DOI: 10.25728/mlsd.2023.1220

МОДЕЛИРОВАНИЕ РЕГУЛЯТОРА ДЛЯ ЗАЩИТЫ ОТ АТАК ТИПА «ОТКАЗ В ОБСЛУЖИВАНИИ» МЕТОДОМ NETWORK CALCULUS[1]

Промыслов В.Г., Тимофеев М.Ю.
Институт проблем управления им. В.А. Трапезникова РАН, Москва, Россия
vp@ipu.ru, timof-1964@mail.ru,
Фонарева О.Е.
Институт проблем управления им. В.А. Трапезникова РАН, Москва, Россия
Россия, Москва
oefonareva@edu.hse.ru

Аннотация. Работа рассматривает модель регулятора сетевого потока со статическим окном, в качестве меры защиты от атак на доступность, типа атаки отказ в обслуживании. Модель построена на базе метода сетевых исчислений «Network Calculus», что позволяет получить детерминированные оценки на характеристики регулятора и потока.

Ключевые слова: Network Calculus, DoS, сетевые исчисления, модель, регулятор, мера защиты.

Введение

Информационные системы в современном мире являются цифровыми системами и как следствие они подвержены кибератакам. Атаки обычно классифицируют в зависимости от свойств кибербезопасности/конфиденциальность/целостность/доступность в модели КЦД, которые они нарушают [1]. Приоритетность целей в модели КЦД может меняться в зависимости от объекта атаки, например, для банковских и обще информационных систем бэк офиса наиболее важным свойством может быть конфиденциальность, для промышленных систем это чаще всего либо целостность или доступность. Каждая из трех указанных свойств безопасности, может подвергаться специфическим видам атак, для промышленных систем с высоким приоритетом доступности наиболее опасными являются атаки типа отказа в обслуживании (Denial of Service, DoS-атака) [2]. DoS атаки часто так же используют и для информационных систем в виду простоты их организации для злоумышленника.

В общем случае принято относить DoS-атаки к сетевым атакам, однако DoS атаки так же часто разделяют по технике реализации [3], и по уровню сетевого протокола, на котором они воздействуют на систему [4] Для описания уровней протокола обычно используется сетевая модель OSI (The Open Systems Interconnection model) серии стандартов ISO/IEC 7498. Модель определяет уровни взаимодействия систем. Схема модели представлена на рисунке 1. DoS-атаки могут совершаться на прикладном уровне, уровне представления, транспортном и сетевом, сеансовый канальный и физические уровни практически не применяется в DoS-атаках.

Рис. 1. Схема OSI-модели

[1] *Исследование выполнено за счет гранта Российского научного фонда № 23-19-00338, https://rscf.ru/project/23-19-00338/*

Наиболее комплексные DoS атаки, с использованием информации об архитектуре атакуемой системы и ее функциях, можно проводить на прикладном уровне. Прикладной уровень представляет наиболее широкий вектор атаки для злоумышленника, так как прикладной уровень обеспечивает взаимодействие пользовательских приложений с сетью, позволяет приложениям использовать сетевые службы; отвечает за передачу служебной информации; предоставляет приложениям информацию об ошибках; формирует запросы к уровню представления. Примерами атак на прикладной уровень являются следующие виды атак:

- BGP угон. Злоумышленник злонамеренно перенаправляет интернет-трафик, ложно объявляя о владении блоками IP-адресов. Поскольку другие сети принимают эту ложную информацию, трафик перенаправляется злоумышленнику. BGP угон может иметь ряд мотивов, включая перехват интернет-трафика и перенаправление его на фальшивый веб-сайт в рамках атаки «человек посередине».
- Slow HTTP POST. Злоумышленник отправляет POST заголовок с легитимным полем «Content-Length», которое позволяет веб серверу понять, какой объём данных к нему поступает. Как только заголовок отправлен, тело POST сообщения начинает передаваться с очень медленной скоростью, что позволяет использовать ресурсы сервера намного дольше, чем это необходимо, и, как следствие, помешать обработке других запросов.
- HTTP flood. В данном случае осуществляется попытка вызова отказа системы путем отправки множества HTTP-пакетов к целевому серверу. Множество нелегитимных HTTP-пакетов вызывают переполнение пропускающей полосы и, как следствие, сбой в работе системы.

Защита от DoS-атак на прикладном уровне сильно разнится в зависимости от имеющейся инфраструктуры. Основные положения для предотвращения: минимизация вектора атаки и настройка мониторинга сети [4].

Уровень представления. Обеспечивает преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с прикладного уровня, на уровне представления преобразуются в формат для передачи по сети, а полученные из сети данные преобразуются в формат приложений. На этом уровне может осуществляться сжатие/распаковка или шифрование/дешифрование, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально. Примерами атак на уровне представления являются атаки вида: ложных SSL запросов. Проверка шифрованных SSL пакетов очень ресурсоемка, злоумышленники используют SSL для HTTP-атак на сервер жертвы. В следствие чего атакуемые системы могут перестать принимать SSL соединения или автоматически перегружаться.

Транспортный уровень. Обеспечивает доставку информации по каналам внешней сети. Блоки данных в данном случае делятся на отдельные фрагменты, размеры которых будут зависеть от используемого протокола. Примерами DoS атак на данном уровне являются атаки следующего типа:

- SYN flood. Атаки типа SYN-flood основываются на некоторых особенностях «тройственного рукопожатия» в результате установки соединения [5]. На каждый входящий пакет программной системе необходимо зарезервировать ресурсы в памяти SYN, сгенерировать ответ SYN+ACK, осуществить поиск в таблицах сессий и т. д. На выполнение данных операций программная система расходует свои ресурсы. В результаты наступает отказ в обслуживании [3]. В качестве мер защиты возможно и настраивать пределы по количеству SYN-пакетов в секунду и блокировать через firewall каналы, где превышен порог.
- ICMP(v6)-flood. В данном виде атак задействуется механизм эхо-ответа. При атаке ICMP(v6)-flood злоумышленник осуществляет отправку системе-жертве большого количества пакетов с различными недействительными поддельными IP-адресами источника. Это приводит к неэффективной трате ресурсов системы жертвы. Атака значительно уменьшает пропускную способность сети, в результате чего настоящие пакеты данных не могут быть обработаны системой [6].
- UDP flooding. Заключается в отправке множества UDP-пакетов (как правило, большого объёма) на определённые или случайные номера портов удалённого хоста, который для каждого полученного пакета должен определить соответствующее приложение, убедиться в отсутствии его активности и отправить ответное ICMP-сообщение «адресат недоступен» [2]. В итоге атакуемая система окажется перегруженной. Так как сервисы, работающие через UDP обычно потоковые, для защиты от атаки можно установить фильтрацию на основе длины поступающих пакетов. Некоторые провайдеры закрывают UDP, размещая свои DNS-сервера внутри сети.

Сетевой уровень. Предназначен для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и

маршрутизацию, отслеживание неполадок и «заторов» в сети. Примерами DoS атак на данном уровне являются атаки следующего типа:

- Атака с отражением UDP-пакетов. Злоумышленник посылает короткие UDP-пакеты на порт 19 одного из компьютеров в сети, подменив IP-адрес и порт источника. В результате сеть на отрезке между двумя компьютерами перегружается, что может отразиться на её производительности в целом. Защита осуществляется аналогично случаю с UDP-flooding.
- Ping-flood. Злоумышленник отправляет большое количество ping-запросов на сервер одновременно.

Практически для всех типов атак (исключая прикладной уровень) для защиты от DoS-атак можно использовать методы Traffic Shaping и Traffic Policer [7]. Traffic Shaping ограничивает скорость путём буферизации лишнего трафика. Для примера, есть некоторый установленный лимит скорости трафика *A*. Если скорость трафика меньше этого значения, он будет проходить через *shaper* без изменений, в случае, если скорость превысит значение *A*, пакеты начнут скапливаться в буфере. Работа данного метода проиллюстрирована на рисунке 2a.

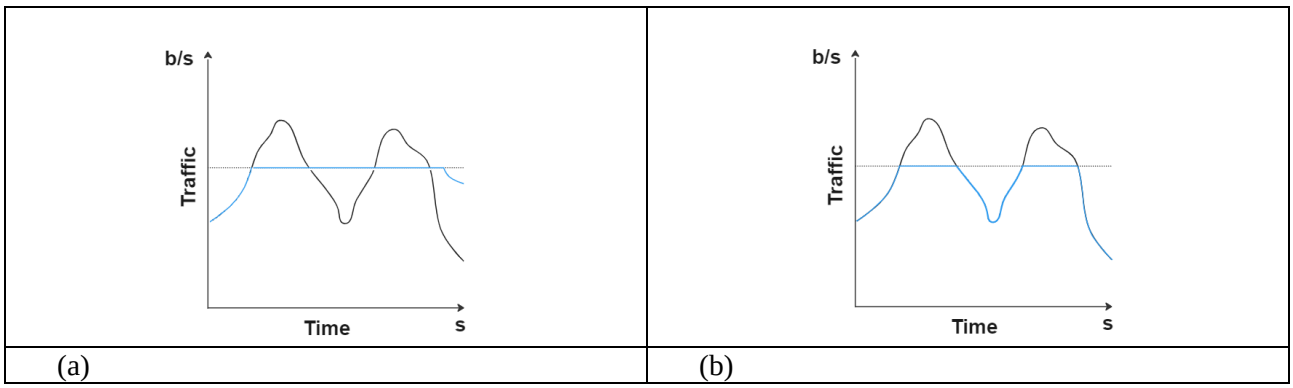


Рис. 2. График скорости сетевого трафика, проходящего через Shaper (a) и Policer (b)

Метод *Traffic Policer* похож на *Traffic Shaping*, но в отличие от второго, этот метод, в случае превышения допустимого лимита скорости *A*, отбрасывает «лишние» пакеты. Работа метода проиллюстрирована на рисунке 2b.

Для проектирования средств защиты от DoS атак, разработки стратегий по управлениям инцидентами безопасности, часто используют модели, в том числе модели Shaper. Модели могут быть как аналитическими [8], так и имитационными [9]. Одним из видов аналитических моделей являются модели на основе сетевых исчислений “Network Calculus” [10]. Данный метод благодаря детерминированному подходу позволяющим получить ограничения на основные характеристики системы, нашел широкое применение в сетевых моделях, в частности для моделирования устройств типа Shaper [11], однако они в основном используют модели без петли обратной связи, например модель типа Time-Aware Shaper (TAS) или Credit Based Shaper [8,11]. В работе рассмотрена простая модель устройства типа Shaper с регулятором с использованием метода сетевых исчислений, позволяющая оценить поток на выходе устройства при изменении потока входе и подобрать параметры регулятора.

Предлагаемая модель является расширением модели Network Calculus регулятора со статическим окном [12]. В работах [13,14] анализируется свойства системы с таким регулятором: в частности, ограничения на задержку передачи сигнала и схема с последовательно соединенными регуляторами соответственно. Однако все работы сосредоточены на свойствах регулятора позволяющие определить ограничения на задержку и размер буфера данных в системе, поэтому внимание уделяется рассмотрению минимальной кривой обслуживания, которая задает нижнюю границу выходного потока для системы с регулятором, вместе с этим для корректного моделирования поведения системы при DoS атаке важно, как верхняя, так и нижняя граница для выходного потока. Нижняя граница служит для расчета максимальные ограничения на задержку и буфер в регуляторе, а верхняя граница определяет конверт выходного потока.

1. Применение Network Calculus для моделирования сетевых потоков

Для моделирования сетевых потоков используется сетевое исчисление («Network Calculus»), описанное в книге [12]. Основа сетевого исчисления лежит в математической теории мини/макси-плюс

алгебры [15]. Далее все используемые уравнения сетевого исчисления будут выражены через определения мини-плюс алгебры, с учетом существующего изоморфизма мини и макси плюс алгебры [16].

С помощью сетевого исчисления можно анализировать некоторые фундаментальные свойства сетей, получив оценку граничных характеристик в виде минимальных и максимальных величин на изменение потока, задержку обработки данных и размера буфера в системе. В разделе приведены только сведения необходимые для понимания модели регулятора, для более полного ознакомления следует изучить литературу.

В теории фильтров и в теории линейных систем важную роль играет оператор $\otimes$ свёртки одномерных функций f и g определённый как:

$$(f \otimes g)(t) = \int_{-\infty}^{+\infty} f(\tau) \cdot g(t-\tau) d\tau$$

В сетевых исчислениях обычно рассматриваются потоки, описываемые функциями специального вида:

$$a: R \to R \cup \{+\infty\},$$

$$a(t) = 0, t < 0;$$

$$a(t) \leq a(s), \forall t < s$$

и для них существует аналог обычной свертки, который так же играет большую роль при моделировании потоков данных в системах, причем в определении свертки сумма заменяется минимумом, а произведение заменяется суммой и оператор мини-плюс свертки представлен выражением:

$$(f \otimes g)(t) = \inf_{0 \leq \tau \leq t} \{f(\tau) + g(t-\tau)\}$$

В мини плюс алгебре, так же существует оператор, обратный свёртке, обозначается как

$$(f \oslash g)(t) = \sup_{\tau \geq 0} \{f(t+\tau) - g(\tau)\}$$

Одно из важных применений оператора обратной свертки, это вычисление конверта потока.

Определение 1. Функция e является «конвертом» потока a, если $a \leq a \otimes e$.

Определение 2. Функция e является минимальным «конвертом» потока a, если $e = a \oslash a$.

В рамках сетевых исчислений система описывается двумя функциями называемыми кривыми обслуживания, которые задаются через операции мини плюс свертки.

Определение 3. Система будет иметь минимальную кривую обслуживания $\underset{\smile}{s}$, если она является функцией потока, а выходной поток системы b удовлетворяет условиям:

$$b \geq a \otimes \underset{\smile}{s}.$$

Определение 4. Система будет иметь максимальную кривую обслуживания $\overset{\frown}{s}$, если она является функцией потока, а выходной поток системы b удовлетворяет условиям:

$$b \leq a \otimes \overset{\frown}{s}.$$

Определение 5. (Полуаддитивное замыкание): пусть функция $f: \mathbb{R} \to \mathbb{R}_+ \cup \{+\infty\}$. Обозначим $f^{(n)}$ как результат $(n-1)$ последовательныз сверток функции f с собой при условии, что $f^{(0)} = \delta_0$,

$$\delta_T(t) = \begin{cases} +\infty, t \geq T \\ 0, t < T \end{cases}, \qquad T \geq 0$$

Тогда полуаддитивное замыкание функции f обозначенное как $\bar{f}$:

$$\bar{f} = \inf_{n \geq 0} \{f^{(n)}\}$$

или

$$\bar{f} = \delta_0 \wedge f \wedge (f \otimes f) \wedge (f \otimes f \otimes f) \wedge \ldots$$

2. Модель регулятора для защиты от DoS атак

Рассмотрим модель устройства реализующего формирователя для потока типа Shaper. Для чего расширим основные положения модели регулятора, которые были приведены в работе [12] для случая максимальной и минимальной кривой обслуживания.

Пусть входящий поток данных $a(t)$ проходит через регулятор, его схема показана на рисунке 3.

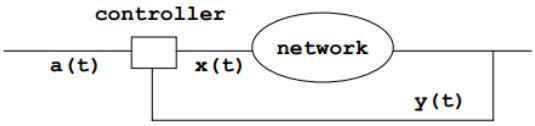


Рис. 3. Схема регулятора со статическим окном

Учитывая определения 3,4 можно увидеть, что сетевой элемент за регулятором обеспечивает трансформацию входного потока $x(t)$ в выходной поток $y(t)$ в соответствии с выражениями:

$$y(t) \le x \otimes \overset{\frown}{\beta}. \tag{1}$$

$$y(t) \ge x \otimes \underset{\smile}{\beta} \tag{2}$$

где $\overset{\frown}{\beta}$ и $\underset{\smile}{\beta}$ максимальная и минимальная кривые обслуживания соответственно, так как выражения (1,2) задают только границы для выходного потока, в общем случае это можно выразить как:

$$\Pi: x \to y = \Pi(x) \tag{3}$$

и сравнивая (1,2) с (3)

$$\Pi(x) \ge \mathrm{C}_{\underset{\smile}{\beta}}(x) \tag{4.1}$$

$$\Pi(x) \le \mathrm{C}_{\overset{\frown}{\beta}}(x) \tag{4.2}$$

где $\mathrm{C}_{\beta}(x)$ оператор мини-плюс свертки [5], $\beta \in \{\overset{\frown}{\beta}, \underset{\smile}{\beta}\}$. Регулятор ограничивает объем данных, на выходе системы следующим образом:

$$\begin{cases} x(t) \le A(t) \\ x(t) \le \Pi(x) + W \end{cases} \tag{5}$$

где $W = const$ – размер окна буферизируемых в сети данных. Обозначим максимальный поток $x(t)$на выходе регулятора x_{max}, можно показать [6] что:

$$x_{max}(t) = \overline{(\Pi + W)}(a)(t) \tag{6}$$

$$\overline{(\mathrm{C}_{\beta} + W)}(a) = \overline{(\mathrm{C}_{\beta+w})}(a) = \mathrm{C}_{\overline{\beta+w}}(a) = \overline{(\beta + W)} \otimes a$$

где черта сверху показывает полуаддитивное замыкание. Обозначим $\beta_w = \overline{\beta + w}$

Соответственно объединяя (6) последовательно с (1) и (2)

$$y(t) \ge (\underset{\smile}{\beta} \otimes x)(t) = (a \otimes \underset{\smile}{\beta_{cl}})(t) \tag{7.1}$$

$$y(t) \le (\overset{\frown}{\beta} \otimes x)(t) = (a \otimes \overset{\frown}{\beta_{cl}})(t) \tag{7.2}$$

где $\beta_{cl} = \beta \otimes \beta_w$, $\beta \in \{\overset{\frown}{\beta}, \underset{\smile}{\beta}\}$ соответствующая кривая обслуживания замкнутой системы при замкнутой петли обратной связи.

Уравнения (7.1) и (7.2) позволяют оценить границы, в которых может находиться выходной поток при заданных характеристиках входного потока. Заметим, что уравнения (7.1) и (7.2) можно формально рассматривать независимо и для каждого использовать собственный размер окна (W_1, W_1), при условии сохранения соотношения (1) и (2) для системы с замкнутой петлей обратной связи.

Переход от потока к его конверту, оставляет уравнения 7.1,2 истинными, в силу монотонности оператора свертки $\otimes$, если для расчета потока вместе с максимальной и минимальной кривой

обслуживании в уравнениях (7.1) – (7.2) использовать максимальный и минимальный конверт потока соответственно.

Однако, с учетом теоремы 1.6.2 [5], можно объединить уравнения 7.1,2 и конверт выходного представить как:

$e_y(t) = ((e_a \otimes \overleftarrow{\beta_{cl}}) \oslash \underbrace{\beta_{cl}})(t)$, где e_a, e_y конверты входного и выходного потока соответственно.

Рассмотрим несколько примеров, моделирующих поведение системы при изменении потока на входе при фиксированных параметрах системы (Пример 1) и при фиксированном потоке на входе и изменении значения окна в системе (Пример 2).

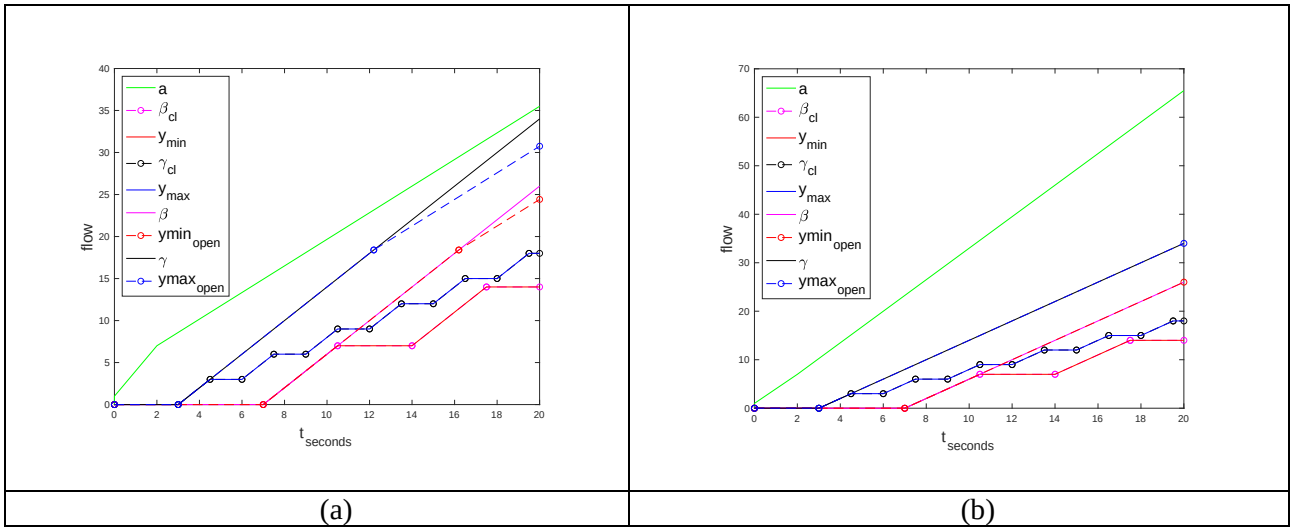
В примерах используются обозначения:

a-входной поток;

γ_{cl}, β_{cl}-максимальная и минимальная кривая обслуживания при замкнутой цепи обратной связи регулятора соответственно;

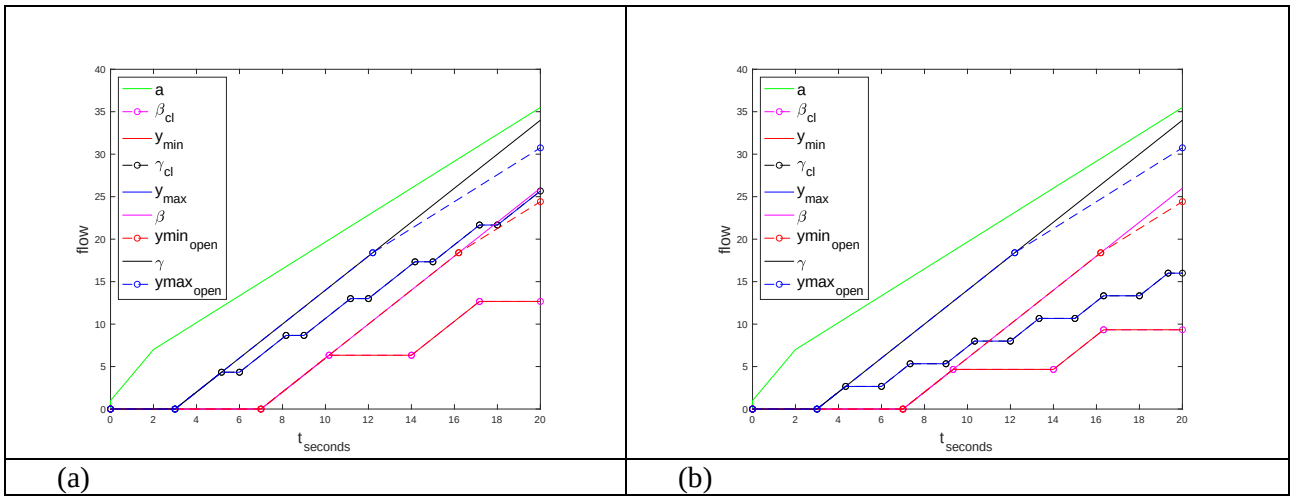
γ ,β -максимальная и минимальная кривая обслуживания при разомкнутой цепи обратной связи регулятора соответственно;

А так же границы выходных потоков: $y_{min} = \beta_{cl} \otimes a$, $y_{max} = \gamma_{cl} \otimes a$, $ymin_{open} = \beta \otimes a$, $ymax_{open} = \gamma \otimes a$.



(a) (b)

Рис. 4. Пример 1. График входного и выходного потоков ограниченного минимальной и максимальной кривой обслуживания для случая с замыканием и без замыкания цепи обратной связи в регуляторе для двух входных потоков (a) и (b) разной интенсивности



(a) (b)

Рис. 5. Пример 2. График входного и выходного потоков ограниченного минимальной и максимальной кривой обслуживания для случая с замыканием и без замыкания цепи обратной связи в регуляторе для двух пар значений статических окон W (a) и (b) разной интенсивности

По поведению y_{min} и y_{max} с регулятором с разомкнутой цепью обратной связи можно видеть, что увеличение интенсивности входного потока (рисунок 4) фактически не влияет на, максимальную границу выходного потока, которая полностью определяется кривой обслуживания, если входной поток превышает возможности его обработки в системе, в то же время замыкание цепи обратной связи, приводит к тому, что регулятор «заваливает» коридор, ограничивающий выходной поток в сторону уменьшения его интенсивности.

Изменение величин статических окон в модели позволяет управлять как наклоном коридора, в котором лежит выходной поток (рисунок 5), так и его шириной. Увеличение статического окна приводит к большему «заваливанию» соответствующей границы коридора.

5. Заключение

Атаки типа отказа в обслуживании (DoS) являются серьёзной проблемой, решаемой для сохранения свойств доступности любой сетевой системы. Для уменьшения риска нарушения доступности выше определенного предела, необходима реализация набора мер защиты, в частности ими могут быть различные виды форматирующих входной поток устройств (Shaper). Это позволяет обеспечить стабильную работу сети без длительных задержек и перегрузки сетевого канала. Так как регулятор со статическим окном рассчитан на ограничение входного потока, он способен защищать систему от DoS-атак типа “flood”, рассчитанных на переполнение сетевого канала (SYN/UDP/ICMP/Ping-flooding). Однако применение таких устройств должно быть обосновано с точки зрения их влияния на характеристики всей системы, в том числе при экстремальных видах работы, во время атак на систему. Предлагаемая модель, позволяет оценить вид выходного потока на входе в систему при различных параметрах внешнего потока на входе регулятора и выбрать параметры самого регулятора.

Литература

1. *Saltzer J. H., Schroeder M. D.* The Protection of Information in Computer Systems [Электронный ресурс]. URL: http://www.acsac.org/secshelf/papers/protection_information.pdf (дата обращения: 30.05.2023).
2. *Gu Q., Liu P.* Denial of Service Attacks. // Handbook of Computer Networks: Distributed Networks, Network Planning, Control, Management, and New Trends and Applications, Volume 3. – 2012.
3. Network Denial of Service https://attack.mitre.org/techniques/T1498/. (дата обращения: 30.05.2023)
4. *Басканов А.Н.* Способы противодействия и средства раннего выявления DDoS–атак. // Экономика и качество систем связи. – 2019.
5. *Prathibha R. C. and Rejimol Robinson R. R.* A Comparative Study of Defense Mechanisms against SYN Flooding Attack. International Journal of Computer Applications 98(18):16-21, July 2014.
6. *Zeb K., Baig O., Asif M.K.* DDoS Attacks and Countermeasures in Cyberspace. // IEEE 2nd World Symposium on Web Applications and Networking (WSWAN). – 2015. – Mar. – p.1-6.
7. *Barreiros M., Lundqvist P.*, "Policing and Shaping," in QOS-Enabled Networks: Tools and Foundations, Wiley, 2015, pp.101-116, doi: 10.1002/9781119109136.ch6.
8. *Docquier T., Y. -Q. Song, V. Chevrier, L. Pontnau and A. Ahmed-Nacer*, "IEC 61850 over TSN: traffic mapping and delay analysis of GOOSE traffic," 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 2020, pp. 246-253, doi: 10.1109/ETFA46521.2020.9212159.
9. *Zhao, Luxi & Pop, Paul & Steinhorst, Sebastian*. (2021). Quantitative Performance Comparison of Various Traffic Shapers in Time-Sensitive Networking.
10. *Agrawal, R, Cruz, R. L., Okino, C., and Rajan, R*. (1999). Performance bounds for flow control protocols. *IEEE/ACM Transactions on Networking (7)*, vol. 3, pages 310–323, June.
11. *Nascimento D. A., Bondorf S. and D. R. Campelo*, "Modeling and Analysis of Time-Aware Shaper on Half-Duplex Ethernet PLCA Multidrop," in IEEE Transactions on Communications, vol. 71, no. 4, pp. 2216-2229, April 2023, doi: 10.1109/TCOMM.2023.3246080.
12. *Le Boudec J., Thiran P.* Network Calculus A Theory of Deterministic Queuing Systems for the Internet. – 2019.
13. *Promyslov V., Semenkov K*. Assessment of deterministic delay bounds for a DoS-attack prevention device with a static window flow control // IFAC-PapersOnLine. 2020. vol. 53, iss. 2. C. 11089-11093.
14. *Promyslov V., Promyslova O.* DoS Attack Prevention Using the Static Window Flow Control in Sequent Connected Devices / Proceedings of the 14th International Conference "Management of Large-Scale System Development" (MLSD). Moscow: IEEE, 2021. C. https://ieeexplore.ieee.org/document/9600137.
15. *Simon, I.* "Recognizable sets with multiplicities in the tropical semiring". Mathematical Foundations of Computer Science 1988. Lecture Notes in Computer Science. Vol. 324. pp. 107–120.
16. *Liebeherr J.* “Duality of the Max-Plus and Min-Plus Network Calculus.” Found. Trends Network , 2017: pp. 139-282.