери может достигать 50 %, а это — основной индикатор состояния загруженности сети для TCP. Высокопродуктивные линии связи представляют другую проблему. Пропускная способность такой линии составляет десятки тысяч пакетов, поэтому даже при экспоненциальном росте TCP потоку нужно достаточно много времени для заполнения существующих мощностей. Если же линия все же будет заполнена и произойдет сброс пакетов (вследствие переполнения буфера узкого места), количество потерь может исчисляться мегабайтами. Для каждой из таких ситуаций разработан специальный TCP-алгоритм, однако глобальная проблема состоит именно в том, что каждый отдельный узел принимает решения, в условиях неполной информированности о состоянии сети. Таким образом, можно утверждать следующее:

1) в настоящее время алгоритмы TCP-семейства решают проблему коллапса переполнения;

2) некоторые алгоритмы специализированны под решение других важных проблем (изменение порядка получения пакетов, потери в беспроводных сетях, работа в высокопродуктивных линиях);

3) комплексное применение существующих алгоритмов для получения наиболее эффективного для данного состояния сети решения остается открытой проблемой.

Понятие эффективности в контексте сетевых протоколов требует пояснения. Пожалуй, наиболее полно развитие механизмов управления переполнениями TCP-алгоритмов описано в работе [5]. Отмечается, что эффективный протокол должен:

1) быстро заполнять пустующие мощности сети;

2) стремиться к равновесному состоянию сети на стационарных участках работы;

3) минимизировать потери пакетов;

4) обеспечивать справедливое распределение ресурсов между пользователями.

Разумеется, эти требования противоречивы и не всегда могут быть формализованы. В настоящее время последний пункт стоит наиболее остро, поскольку зависит от поведения внешних пользователей. Например, если два пользователя используют общий канал связи и один из них откроет TCP-соединение, а другой — два TCP-соединения, то последний получит (при других равных характеристиках) в два раза больше пропускной способности сети, поскольку алгоритм будет делить сеть между тремя, а не двумя соединениями. Второй пример связан с особенностями реализации алгоритмов TCP. Некоторые из них ведут себя «недружественно» по отношению к другим и могут отбирать у них пропускную мощность канала. Игровая природа взаимодействия пользователей в Интернете вызвала большое количество публикаций за последние 15 лет, в которых исследуются различные аспекты этих процессов (подробно см. [6] и библиографию к ней). Следует отметить, что эти задачи неявно предполагают кооперативную игру (даже если математическая постановка некооперативна), поскольку у пользователей есть общая цель — работоспособная сеть, которую они хотят использовать наилучшим образом.

Иначе стоит вопрос, если в системе появляются злоумышленники — пользователи по тем или иным причинам, желающие ухудшить работу сети. Существует много типов зловредной деятельности в сети Интернет, в данном случае остановимся только на атаках типа «отказ в обслуживании» (Denial of service attacks). Это направление имеет наиболее ярко выраженную игровую природу и представляет значительный практический интерес исходя из ежегодного ущерба, наносимого пользователям сети. Развитие Интернета, которое наблюдается в настоящее время, приводит к возникновению новых проблем управления потоками данных. Процессы, которые происходят в этой сложной структуре, требуют создания адекватных математических моделей. Пожалуй, одной из первых работ по математическому моделированию Интернета была публикация Ф. Кэлли [7]. Дальнейшие исследования можно условно разделить на следующие направления. В условиях статической постановки (сеть рассматривается в окрестности точки равновесия) исследуются модели с одним или несколькими соединениями, для которых существуют развитые методы теории математического программирования и теории игр соответственно. Динамическая постановка задачи предполагает более сложные модели (описываемые, как правило, дифференциальными уравнениями), которые относятся к области оптимального управления и дифференциальных игр

В настоящей публикации рассмотрена новая игра N участников для протокола с фиксированным окном в сети с потерями. В первом разделе описаны основные подходы к детерминистическому моделированию процессов в Интернете. Мы не претендуем на полноту, но хотим продемонстрировать главные вехи развития подходов к моделированию конфликтных процессов в Интернете по схеме: модели математического программирования, модели оптимального управления, игровые модели. Второй раздел посвящен обзору существующих типов атак на отказ, причин их появления, угроз. В третьем разделе построена игровая модель взаимодействия пользователей, загружающих на общий сервер некоторые данные. Стратегиями при этом являются скорости загрузки, на которые наложены ограничения различного характера (фазовые и интегральные). Динамика системы соответствует общепринятой потоковой модели сетевого трафика. Новыми элементами являются ограничения размера окна, характерные для динамики TCP-алгоритмов. Для этой игры найдены точки равновесия по Нэшу и условия нулевых потерь. Далее для описанной динамики поставлена и решена задача оценки уязвимости к атакам на отказ.

1. МОДЕЛИРОВАНИЕ ПРОЦЕССОВ В СЕТИ ИНТЕРНЕТ. ОБЗОР ПОДХОДОВ

Модели массового обслуживания. Исторически первым подходом к описанию сетей были модели теории массового обслуживания. В детерминистической постановке поток входных пакетов, как правило, считается постоянным. Пусть сеть состоит из N узлов, обозначим $q_i(t)$ очередь i-го узла. Поток входных пакетов, который попадает на вход узла i, описывается скоростью λ_i, возможности обработки узла — мощностью u_i, $i = 1, \ldots, N$. Динамика сети описывается системой дифференциальных уравнений

$$\dot{\bar{q}}(t) = \bar{\lambda} + B\bar{u}, \tag{1}$$

где $\bar{u} \in U$ — управление, B — матрица, описывающая структура сети. При этом $\bar{u}, \bar{\lambda}, \bar{q} \in R_+^N$, $C\bar{u} \le \bar{1}$, $\bar{q} \in Q$, где C — структурная матрица (определяет узлы с общими мощностями), Q — компакт из пространства R_+^N, который описывает возможные состояния системы исходя из физических ограничений очередей, $\bar{1}$ — единичный вектор, а неравенство понимается построчно. Модели такого типа применительно к сетям подробно разработаны в работах Ш. Мейна [8]. Основной задачей, которая ставится, является нахождение условий существования управления $u(\cdot)$, приводящего состояние системы в нуль. Если представить процесс (1) в виде игры с управлениями $\bar{\lambda}$ и $\bar{u}$, то условия, полученные Мейном, аналогичны условиям существования отображения Л.С. Понтрягина. Если задан критерий качества работы системы $J(q(\cdot), u(\cdot))$, то задача (1) превращается в задачу оптимального управления:

$$J(q(\cdot), u(\cdot)) \xrightarrow[u(\cdot)]{} \min, \ \dot{\bar{q}}(t) = \bar{\lambda} + B\bar{u}, \ \bar{u}, \bar{\lambda}, \bar{q} \in R_+^N, \ C\bar{u} \le \bar{1}, \ \bar{q} \in Q. \tag{2}$$

Для этих моделей существует хорошо развитая теория, состоящая в применении необходимых условий экстремума и принципа максимума Понтрягина.

Модель распределения ресурсов Кэлли. В [7] Ф. Кэлли , по всей видимости, впервые предложил комплексное математическое исследование принципов распределения потоков в системе, которая работает на основе оптимизации функций полезности пользователей. Применение теории выпуклого программирования позволило ему сформулировать необходимые условия существования решения — точки равновесного распределения ресурсов сети. Рассмотрим эту модель. Пусть сеть описывается набором линий связи со скоростью передачи c_l, $l = 1, \ldots, M$. Сеть совместно используется источниками, каждый из которых характеризуется скоростью x_i, $i = 1, \ldots, P$. С каждым источником (пользователем) связана функция полезности $U_i(x_i)$ и фиксированный маршрут — последовательность линий связи, по которому перемещаются пакеты данного пользователя. Пусть $r(i, j)$ — функция, равная 1, если источник i использует линию j, то введем матрицу маршрутизации R : $R = \{r_{ij} = r(i, j)\}_{\substack{i=1,\ldots,P \\ j=1,\ldots,M}}$. Задача распределения ресурсов представляется в виде нелинейной программы

$$\max_{\bar{x} \ge 0} \sum_i U_i(x_i), \ R\bar{x} \le \bar{c}, \ \bar{x} \in R_+^P. \tag{3}$$

Если $U_i(\cdot)$ — возрастающие, строго вогнутые функции, то существует единственное решение задачи (3). Представляет интерес конструктивный алгоритм нахождения решения $\bar{x}$. Введем понятие стоимости использования линии l, пусть $p_l(t) = f_l\left(\sum_s x_s(t))\right)$, где суммирование происходит по всем индексам s, кото-

рые используют линию l, а $f_l(\cdot)$ — непрерывная возрастающая функция. Тогда из решения прямой и двойственной задач следует, что $x_r = \dfrac{\partial U}{\partial x_r}\left(\sum_s p_s\right)$, где суммирование идет по всем линиям маршрута источника r. При этом для нахождения своего лучшего решения пользователю r необходимо знать только загруженность вдоль его маршрута, что очень важно для практической реализации. В работе также показано, что если функции полезности имеют вид $U_s(x_s) = \log x_s$, то оптимальное решение пропорционально справедливо. Вектор $\hat{x}$ называется пропорционально справедливым, если для любого другого допустимого вектора скоростей x выполняется неравенство $\sum_r \dfrac{x_r - \hat{x}_r}{\hat{x}_r} \le 0$.

Подход Кэли позволил найти условия существования точки равновесия сети со многими пользователями и предложил конструктивный распределенный алгоритм ее достижения. Позднее в работах [9, 10] этот подход получил развитие в направлении теории распределенного управления в сетях. В частности, в [9] предложен революционный алгоритм управлением переполнениями на основании задержек. Дело в том, что одним из фундаментальных ограничений улучшения работы TCP являлась бинарность обратной связи. Потерю пакета может вызвать случайный сбой оборудования, краткосрочная флуктуация, перегрузка сети. Даже в идеальном случае состояние системы под управлением этого протокола колеблется возле точки равновесия, никогда в нее не попадая. Если же взять за основу величину RTT и рассмотреть его изменение во времени, то возрастание задержек будет пропорционально загруженности сети. Созданный на основе этого алгоритм управления окном [10] не только решает проблему коллапса переполнения, но и является пропорционально справедливым (если в сети все пользователи будут использовать этот протокол). К сожалению, сосуществование алгоритмов TCP с обратной связью на основании потерь и на основании задержек приводит к несправедливому распределению ресурсов. Совмещение этих подходов — одна из актуальных проблем современной теории.

2. МОДЕЛИРОВАНИЕ ПРОЦЕССОВ В СЕТИ ИНТЕРНЕТ В УСЛОВИЯХ КОНФЛИКТА

Игровые модели. На данный момент существует множество публикаций, связанных с применением теории игр при моделировании процессов в Интернете, как в кооперативной [11], так и в некооперативной [12] постановках. Центральным подходом при этом является исследование точек равновесия системы по Нэшу [12], Штакельбергу [13], Вардропу [14]. Опишем достаточно общую игровую постановку задачи [6]. Обозначим пространство стратегий $\Omega \subset R^N$, в каждый момент времени игроки выбирают свои действия, формируя вектор $x \in \Omega$. С каждым игроком свяжем индекс $i \in \{1, \ldots, N\}$ и функцию $J_i(\alpha_i, x) = \alpha_i p_i(x) - U_i(x)$. Цель каждого игрока состоит в минимизации своей функции $J_i(\cdot)$. Здесь функция $p_i(\cdot)$ — это мера загруженности или стоимости использования сети для i-го игрока, $U_i(x)$ — полезность, α_i — параметр, используемый для улучшения точки равновесия. При выполнении определенных ограничений на функции эта игра имеет по крайней мере одно состояние равновесия по Нэшу. Как показано в работе [15], большое количество практических задач управления в сетях может быть представлено в виде такой игры.

Исследуется также расширение этой игры на динамический случай. Распространенным подходом при моделировании является рассмотрение работы узкого места сети. Пусть мерой загруженности сети будет суммарная задержка

вдоль маршрута следования пакетов пользователя $D_i(t) = \sum_l d_l(t)$, тогда динамика системы описывается уравнениями

$$\dot{x}_i(t) = \frac{dU_i(x_i)}{dx_i} - \alpha_i D_i(t),\ \dot{d}_l(t) = \frac{x_l}{C_l} - 1.$$

Атаки на отказ. Комитет по национальной безопасности США компьютерную атаку определяет как любую форму злоумышленной деятельности с целью собрать, исказить, отключить или уничтожить ресурсы информационной сети или саму информацию.

Атака типа «отказ в обслуживании» — это атака на вычислительную систему с целью довести ее до отказа, т.е. создание таких условий, при которых пользователи системы не могут получить доступ к ее ресурсам. Принципиально такая атака может быть осуществлена двумя способами. Первый использует ошибки программного обеспечения, служб и протоколов. Применение специально подготовленных (например, некорректных) пакетов позволяет значительно усложнить или заблокировать работу системы. Второй способ связан с использованием большого количества корректного с точки зрения протокола, но бессмысленного трафика для загрузки необходимых пользователям ресурсов.

Часто целью таких атак является получение выгоды. Ресурс шантажируется возможностью запуска атаки, которая приведет к его недоступности для пользователей. Однако существует также много примеров атак сайтов государственных учреждений, банков. Для организации крупной атаки злоумышленнику необходимо взять под свой контроль армию ботов — промежуточных компьютеров, которые будут непосредственно осуществлять пересылку пакетов. Это осуществляется с помощью червей или вирусов, пользователь может и не подозревать, что его компьютер заражен вредоносным кодом. Атаки такого типа называются распределенными, и их мощность возрастает пропорционально количеству пользователей сети.

Существует обширная литература по обнаружению, противодействию, расследованию атакующих действий (например, обзор [16]). В настоящее время моделирование атак, как правило, проводится имитационными средствами. Аналитические модели атак, основанных на первом подходе, неэффективны, поскольку они в некотором смысле «одноразовые». Поэтому нынешние модели в основном связаны с моделированием атак второго типа. Пусть известна функция стоимости атаки $U_A(\lambda_A(\cdot))$ злоумышленника, тогда его цель состоит в максимизации загрузки сети вдоль маршрута игрока i минус стоимость атаки:

$$J_A(\lambda_A(\cdot)) = p_i(\lambda_A(\cdot)) - U_A(\lambda_A(\cdot)) \to \max.$$

3. АНАЛИТИЧЕСКАЯ МОДЕЛЬ ВЗАИМОДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЕЙ

При построении аналитической модели будем считать выполненными следующие предположения:

- процесс передачи пакетов через сеть аппроксимируется непрерывными потоками;
- ACK-пакеты и информация о потерях достигает источников моментально (нулевые информационные задержки);
- пакеты перемещаются между маршрутизаторами моментально (нулевые транспортные задержки);
- у всех источников одинаковый маршрут и протокол работы.

Рассмотрим потоковую модель с одним сервером и N однотипными пользователями. Динамика системы описывается уравнениями:

$$\begin{pmatrix}\dot q_1(t)\\ \dot q_2(t)\end{pmatrix}=\begin{pmatrix}\sum\limits_{i=1}^{N}\lambda_i(t)\\ 0\end{pmatrix}-B\begin{pmatrix}u_1(t)\\ u_2(t)\end{pmatrix},\ B=\begin{pmatrix}1 & 0\\ -1 & 1\end{pmatrix},\tag{4}$$

$$q_k(t)\in[0,\, q_k^{\max}],\ u_k(t)\in[0,\, u_k^{\max}],\ k=1,2,$$

$$\lambda_i(t)\in[0,\, \lambda_i^{\max}],\ i=1,\ldots,N,\ \bar q(t_0)=0,$$

где $q_k(t)$ — размер очереди, $\lambda_i(t)$ — скорость пользователя i, $u_k(t)$ — скорость обслуживания узла k. Эта система достаточно изучена для случая постоянных λ_i (условие устойчивости работы имеет вид $\sum\limits_{i=1}^{N}\lambda_i^{\max}\le u_1^{\max}$, $u_1^{\max}\le u_2^{\max}$). Не ограничивая общности, полагаем $t_0=0$.

В данной работе динамика системы (4) обобщается на TCP подобное поведение. Вводятся функции потерь и распределение ресурсов между пользователями. Скорость пользователей ограничивается окном. Введем вспомогательные обозначения: $q_{ij}(t)$ — количество пакетов пользователя i в очереди $q_j(t)$; $u_{ij}(t)$ — часть ресурсов очереди j, которая используется для обработки пакетов пользователя i; $l_{ij}(t)$ — интенсивность потерь пакетов пользователя i.

В соответствии с алгоритмом droptail (наиболее широко представленным в современной сети Интернет) все поступившие во время переполнения очереди пакеты отбрасываются:

$$l_{ij}(t)=\begin{cases}\max\left\{0,\begin{bmatrix}\lambda_i(t)-u_{i1}\\ u_{i1}-u_{i2}\end{bmatrix}_j\right\}, & q_j(t)=q_j^{\max},\\ 0, & q_j(t)<q_j^{\max},\end{cases}\quad j=1,2.\tag{5}$$

При определении $u_{ij}(t)$ необходимо учесть, что в момент t происходит обработка пакетов, попавших в очередь в момент времени $t-\theta$. Для определения θ введем функцию $T_j(t)=\min\{\tau\in[0,t]:\tau+q_j(\tau)/u^{\max}=t\}$. Тогда FIFO-алгоритм описывается так:

$$u_{i1}(t)=\begin{cases}\dfrac{u_1^{\max}\lambda_i(T_1(t))}{\sum\limits_{i=1,\ldots,N}\lambda_i(T_1(t))}, & \sum\limits_{i=1,\ldots,N}\lambda_i(T_1(t))>u_1^{\max},\\ \sum\limits_{i=1,2}\lambda_i(T_1(t)), & \sum\limits_{i=1,2}\lambda_i(T_1(t))\le u_1^{\max},\end{cases}$$

$$u_{i2}(t)=\begin{cases}\dfrac{u_2^{\max}u_1(T_2(t))}{u_1(T_2(t))}, & u_1(T_2(t))>u_2^{\max},\\ u_1(T_2(t)), & u_1(T_2(t))\le u_2^{\max}.\end{cases}\tag{6}$$

Ограничение окна потока i означает, что в любой момент времени значение окна w_i больше или равно количеству посланных пакетов минус количество подтвержденных и потерянных:

$$\int_0^{\infty}\left[\lambda_i(t)-u_{i2}(t)-\sum_{j=1}^{2}l_{ij}(t)\right]dt\le w_i.\tag{7}$$

Введем функцию полезности

$$U_i(\lambda_i(\cdot))=\min\left\{t\ge 0:\int_0^t u_{i2}(\tau)d\tau\ge\lambda_i^{\text{int}}\right\}.\tag{8}$$

Другими словами, игроку необходимо переслать на узел 2 количество данных λ_i^{int}, после подтверждения всех пакетов он заканчивает игру. Это время и является его показателем качества. Рассмотрим систему (4)–(8) с одним игроком, тогда задача оптимального управления состоит в минимизации $U(\lambda(\cdot))$. Введем функцию $\lambda^*(t)$ следующим образом.

• Если выполнено $\lambda^{\max} \le u_1^{\max}$, то $\lambda^*(t) = \lambda^{\max}$.

• Если выполнено $\lambda^{\max} > u_1^{\max}$, $u_1^{\max} \le u_2^{\max}$, то возникает два варианта:

если $\dfrac{w}{\lambda^{\max} - u_1^{\max}} \le \dfrac{\lambda^{\text{int}}}{u_1^{\max}}$, то $\lambda^*(t) = \lambda^{\max}$,

иначе $\lambda^*(t) = \begin{cases} \lambda^{\max}, & t \in [0,\ w/(\lambda^{\max} - u_1^{\max})], \\ u^{\max}, & t \in \left(\dfrac{w}{\lambda^{\max} - u_1^{\max}}, \dfrac{\lambda^{\text{int}}}{u_1^{\max}} \right]. \end{cases}$

• Если выполнено $\lambda^{\max} > u_1^{\max}$, $u_1^{\max} > u_2^{\max}$, то аналогично:

если $\dfrac{w}{\lambda^{\max} - u_1^{\max}} \le \dfrac{\lambda^{\text{int}}}{u_1^{\max}}$ и $\dfrac{w}{u_1^{\max} - u_2^{\max}} \le \dfrac{\lambda^{\text{int}}}{u_2^{\max}}$, то $\lambda^*(t) = \lambda^{\max}$,

иначе $\lambda^*(t) = \begin{cases} \lambda^{\max}, & t \in [0,\ t_1], \\ u_1^{\max}, & t \in (t_1,\ t_2], \\ u_2^{\max}, & t \in (t_2,\ \lambda^{\text{int}} / u_2^{\max}], \end{cases}$

где $t_1 = \min \left\{ \dfrac{w}{\lambda^{\max} - u_1^{\max}}, \dfrac{w}{u_1^{\max} - u_2^{\max}} \right\}$, $t_2 = \max \left\{ \dfrac{w}{\lambda^{\max} - u_1^{\max}}, \dfrac{w}{u_1^{\max} - u_2^{\max}} \right\}$.

Утверждение 1. Существует оптимальное решение системы (4)–(8) с одним игроком и при этом $\lambda^*(t) = \arg\min\limits_{\lambda(\cdot)} U(\lambda(\cdot))$.

Доказательство этого и других утверждений содержится или близко по технике к результатам, указанным в работах [17, 18]. Рассмотрим теперь игру N пользователей.

Вероятности потерь. Важным элементом анализа динамики сетей является определение вероятности потерь и возможности их уменьшения. Следует отметить, что потери — это не только негативная характеристика, но и важная информация, необходимая для синтезирования управления. Для TCP потери являются необходимым элементом нормальной работы. В данном разделе сформулируем условия нулевых потерь в системе (4)–(8). Будем иметь в виду, что функция $l_{ij}(t)$ является интенсивностью потерь, суммарные потери на интервале $[t, T]$ равняются $\int\limits_t^T l_{ij}(\tau)d\tau$. Введем обозначение $\Lambda = \sum\limits_{i=1}^{N} \lambda_i^{\max}$.

Утверждение 2. Если выполнено одно из перечисленных ниже условий, то $\sum\limits_{i=1}^{N} \sum\limits_{j=1}^{2} l_{i1}(t) \equiv 0$ для любого момента времени $t \in [0, +\infty]$:

1) $\Lambda \le \min\{u_1^{\max}, u_2^{\max}\}$;

2) $\sum_{i=1}^{N} w_i < \min\{q_1^{\max}, q_2^{\max}\}$;

3) $\Lambda > u_1^{\max}$, $u_1^{\max} \le u_2^{\max}$, $\dfrac{q_1^{\max}\lambda_i^{\max}}{\Lambda - u_1^{\max}} \ge \lambda_i^{\text{int}}\Lambda$, $i = 1,\ldots, N$.

Равновесие по Нэшу. Покажем, что в игре (4)–(8) существует равновесие по Нэшу.

Утверждение 3. Пусть выполняется $w_i \ge \max\{q_1^{\max}, q_2^{\max}\}$, $i = 2,\ldots, N$. Тогда для фиксированных стратегий $\lambda_i^{\max}$, $i = 2,\ldots, N$, время завершения игры первого игрока T_1 может быть оценено неравенством $T_1 \le \dfrac{\lambda_1^{\text{int}}\Lambda}{\lambda_1^{\max}\min\{u_1^{\max}, u_2^{\max}\}}$. Следовательно, точка $(\lambda_1^{\max},\ldots,\lambda_N^{\max})$ будет равновесием по Нэшу в этой игре.

Замечание 1. Парето-оптимальным распределением в игре (4)–(8) будет точка $\dfrac{\min\{u_1^{\max}, u_2^{\max}\}}{\Lambda}(\lambda_1^{\max},\ldots,\lambda_N^{\max})$, эти равновесия будут совпадать только в случае выполнения условия $\Lambda \le \min\{u_1^{\max}, u_2^{\max}\}$.

Замечание 2. Условие $w_i \ge \max\{q_1^{\max}, q_2^{\max}\}$ практически означает, что ограничение (7) не играет роли.

В следующем утверждении рассматривается это ограничение.

Пусть для игры (4)–(8) выполняется $w_1 < \max\{q_1^{\max}, q_2^{\max}\}$, $w_i \ge$ $\ge \max\{q_1^{\max}, q_2^{\max}\}$, $i = 2,\ldots, N$.

Утверждение 4. Пусть $q_1^{\max} > q_2^{\max}$, $u_1^{\max} \le u_2^{\max}$. Если $\dfrac{q_1^{\max}\lambda_1^{\max}}{\Lambda} \le w_1$, то стратегия $\lambda_1(t) = \lambda_1^{\max}$ оптимальна по времени. Если же $\dfrac{q_1^{\max}\lambda_1^{\max}}{\Lambda} > w_1$, то стратегия $\lambda_1(t) = \dfrac{w_1\sum_{i=2}^{N}\lambda_i^{\max}}{q_1^{\max} - w_1}$ оптимальна по времени.

Рассмотрим атаки системы (4)–(8). Пусть злоумышленник осуществляет прямую атаку, он подает на вход системы трафик λ_A.

Утверждение 5. Предположим, что система (4)–(8) находится в точке равновесия по Нэшу $(\lambda_1^{\max},\ldots,\lambda_N^{\max})$ и выполнены условия утверждения 3. Ухудшение времени окончания игры пользователя i при прямой атаке оценивается неравенством $T_1(\lambda_A) \le \dfrac{\lambda_1^{\text{int}}(\Lambda + \lambda_A)}{\lambda_1^{\max}\min\{u_1^{\max}, u_2^{\max}\}}$.

Еще одной возможностью является атака фальшивыми АСК-пакетами, они более приоритетны, и сеть обслуживает их в первую очередь. В этом случае ресурсы сервера уменьшаются . Поскольку пакеты АСК идут от сервера к пользователю, такие атаки назвали обратными.

Утверждение 6. Предположим, что система (4)–(8) находится в точке равновесия по Нэшу $(\lambda_1^{\max},\ldots,\lambda_N^{\max})$ и выполнены условия утверждения 3. Ухудшение времени окончания игры пользователя i при обратной атаке оценивается неравенством $T_1(\lambda_A) \le \dfrac{\lambda_1^{\text{int}}(\Lambda)}{\lambda_1^{\max}\min\{u_1^{\max}, u_2^{\max} - \lambda_A\}}$.

ЗАКЛЮЧЕНИЕ

В данной работе рассматриваются модели конфликтного управления в Интернете, представлен обзор проблемы, актуальность, описаны существующие подходы к ее решению. Поставлена игровая задача взаимодействия пользователей при условии злоумышленных действий, направленных на ухудшение работы сети — атак на отказ. Построена модель одного класса сетей с фиксированным окном доступа и показано существование оптимального в смысле быстродействия решения. Найдены условия возникновения потерь и точка равновесия по Нэшу. Для некоторых распространенных типов атак оценена уязвимость системы.

СПИСОК ЛИТЕРАТУРЫ

1. Jacobson V. Congestion avoidance and control // ACM Comput. Com. Rev. — 1988. — **18**. — P. 314–329.
2. Nagle J. RFC896 — Congestion control in IP/TCP internetworks // RFC, 1984.
3. Chiu D.M., Jain R. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks // Comput. Networks and ISDN Systems. — 1989. — **17**. — P. 1–14.
4. Afanasyev A., Tilley N., Reiher P., Kleinrock L. Host-to-host congestion control for TCP // Commun. Surveys Tuts. — 2010. — **12**, N 3. — P. 304–342.
5. Welzl M. Network congestion control: Man-aging internet traffic. — Wiley, 2005. — 263 p.
6. Alpcan T. Noncooperative games for control of networked systems // Ph.D. Thesis, Un-it of Illinois at Urbana-Champaign, Urbana, IL, May 2006.
7. Kelly F.P. Charging and rate control for elastic traffic // Europ. Trans. on Telecom. — 1997. — **8**. — P. 33–37.
8. Meyn S. Control techniques for complex networks. — Cambridge Un-t Press, 2007. — 582 p.
9. Mo J., Walrand J. Fair end-to-end window-based congestion control // IEEE/ACM Transact. on Network. — 2000. — **8**. — P. 556–567.
10. Paganini F., Doyle J.C., Low S.H. Scalable laws for stable network congestion control // Proc. of IEEE Conf. on Decision and Contr. — 2001. — **1**. — P. 185–190.
11. Yaiche H., Mazumdar R.R., Rosenberg C. A game theoretic framework for bandwidth allocation and pricing in broadband networks // IEEE/ACM Transact. on Network. — 2000. — **8**. — P. 667–678.
12. Altman E., Basar T., Jimenez T. Shimkin N. Competitive routing in networks with polynomial costs // IEEE Transact. on Automat. Contr. — 2002. — **47**, N 1. — P. 92–96.
13. Garg R., Kamra A., Khurana V. A game-theoretic approach towards congestion control in communication networks // ACM SIGCOMM Comput. Com. Rev. — 2002. — **32**, N 3. — P. 47–61.
14. Altman E. et al. A survey on networking games in telecommunications // Comput. & Oper. Res. — 2006. — **33**, N 2. — P. 286–311.
15. Alpcan T., Pavel L., Stefanovic N. An optimization and control theoretic approach to noncooperative game design // arXiv preprint arXiv:1007.0144. — 2010.
16. Андон П.І., Ігнатенко О.П. Атаки на відмову в мережі Інтернет: опис проблеми та підходів щодо її вирішення. — Київ, 2008. — 50 с. (Препр. / Ін-т програмних систем НАН України).
17. Андон П.І., Ігнатенко О.П. Потокові моделі мережі Інтернет за умов атак на відмову // Проблеми програмування. — 2012. — № 2–3. — С. 86–96.
18. Ігнатенко О.П. Одна динамічна конфліктно керована модель взаємодії користувачів у відкритих інформаційних середовищах // Там же. — 2012. — № 4. — С. 50–63.

Поступила 04.02.2013