ISSN 2307-1303

ЭЛЕКТРОННЫЙ НАУЧНЫЙ ЖУРНАЛ СПБГУТ

ВЫПУСК 2

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ

ON-LINE JOURNAL IT & TELECOM

2013

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ

Учредитель и издатель федеральное государственное образовательное бюджетное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича» (СПбГУТ)

Settler and publisher Federal State Educational Budget-Financed Institution of Higher Vocational Education «The Bonch-Bruevich Saint - Petersburg State University of Telecommunications»

Адрес учредителя, издателя и редакции
Россия, 191186, Санкт-Петербург, наб. р. Мойки, д. 61, e-mail: telecomsut@gmail.com

Address of the Settler and Publisher, Editorial Office
Naberezhnaya reki Moika, 61, 191186, Saint-Petersburg, Russia, e-mail: telecomsut@gmail.com

Электронное представительство журнала - Electronic representative of the Journal
www.itt.sut.ru

Журнал зарегистрирован в Российском индексе научного цитирования (РИНЦ)
The Journal is included in Russian Index of Scientidic Quotation (RINTS)

Журнал имеет институт рецензирования.
The Journal has review institute

Журнал распространяется через электронное представительство без ограничений и по адресно-целевой подписке через редакцию
The Journal is distributed by subscription through the editorial and electronic representative

Электронное издание. Цена свободная - On-line magazin. Free price

Минимальные системные требования для просмотра сайта www.itt.sut.ru
Тип компьютера, процессор, сопроцессор, частота: Pentium IV и выше / аналогичное; оперативная память (RAM): 256 Мб и выше; необходимо на винчестере: не менее 64 Мб; ОС MacOS, Windows (XP, Vista, 7) / аналогичное; видеосистема: встроенная; дополнительное ПО: Adobe Reader версия от 7.X или аналогичное. Защита от незаконного распространения: реализуется встроенными средствами Adobe Acrobat.

СОДЕРЖАНИЕ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 621.395

И. А. Богданов, А. Е. Кучерявый

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича

АНАЛИЗ ОСОБЕННОСТЕЙ ОБЕСПЕЧЕНИЯ СЕТЕВОЙ БЕЗОПАСНОСТИ ВО ВСЕПРОНИКАЮЩИХ СЕНСОРНЫХ СЕТЯХ

всепроникающие сенсорные сети, сетевая безопасность, потоки ложных событий

Всепроникающие (беспроводные) сенсорные сети являются самоорганизующимися сенсорными сетями структурно построенными, как правило, с помощью кластеризации и состоящими из очень большого числа сенсорных узлов [1–3]. Основными характеристиками таких сетей являются жизненный цикл и доля покрытия пространства в течение этого жизненного цикла, что определяет возможности всепроникающей сенсорной сети по реализации задач мониторинга процессов, состояний, явлений и т. д. Кроме того, вследствие большого числа узлов в кластере и большого числа кластеров в сети, а также проблем с восстановлением электропитания отдельных сенсорных узлов, важнейшей составляющей сенсорной сети является ее энергосистема. Возможность минимизации расхода энергии узлами в течение жизненного цикла сети рассматривается как одна из приоритетных задач при создании всепроникающих сенсорных сетей [4]. Все вышесказанное приводит к наличию целого ряда особенностей в обеспечении сетевой безопасности всепроникающих сенсорных сетей.

Анализ особенностей обеспечения сетевой безопасности во всепроникающих сенсорных сетях начнем с рекомендации МСЭ-Т X.1311 «Структура безопасности для всепроникающих сенсорных сетей» [5].

В рекомендации X.1311 «Структура безопасности для всепроникающих сенсорных сетей» выделяются пять категорий взаимосвязей в USN, которые могут представлять интерес с точки зрения сетевой безопасности:

– взаимодействие сенсорного узла с базовой станцией (шлюзом);

– взаимодействие базовой станции (шлюза) с сенсорным узлом;

– взаимодействие базовой станции со всеми сенсорными узлами, например, при перепрограммировании сенсорной сети;

– взаимодействие между узлами сенсорной сети, включая взаимодействие головного узла кластера с сенсорными узлами и взаимодействие близлежащих сенсорных узлов между собой;

– взаимодействие между базовой станцией (шлюзом) и определенной группой сенсорных узлов, объединенных, например, общим местоположением.

Приведенный перечень взаимосвязей достаточно полно характеризует USN, за исключением такого специфичного взаимодействия как взаимодействие головного узла кластера с головным узлом другого кластера.

Далее в рекомендации X.1311 отмечается, что особенности построения и функционирования всепроникающих сенсорных сетей предопределяют проблемы с использованием традиционных способов и средств обеспечения безопасности в сетях связи.

Действительно, сложно использовать общеупотребительную систему криптографических ключей вследствие достаточно строгих требований к сенсорным узлам по вычислительной мощности, объему памяти и энергопотреблению. Эти же требования способствуют и повышенной уязвимости сенсорных узлов из-за отсутствия экономической возможности сделать их защищенными при массовом применении. Кроме того, при создании сенсорной сети ее узлы размещаются на сенсорном поле, как правило, случайным образом, исходя из чего, могут возникнуть проблемы с определением местоположения при использовании традиционных протоколов безопасности. Достаточно сложной проблемой обеспечения безопасности в USN является также тот факт, что базовая станция (шлюз) является точкой концентрации всей информации от всех сенсорных узлов сети, что делает ее привлекательным объектом для разнообразных атак.

Для USN, естественно, как для сети, возможны все известные виды угроз (вторжений). Однако особенности построения и функционирования всепроникающих сенсорных сетей порождают как новые виды угроз (вторжений), так и модифицируют содержание существующих.

Наиболее оригинальными и новыми угрозами являются так называемые энергетические вторжения, призванные уменьшить жизненный цикл сенсорной сети за счет несанкционированного использования ограниченных ресурсов энергетической системы сенсорной сети [6]. Этому вопросу в рамках рекомендации X.1311, к сожалению, не уделено должного внимания, и он будет рассмотрен отдельно в заключительной части статьи. В рекомендации X1311 рассматриваются следующие угрозы, имеющие существенную специфику для всепроникающих сенсорных сетей:

– уязвимость отдельных сенсорных узлов;

– секретность данных сенсорных сетей и съем информации;

– отказ в обслуживании;
– злонамеренное использование сенсорных сетей.

В части уязвимости отдельных сенсорных узлов можно отметить, что каждый из них индивидуально может быть подвергнут различным атакам. Кроме того, в сеть могут быть внедрены злонамеренные узлы, по своим характеристикам не отличающиеся от легальных сенсорных узлов, но выполняющие задачи в иных целях, чем атакуемая сенсорная сеть. Такие вторжения во всепроникающие сенсорные сети называют клонированием, и выявление клонированных узлов представляет собой достаточно сложную задачу. В случае обнаружения клонированный узел должен быть изолирован от сети. Атакованные легальные сенсорные узлы могут изменить свои свойства, и для их возвращения в режим штатного функционирования может потребоваться перепрограммирование узла или перезапуск сенсорной сети в целом при достаточно большом числе поврежденных узлов.

В составе информации, собираемой сенсорной сетью, возможно присутствие информации, имеющий конфиденциальный или частный характер. Действительно, информация об энергопотреблении в квартире [7] или освещенности имеет частный характер и может свидетельствовать об активности, которую проявляют хозяева дома (квартиры) в течение определенного отрезка времени. Таких примеров можно привести достаточно много, но особенно актуальны эти примеры для системы электронного здоровья (e-health) [8]. Поэтому, проблемы секретности информации и ее съема являются чрезвычайно актуальными для всепроникающих сенсорных сетей. Наиболее приемлемым решением таких проблем представляется передача информации по разным маршрутам, но этот метод можно использовать только при широком внедрении протокола RPL (Routing Protocol for Low energy and lossy networks) [9] и ему подобных.

Хорошо известные атаки DoS (Denial of Service) – отказ в обслуживании – в сенсорных сетях проявляются, например, на физическом уровне в форме джамминга (глушение радиопередачи).

Злонамеренное использование сенсорных сетей возможно криминальными элементами для реализации нелегальных целей. Это относится и к собственно сенсорным сетям, но, в первую очередь, все-таки к сенсорно-актуаторным сетям [10], в которых возможен не только сбор данных с Интернет Вещей, но и управление ими.

Кроме рассмотренных выше атак и угроз, существенное значение имеют также атаки, направленные на маршрутизацию сообщений. Среди этих атак выделяются следующие: атаки по созданию ложной информации маршрутизации (spoof), видоизмененной информации (alter), замещению информации (replay). Эти атаки направлены на увеличение задержки при передаче информации из конца в конец сети, изменение

маршрута, перераспределение трафика в сети. Атаки по избирательной переадресации (selective forwarding) предусматривают отказ в передаче определенных сообщений, сброс сообщений, которые должны были быть направлены в определенных направлениях. Атаки типа «бездонная воронка» (sinkhole) направлены на то, чтобы обеспечить передачу всего трафика из вполне определенной зоны через клонированный или специально поврежденный узел сенсорной сети. Один или несколько злонамеренных (клонированных) узлов могут иметь множество псевдо идентификаторов, в том числе и по местоположению. В этом случае имеет место так называемая «колдовская» атака (Sybil). Атака «червоточина» (wormhole) предусматривает создание специального пути между двумя и более злонамеренными узлами сенсорной сети для передачи по ним перехваченных пакетов, доступных только для атакующей системы. Как правило, для создания такого пути на физическом уровне используется иной диапазон частот, чем в атакуемой всепроникающей сенсорной сети. Атака «переполнение» (HELLO flood attack) является широковещательной атакой, призванной направить в сенсорную сеть массу необязательных сообщений, которые должны лишить сеть разнообразных ресурсов – канальной емкости, вычислительной мощности, энергетических и т. д. Задача атаки «ложное подтверждение» (acknowledgement spoofing) – сфабриковать пакет «подтверждение», например, от погибшего сенсорного узла.

Следует отметить, что самоорганизация привносит и некоторые новые положительные свойства по устойчивости к атакам, связанные, в первую очередь, с тем, что после атак или во время атаки USN может быть реконфигурирована.

Специфика предназначения всепроникающих сенсорных сетей, заключающаяся в мониторинге n-мерного пространства, приводит к тому, что для исследования безопасности в сенсорных сетях наибольшее употребление нашел термин вторжение. При этом зачастую рассматриваются как периметрические вторжения, так и вторжения по всему пространству сенсорного поля.

В [11] рассмотрены концептуальные вопросы использования периметрической защиты для всепроникающих сенсорных сетей. Используется плоскостная модель сенсорной сети, в которой границы сенсорного поля представляют собой окружность с радиусом R. В работах по сенсорным сетям, как правило, в качестве плоскостной модели рассматривается квадрат. Однако рассмотрение вместо квадрата вписанной в квадрат со стороной $2R$ окружности практически не влияет на суть результатов. Шлюз находится в центре сети и основной задачей по защите от вторжений является обеспечение раннего предупреждения шлюза об обнаружении вторжения. На рисунке приведена традиционная модель однородной всепроникающей сенсорной сети, из которой видно, что ряд

сенсорных узлов перекрывают периметрическую границу и, естественно, эти узлы могут обеспечивать раннее предупреждение шлюза о вторжении [12].

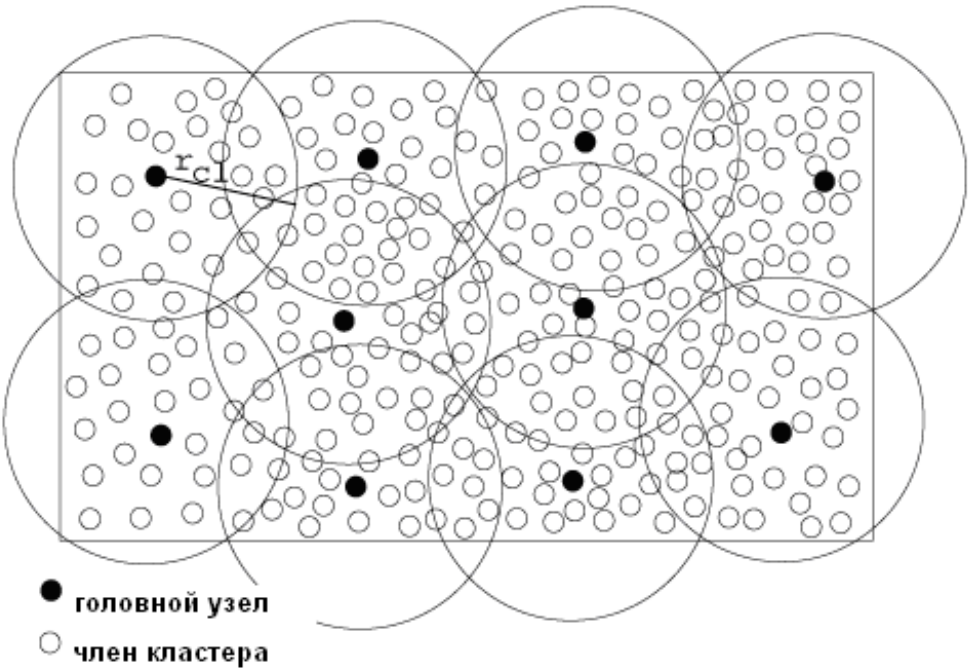

Рисунок. Модель всепроникающей сенсорной сети
с покрытием по периметру

С целью улучшения характеристик сетевой безопасности всепроникающих сенсорных сетей в [11] предлагается в границах сенсорного поля создавать несколько слоев, обеспечивающих заблаговременное предупреждение шлюза о вторжении.

В [13] предлагается новый вид вторжений во всепроникающие сенсорные сети, предусматривающий лишение сна сенсорных узлов, находящихся в момент вторжения в спящем состоянии. Это достаточно эффективное воздействие на всепроникающие сенсорные сети в [13] рассматривается как вторжение на физическом и канальном уровнях.

В [6] предложен новый вид вторжений – создание потоков ложных событий, который должен воздействовать на сеть на сетевом уровне модели взаимодействия открытых систем. Ложное событие так же, как и легальное событие требует реакции от сенсорного узла, которая в данном случае выражается в детектировании события и формировании пакета информации для передачи информации о возникновении события либо для головного узла кластера, либо непосредственно от головного узла к шлюзу. Это требует дополнительного расхода энергии и уменьшает жизненный цикл сети. Имея информацию о реальных потоках событий, технически несложно создать подобные им потоки ложных со-

бытий. Вопросы циркулирующих во всепроникающих сенсорных сетях потоках вызовов достаточно подробно изучены в работах [14, 15]. Для нашего анализа важно, что все потоки трафика для приложений, изученных в указанных выше работах, самоподобны, хотя и с различными численными значениями параметра Херста.

Таким образом, анализ особенностей обеспечения сетевой безопасности во всепроникающих сенсорных сетях показывает, что появляется достаточно много новых научных задач, связанных как с исследованиями модификаций известных процессов вторжения в сети связи, так и с формированием специфических вторжений, обусловленных природой всепроникающих сенсорных сетей. При этом важную роль играют модели трафика сети, например, для создаваемых потоков ложных событий.

Библиографический список

1. **Сети** связи пост-NGN / Б. С. Гольдштейн, А. Е. Кучерявый. – СПб. : БХВ, 2012.

2. **Самоорганизующиеся** сети / А. Е. Кучерявый, А. В. Прокопьев, Е. А. Кучерявый. – СПб. : «Любавич», 2011.

3. **Сети** связи общего пользования. Тенденции развития и методы расчета / А. Е. Кучерявый, А. И. Парамонов, Е. А. Кучерявый. – М. : ФГУП ЦНИИС, 2008.

4. **Prediction-based** Clustering Algorithm for Mobile Wireless Sensor Networks / A. Koucheryavy, A. Salim // Proceedings, International Conference on Advanced Communication Technology, 2010. ICACT 2010. Phoenix Park, Korea.

5. **Recommendation X.1311** «Security Framework for Ubiquitous Sensor Networks». ITU-T. – Geneva, February, 2011.

6. **Характеристики** жизненного цикла мобильной сенсорной сети при различных потоках ложных событий / И. А. Богданов, А. И. Парамонов, А. Е. Кучерявый // Электросвязь. – 2013. – № 1.

7. **On M2M** Communications Standards for Smart Metering / M. Sneps-Sneppe, A. Maximenko, D. Namiot // INTHITEN (Internet of Things and its Enablers) Conference. Proceedings. SPb SUT, 3–5 June, 2013.

8. **The QoS** Estimation for Physiological Monitoring Service in the M2M Network / A. Koucheryavy, Y. Al-Naggar // INTHITEN (Internet of Things and its Enablers) Conference. Proceedings. SPb SUT, 3–5 June, 2013.

9. **Модели** трафика для приложений передачи изображений во всепроникающих сенсорных сетях / А. Мутханна, А. В. Прокопьев // Электросвязь. – 2013. – № 1.

10. **Adaptive** Learning and Reconfiguration in Wireless Sensor Actuator Networks for Control Applications / D. N. Nkwogu, A. R. Allen. // The 12th Annual Post Graduate symposium on the Convergence of Telecommunica-

tions, Networking and Broadcasting. Proceedings. Liverpool, Great Britain, 27–28 June, 2011.

11. **Protecting** with Sensor Networks: Attention and response / J. V. Nickerson, S. Olariu // HICSS, 2007. 40th Hawaii International Conference jn System Science. 3–6 January, 2007. Waikoloa, Big Island, HI, USA.

12. **Выбор** головного узла кластера в однородной беспроводной сенсорной сети / А. Е. Кучерявый, А. Салим / Электросвязь. – 2009. – № 8.

13. **Sleep** Deprivation Attack Detection in Wireless Sensor Networks / T. Bhattassali, R. Chaki, S. Sanyal. // International Journal of Computer Applications, v. 40, № 15, February 2012.

14. **Ubiquitous** Sensor Networks Traffic Models for Telemetry Applications / A. Koucheryavy, A. Prokopiev // Smart Spaces and Next Generation Wired/Wireless Networking. 11th International Conference, NEW2AN 2011, and 4th Conference on Smart Spaces, ruSMART 2011. St. Petersburg, Russia, August 2011, Proceedings. LNCS 6869. Springer, 2011.

15. **Ubiquitous** Sensor Networks Traffic Models for Medical and Tracking Applications / A. Vybornova, A. Koucheryavy // Smart Spaces and Next Generation Wired/Wireless Networking. 12th International Conference, NEW2AN 2012, and 5th Conference on Smart Spaces, ruSMART 2012. St. Petersburg, Russia, August 2012, Proceedings. LNCS 7469. Springer, 2011.

Аннотация

Всепроникающие сенсорные сети USN (Ubiquitous Sensor Networks) являются технологической основой концепции Интернета Вещей и представляют собой один из наиболее динамично развивающихся сегодня сегментов рынка телекоммуникаций. Стремительное развитие USN поставило ряд задач, которые естественным образом возникают при масштабном внедрении новых технологий. Это, в первую очередь, задачи по обеспечению гарантированного уровня качества обслуживания и обеспечению информационной безопасности. В рамках задач информационной безопасности одно из ключевых мест для новых технологий телекоммуникаций занимают задачи обеспечения сетевой безопасности, рассматриваемой Международным Союзом Электросвязи между двумя интерфейсами пользователь-сеть UNI (User Network Interface). В статье анализируются особенности обеспечения сетевой безопасности во всепроникающих сенсорных сетях.

I. Bogdanov, A. Koucheryavy

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

THE NETWORK SECURITY FEATURES FOR UBIQUITOUS SENSOR NETWORKS ANALYSIS

Annotation

The Ubiquitous Sensor Networks (USN) is the technology base for the Internet of Things concept implementation. The USN network security features are in the focus of this paper. The USN is the self-organizing network with multitude of sensor nodes. It determines the network security features for USN. The energy system is the most vulnerable USN enabler. The modified and new attacks to USN are analyzed in the paper.

Key words: ubiquitous sensor networks, network security, spurious flows.

References

1. **Seti** svjazi post-NGN / B. S. Gol'dshtejn, A. E. Kucherjavyj. – SPb. : BHV, 2012.

2. **Samoorganizujushhiesja** seti / A. E. Kucherjavyj, A. V. Prokop'ev, E. A. Kucherjavyj. – SPb. : «Ljubavich», 2011.

3. **Seti** svjazi obshhego pol'zovanija. Tendencii razvitija i metody rascheta / A. E. Kucherjavyj, A. I. Paramonov, E. A. Kucherjavyj. – M. : FGUP CNIIS, 2008.

4. **Prediction-based** Clustering Algorithm for Mobile Wireless Sensor Networks / A. Koucheryavy, A. Salim // Proceedings, International Conference on Advanced Communication Technology, 2010. ICACT 2010. Phoenix Park, Korea.

5. **Recommendation X.1311** «Security Framework for Ubiquitous Sensor Networks». ITU-T. – Geneva, February, 2011.

6. **Harakteristiki** zhiznennogo cikla mobil'noj sensornoj seti pri razlichnyh potokah lozhnyh sobytij / I. A. Bogdanov, A. I. Paramonov, A. E. Kucherjavyj // Jelektrosvjaz'. – 2013. – № 1.

7. **On M2M** Communications Standards for Smart Metering / M. Sneps-Sneppe, A. Maximenko, D. Namiot // INTHITEN (Internet of Things and its Enablers) Conference. Proceedings. SPb SUT, 3–5 June, 2013.

8. **The QoS** Estimation for Physiological Monitoring Service in the M2M Network / A. Koucheryavy, Y. Al-Naggar // INTHITEN (Internet of Things and its Enablers) Conference. Proceedings. SPb SUT, 3–5 June, 2013.

9. **Modeli** trafika dlja prilozhenij peredachi izobrazhenij vo vse-pronikajushhih sensornyh setjah / A. Muthanna, A. V. Prokop'ev // Jelektrosvjaz'. – 2013. – № 1.

10. **Adaptive** Learning and Reconfiguration in Wireless Sensor Actuator Networks for Control Applications / D. N. Nkwogu, A. R. Allen. // The 12th Annual Post Graduate symposium on the Convergence of Telecommunications, Networking and Broadcasting. Proceedings. Liverpool, Great Britain, 27–28 June, 2011.

11. **Protecting** with Sensor Networks: Attention and response / J. V. Nickerson, S. Olariu // HICSS, 2007. 40th Hawaii International Conference jn System Science. 3–6 January, 2007. Waikoloa, Big Island, HI, USA.

12. **Vybor** golovnogo uzla klastera v odnorodnoj besprovodnoj sensornoj seti / A. E. Kucherjavyj, A. Salim / Jelektrosvjaz'. – 2009. – № 8.

13. **Sleep** Deprivation Attack Detection in Wireless Sensor Networks / T. Bhattassali, R. Chaki, S. Sanyal. // International Journal of Computer Applications, v. 40, № 15, February 2012.

14. **Ubiquitous** Sensor Networks Traffic Models for Telemetry Applications / A. Koucheryavy, A. Prokopiev // Smart Spaces and Next Generation Wired/Wireless Networking. 11th International Conference, NEW2AN 2011, and 4th Conference on Smart Spaces, ruSMART 2011. St. Petersburg, Russia, August 2011, Proceedings. LNCS 6869. Springer, 2011.

15. **Ubiquitous** Sensor Networks Traffic Models for Medical and Tracking Applications / A. Vybornova, A. Koucheryavy // Smart Spaces and Next Generation Wired/Wireless Networking. 12th International Conference, NEW2AN 2012, and 5th Conference on Smart Spaces, ruSMART 2012. St. Petersburg, Russia, August 2012, Proceedings. LNCS 7469. Springer, 2011.

Богданов Игорь Александрович – аспирант кафедры «Сети связи» ФГОБУ ВПО «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

Кучерявый Андрей Евгеньевич – доктор технических наук, профессор, заведующий кафедрой «Сети связи» ФГОБУ ВПО «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», akouch@mail.ru

УДК 621.395

А. А. Дорт-Гольц, О. А. Симонина

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича

БАЗОВЫЕ МОДЕЛИ СЕТЕВЫХ УСТРОЙСТВ АГРЕГАЦИИ ТРАФИКА

самоподобие, долговременные зависимости, агрегирование трафика

Введение

Вопрос самоподобия трафика различных приложений и на различных уровнях активно исследуется последние десятилетия [1, 2]. Исследования показывают, что трафик обладает пачечностью и долговременными зависимостями по нескольким причинам [3, 4]. Если рассматривать возникновение самоподобия во временной области, то чаще всего это поведение пользователя, а также влияние приложения, т. е. обусловленные способом возникновения данных. Таким образом, самоподобный характер трафика возникает на прикладном уровне и на оконечных устройствах. Отчасти характеристики такого трафика можно предсказать на основе статистических наблюдений и алгоритмов формирования потока. Например, трафик IPTV имеет предсказуемые характеристики потока, если заранее известен видеоряд.

Но существует еще и пространственная зависимость от количества пользователей, географического расположения сетевых узлов, плотности населения. При этом появляется дополнительный фактор – слияние различных потоков, обладающих временным самоподобием, в единый поток. Такой трафик, называемый агрегированным, предположительно обладает характеристиками, которые не лучше характеристик самого пачечного потока, т. е. коэффициент Хёрста

$$H_{agr} \geq H_{max}, \qquad (1)$$

где H_{agr} – коэффициент Хёрста агрегированного потока, H_{max} – наибольший коэффициент Хёрста агрегируемых потоков [5].

Однако при передаче по сети трафик подвергается ряду преобразований и воздействий. Экспериментальные данные показывают, что агрегированный трафик чаще всего не соответствует данному условию (1),

а значения коэффициента Хёрста могут варьироваться в широком диапазоне. При передаче по сети на трафик влияют сетевые устройства, политика обработки в этих сетевых устройствах, механизмы обеспечения качества обслуживания, перегрузки на узлах, работа протоколов транспортного уровня, динамическая маршрутизация, балансировка нагрузки и прочее. Таким образом, характеристики потока могут быть существенно изменены влиянием сети.

Отдельно отметим наложенные сети, которые реализуют функции сетевых устройств на более высоких уровнях. В этом случае добавляется влияние технологий нижних уровней на сформированный поток.

На начальном этапе попробуем учесть влияние сетевого устройства и его политики обслуживания на агрегируемый поток. Для этого предлагается выделить две базовые модели, которые могут быть реализованы на разных уровнях: как на канальном и сетевом, так и на более высоких, в случае наложенных сетей. Для первой базовой модели характерно наличие корреляции между длинами пакетов и временами обслуживания. Примером такой модели может являться коммутационное устройство канального уровня, например, коммутатор Ethernet. Во второй модели время обслуживания не зависит от длины пакета и, более того, может быть постоянным. Подобная модель реализована в маршрутизаторах IP и MPLS. При этом основной функцией сетевых устройств является агрегирование и перераспределение трафика.

Базовые модели сетевых устройств

Обозначим *i*-й самоподобный случайный процесс как $X_i(t)$. В случае рассмотрения процесса появления интервалов между пачками обозначение будет $X_{IAT_i}(t)$, а для процесса, характеризующего длины пачек – $X_{PL_i}(t)$. Предположим, что все исходные рассматриваемые случайные процессы взаимно независимы и стационарны. Результатом агрегации случайных процессов $X_i(t)$ будет случайный процесс $Y(t)$. Основными характеристиками данных процессов являются математическое ожидание m и автокорреляционная функция $C(t_1,t_2)$ t_2). Т. к. процессы независимы и стационарны, то:

$$m_{xi} = E\left[X_i(T)\right];$$

$$R_i(\tau) = E\left[X_i(0)X_i(\tau)\right];$$

$$C_i(\tau) = E\left[(X_i(0)-m_{xi})(X_i(\tau)-m_{xi})\right] = E\left[X_i(0)X_i(\tau)\right] - m_{xi}^2,$$

где m_{xi}, $R_i(\tau)$ и $C_i(\tau)$, соответственно математическое ожидание, автоковариационная и автокорреляционная функции *i*-го процесса. Как из-

вестно, в случае случайного процесса, стационарного в широком смысле и обладающего долговременной зависимостью, убывание его автоковариационной функции характеризуется следующим соотношением:

$$R(\tau) \sim \tau^{2H-2}$$

где H – параметр Хёрста данного случайного процесса.

На рисунке 1 представлена обобщенная схема объединения различных потоков трафика на сетевом устройстве.

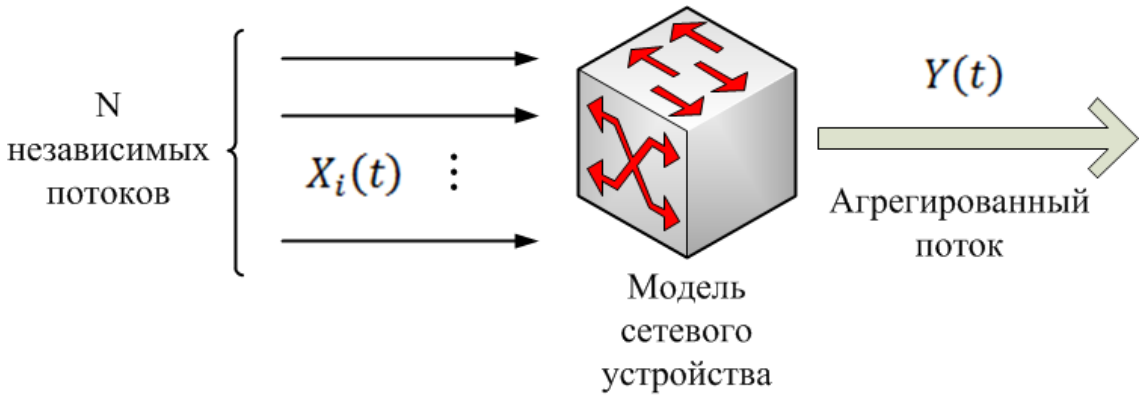

Рис. 1. Схема агрегации потоков на сетевом устройстве

В данной статье множество сетевых устройств ограничено двумя базовыми моделями, назовем их «маршрутизатор» в случае фиксированного времени обслуживания, и «коммутатор» в случае влияния длин пакетов на время обслуживания. Для каждой модели рассмотрим процессы агрегации в отдельности.

Агрегация потоков на «коммутаторе»

В модели сетевого устройства «коммутатор» (рис. 2) процесс объединения потоков трафика условно разделен на две фазы: обработка трафика и слияние потоков. Под обработкой понимаются действия, совершаемые сетевым устройством над пакетами, например, проверка контрольной суммы принятого кадра, поиск адреса в таблице коммутации и т. п.

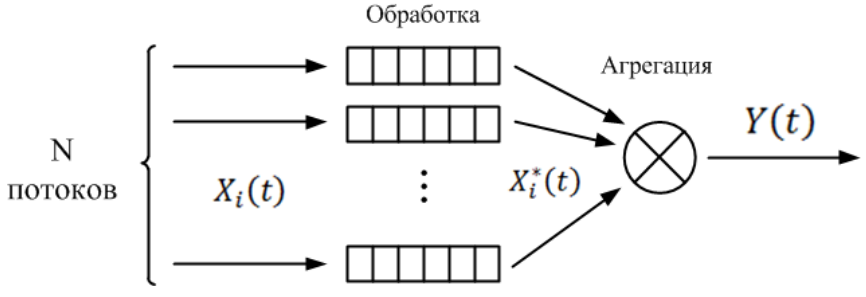

Рис. 2. Модель агрегации трафика на сетевом устройстве

Таким образом, характеристики исходных потоков, представленных случайными процессами $X_i(t)$, в процессе обработки будут изменяться. Обозначим полученные процессы как $X_i^*(t)$. В общем случае будем считать время обработки каждого кадра пропорциональным длине данного кадра, а длины кадров не изменяющимися в процессе обработки. Тогда:

$$\begin{cases} X^*_{IAT_i}(t) = X_{IAT_i}(t) + kX_{PL_i}(t); \\ X^*_{PL_i}(t) = X_{PL_i}(t). \end{cases} \quad (2)$$

Заметим, что касательно интервалов между пачками промежуточный процесс сам по себе является результатом объединения двух независимых случайных процессов $X_{IAT_i}(t)$ и $X_{PL_i}(t)$ со следующими характеристиками:

$$m^*_{IATx_i} = E[X^*_{IAT}(t)] = E[X_{IAT_i}(t)] + kE[X_{PL_i}(t)], \quad (3)$$

$$\begin{aligned} R^*_{IATx_i}(\tau) &= E\left[\left(X_{IAT_i}(0) + kX_{PL_i}(0)\right)\left(X_{IAT_i}(\tau) + kX_{PL_i}(\tau)\right)\right] = \\ &= E\left[X_{IAT_i}(0)X_{IAT_i}(\tau)\right] + kE\left[X_{IAT_i}(\tau)X_{PL_i}(0)\right] + \\ &+ kE\left[X_{IAT_i}(0)X_{PL_i}(\tau)\right] + k^2E[X_{PL_i}(0)X_{PL_i}(\tau)]. \end{aligned} \quad (4)$$

Поскольку процессы независимы, их взаимные корреляционные функции равны нулю.

$$m^*_{IATx_i} = m_{IATx_i} + km_{PLx_i};$$

$$R^*_{IATx_i}(\tau) = R_{IATx_i}(\tau) + k^2R_{PL_i};$$

$$m^*_{PLx_i} = m_{PLx_i};$$

$$R^*_{PLx_i}(\tau) = R_{PLx_i}(\tau).$$

После предварительной обработки потоки сливаются в один:

$$Y(t) = \sum\nolimits_{i=1}^{N} X_i^*(t), \quad (5)$$

где N – количество объединяемых потоков.

В случае слияния потоков интервалов (далее *IAT-процессов*), необходимо воспользоваться дополнительными преобразованиями:

$$\lambda_i(t) = 1 / X^*_{IAT_i}(t),$$

$$\lambda_y(t) = \sum\nolimits_{i=1}^{N} \lambda_i(t),$$

$$Y_{IAT}(t) = 1/\lambda_y(t).$$

Соответственно, зная характеристики процессов $X^*_{IAT_i}(t)$ и воспользовавшись данным промежуточным преобразованием, можно определить характеристики результирующего процесса. Проблема, однако, состоит в том, что оператор $\frac{1}{x}$ является нелинейным. Как известно, если к исходному случайному процессу было применено нелинейное преобразование вида $z = \varphi[x(t)]$, то математическое ожидание и автокорреляционную функцию полученного процесса, при условии его стационарности, можно найти следующим образом:

$$m_z(t) = \overline{\varphi[x(t)]} = \int_{-\infty}^{\infty} \varphi(x) w(x,t) dx;$$

$$C_z(\tau) = \iint_{-\infty}^{\infty} [\varphi(x(0)) - m_z(t)][\varphi(x(\tau)) - m_z(t)] w(x(0), x(\tau), \tau)\, dx\, dx,$$

где *w(x,t)* – плотность распределения вероятности процесса *x(t)*.

При анализе длин пачек (далее *PL-процесс*) характеристики объединенного процесса могут быть найдены как

$$Y_{PL}(t) = \sum_{i=1}^{N} X_{PLi}(t);$$

$$m_{PLy} = \sum_{i=1}^{N} m_{PLi};$$

$$R_{PLy}(\tau) = E[Y_{PL}(0) Y_{PL}(\tau)] = E\left[\sum_{i=1}^{N} X_{PLi}(0) \sum_{i=1}^{N} X_{PLi}(\tau)\right] = \sum_{i=1}^{N} R_{PLi}. \quad (6)$$

Агрегация потоков на «маршрутизаторе»

Вторая модель может рассматриваться как частный случай первой. Отличие будет заключаться в промежуточных преобразованиях процессов при предварительной обработке. Упрощенно можно считать, что время обработки пакетов в маршрутизаторах с использованием дисциплины FIFO будет постоянной величиной, а длины пакетов остаются неизменными. Тогда промежуточные процессы будут выглядеть следующим образом:

$$\begin{cases} X^*_{IAT_i}(t) = X_{IAT_i}(t) + h; \\ X^*_{PL_i}(t) = X_{PL_i}(t), \end{cases}$$

где h – время обработки пакета в маршрутизаторе.

Найдем характеристики промежуточных процессов:

$$m^*_{IATx_i} = E[X_{IAT_i}(t) + h];$$

$$R^*_{IATx_i}(\tau) = E\left[\left(X_{IAT_i}(0) + h\right)\left(X_{IAT_i}(\tau) + h\right)\right] =$$

$$= E\left[X_{IAT_i}(0)X_{IAT_i}(\tau) + hX_{IAT_i}(0) + hX_{IAT_i}(\tau) + h^2\right].$$

Откуда с учетом неизменности длин пакетов:

$$m^*_{IATx_i} = m_{IATx_i} + h;$$

$$R^*_{IATx_i}(\tau) = R_{IATx_i}(\tau) + h(2m_{IATx_i} + h);$$

$$m^*_{PLx_i} = m_{PLx_i};$$

$$R^*_{PLx_i}(\tau) = R_{PLx_i}(\tau).$$

Дальнейшая агрегация процессов производится методом, описанным для модели «коммутатор». Для агрегированного PL-процесса основные характеристики будут выглядеть следующим образом:

$$Y_{PL}(t) = \sum_{i=1}^{N} X_{PLi}(t);$$

$$m_{PLy} = \sum_{i=1}^{N} m_{PLi};$$

$$R_{PLy}(\tau) = E\left[Y_{PL}(0)Y_{PL}(\tau)\right] = E\left[\sum_{i=1}^{N} X_{PLi}(0)\sum_{i=1}^{N} X_{PLi}(\tau)\right] = \sum_{i=1}^{N} R_{PLi}.$$

Заключение

Предложенные в статье базовые модели позволяют оценить изменения параметров трафика, вызванные прохождением множества потоков через уровни доступа и агрегации и их неизбежным слиянием на сетевых устройствах. Отметим, что данный подход позволяет совместить два традиционных подхода: первый – изменение трафика во временной области и нахождение признаков самоподобия, проявляющихся в виде пачечности; второй – географический, связанный с проектированием сетей, неравномерностью распределения пользователей по регионам и расположения информационных ресурсов.

В дальнейшем особый интерес представляет оценка агрегации трафика наложенной сети, обусловленная несовпадением логических (на уровне наложенной сети) и физических маршрутов, в том числе на магистральных участках сетей.

Библиографический список

1. **On the self-similar** nature of Ethernet traffic / W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson // ACM SIGCOMM Computer Communication Review, vol. 23, no. 4, Oct. 1993. – PP. 183–193.
2. **Statistical** analysis and modeling of peer-to-peer multimedia traffic / N. M. Markovich, U. R. Krieger // Network Performance Engineering, Lecture Notes in Computer Science Volume 5233, 2011. – PP. 70–97.
3. **Explaining** world wide web traffic self-similarity / M. Crovella, A. Bestavros. – Computer Science Department , Boston University, Technical Report TR-95-015, October 12, 1995.
4. **On the aggregation** of self-similar processes / G. Mazzini, R. Rovatti, G. Setti // IEICE Trans. Fundamentals, Vol. E88-A, no. 10, 2005. – PP. 2656–2663.
5. **Фрактальные** процессы в телекоммуникациях / О. И. Шелухин, А. М. Тенякшев, А. В. Осин. – М. : Радиотехника, 2003. – 480 с.

Аннотация

В статье вводятся две базовые модели сетевых устройств. В первой модели длины пакетов влияют на время обслуживания, и, таким образом, поведение трафика на выходе такого устройства является результатом взаимодействия двух процессов. Вторая модель предполагает обслуживание независимо от длин пакетов. Анализ моделей сделан с учетом самоподобия трафика.

A. Dort-Golts, O. Simonina

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

BASIC MODELS OF TRAFFIC AGGREGATING NETWORK NODES

Annotation

There are two basic network nodes models introduced in the article. In the first model packet lengths influence the service time and traffic behavior after leaving such node results from two stochastic processes interaction. Second model assumes packet length independent serving. Analysis of the models is made taking into account traffic self-similarity.

Key words: self-similarity, LRD, traffic aggregation.

References

1. **On the self-similar** nature of Ethernet traffic / W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson // ACM SIGCOMM Computer Communication Review, vol. 23, no. 4, Oct. 1993. – PP. 183–193.

2. **Statistical** analysis and modeling of peer-to-peer multimedia traffic / N. M. Markovich, U. R. Krieger // Network Performance Engineering, Lecture Notes in Computer Science Volume 5233, 2011. – PP. 70–97.

3. **Explaining** world wide web traffic self-similarity / M. Crovella, A. Bestavros // Computer Science Department , Boston University, Technical Report TR-95-015, October 12, 1995.

4. **On the aggregation** of self-similar processes / G. Mazzini, R. Rovatti, G. Setti // IEICE Trans. Fundamentals, Vol. E88-A, no. 10, 2005. – PP. 2656–2663.

5. **Fraktal'nye** processy v telekommunikacijah / O. I. Sheluhin, A. M. Tenjakshev, A. V. Osin. – M. : Radiotehnika, 2003. – 480 s.

Дорт-Гольц Антон Александрович – аспирант ФГОБУ ВПО «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», dortgolts@gmail.com

Симонина Ольга Александровна – кандидат технических наук, доцент кафедры «Сети связи» ФГОБУ ВПО «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», simonina@bk.ru

УДК 004.43

К. Е. Израилов

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича

РАСШИРЕНИЕ ЯЗЫКА «*C*» ДЛЯ ОПИСАНИЯ АЛГОРИТМОВ КОДА ТЕЛЕКОММУНИКАЦИОННЫХ УСТРОЙСТВ

современное общество, информационное пространство, телекоммуникационные устройства, язык С, расширение языка, эффективность кода, восстановление кода, язык описание алгоритмов, безопасность, уязвимость

Введение

Современное общество, присущее большинству высокоразвитых стран, без преувеличения можно считать потребительским. Оставляя обсуждение аргументов в защиту и против такого общества, рассмотрим следующие его черты, имеющие отношение к информационным технологиям [1] и их безопасности.

Во-первых, массовое потребление продуктов в сфере вычислительной техники (с целью самовыделения потребителя без объективной необходимости) привело к развитию компьютерных технологий, и, как следствие, к стремительному росту производительности платформ последних. Это, в свою очередь, сместило баланс от написания эффективного исходного кода программного обеспечения (далее – ПО) в сторону удобство и скорости процесса его разработки. Например, такая тенденция поспособствовала появлению языков, выполняемых интерпретаторами (*Perl*, *Ruby*) и на виртуальных машинах (*C#*, *Java*).

Во-вторых, революция в сфере коммуникаций привела к популяризации информационного пространства (далее – ИП), используемого для общения социумов. Популяризация же явилась одной из причин роста количество индивидуумов, вовлеченный в процесс общения, что автоматически дало серьезную нагрузку на телекоммуникационные устройства (далее – ТКУ), обслуживающие ИП. Для поддержания возросшей нагрузки простого усиления мощностей оборудования уже не достаточно, и задача высокой производительности его ПО (в противовес распространенному мнению) становится крайне актуальной.

За популяризацией также последовала доля критичных для компрометации данных, проходящих через ИП (например, передаче промышленных секретов или документов под грифом секретности через открытые сети, хотя и защищенным способом). Это актуализировало

требования к ПО, ужесточающие безопасность механизмов работы с данными.

И в-третьих, развитие различных направлений в области программирования (например, парадигма объектно-ориентированного программирования, оптимизация компилируемого кода) до сих пор не нашло достаточного применения в сфере ТКУ – многие промышленные устройства имеют ПО на процедурном языке *C*, построенное без оптимизации. Это, скорее всего, связано с настолько высокой значимостью правильной работы алгоритмов кода, что из-за этого производители готовы отказаться от использования сложных компиляторов и оптимизаций, не лишенных ошибок в генерируемом коде.

Резюмируя вышесказанное, сложилась ситуация, когда используемое ИП требует обслуживание с помощью ПО, обладающего, как высокой эффективностью (с точки зрения производительности), так и возможностями по защите данных. При этом создание новых языков пошло по иному пути развития, а именно в сторону эффективной (с точки зрения скорости) разработки ПО. Таким образом, разработка специализированного языка программирования, адаптированного для выполнения указанных общих требований к ПО, позволила бы снизить риски информационной безопасности, возникающие от угроз в ИП, связанных с экспансией информационных технологий.

Специализированный язык программирования

В качестве основы для предлагаемого специализированного языка выберем являющийся достаточно популярным и априори считающийся эффективным – язык *C*. А выполнение общих требований (увеличение производительность получаемого кода и дополнительные возможности защиты данных) реализуем с помощью расширения его синтаксиса (и, как следствие, семантики) – подобный процесс часто называется «наклонением языка». При этом нововведения должны хорошо «ложиться» на типовые процессоры, быть более удобны для применения программистами и не переводить язык в разряд более низкоуровневых; в ином случае суммарный эффект от использования нового языка может оказаться хуже первоначального.

С точки зрения разработки компилятора, новый язык не будет заметно сложнее *C*, а значит и качество генерируемого кода с точки зрения ошибок не изменится.

Преследуя цели лаконичности (сокращение до букв) и традиционности в названиях предшествующих языков (линейка языков *C*, *C++*, *C++11*, *C#*), дадим имя новому как *CPL*. Оно расшифровывается на английском языке как «*C with Performance and Security*», что переводится как «язык *C*, дополненный возможностями производительности и безопасности». Далее в статье будем использовать именно это название.

Предпосылками к определению требований можно считать такие особенности инструкций типовых процессоров, как битовый доступ к данным, многообразие выполнения итераций, особенности реального размещения переменных на регистрах процессора, менее абстрактная работа с памятью и другие. Также язык должен иметь конструкции, описывающие обеспечение безопасности данных.

Ну а поскольку *CPL* служит для написания программ обработки данных ИП, то и способ описания самих алгоритмов на нем должен быть удобным для применения человеком. В качестве противопоставления приведем область разработки сайтов, где основной задачей языков, как правило, считается создание эргономичного *Web*-интерфейса для конечного потребителя.

На основании предпосылок составим уточненный список требований (каждое из которых реализуется с помощью соответствующего языкового расширения) к *CPL*, включая их идентификаторы (*ID*).

1) Битовый доступ – возможность доступа к отдельным битам переменной.

2) Проверка значения бита – неявное введение типа *bool* (появившееся в языке *C++* и хранящее значение *true* или *false*) для проверки статуса бита.

3) Байтовый доступ – возможность доступа к отдельным байтам переменной.

4) Битово-диапазонный доступ – возможность доступа к диапазонам бит переменной.

5) Универсализация циклов – использование более однотипных по сравнению с языком *C* конструкций описания циклов, а именно:

   a) бесконечный цикл – определение бесконечного цикла с возможностью выхода только по командам *return*, *break* или *goto*;

   b) условие входа в цикл – условие выполнения итерации цикла, которое проверяется перед ее началом (соответствует условию входа в цикл);

   c) условие повторения цикла – условие повторения итерации цикла, которое проверяется после ее окончания (соответствует условию выхода из цикла);

   d) повторение цикла – команда повторения текущей итерации без изменения значения итерационной переменной.

6) Многокаскадный выход из цикла – выход из вложенного цикла на несколько уровней наверх.

7) Выходные параметры – возвращение нескольких значений функции (имеет смысл, когда использование указателя на структуру вместо набора регистров в качестве значения возврата неоправданно).

8) Директивы оптимизации – директивы, используемые пользователем для улучшения эффективности работы оптимизации (задается поль-

зователем на свой страх и риск, поскольку строго используется компилятором), а именно:

a) диапазон значений переменной – принудительная установка возможных значений переменной;
b) диапазон возвращаемых значений функции – принудительная установка возможных возвращаемых значений функции.

9) Адресное размещение объектов – задание точного адреса размещения переменной в памяти.

10) Безопасность объектов – указание объектов программы, содержащих или обрабатывающих критически важную информацию. Состоит из следующих типов:

a) безопасность переменных и памяти – указание переменных и областей памяти с данными, необходимых для защиты от доступа злоумышленника (например, путем хранения в защищенных областях памяти, по случайным адресам и «затирании» по окончанию использования);
b) безопасность функций – указание функций, реализующие механизмы работы с данными, алгоритмы которых необходимо скрыть от доступа злоумышленника (например, путем обфускации).

11) Инвариантность объектов – указание объектов, изменение которых недопустимо (как злоумышленником, так и в результате внутренних ошибок кода). Реализация требования может быть вынесена в специальный сервис проверки целостности объектов в случае разработки операционной системы. Состоит из следующих типов:

a) инвариантность переменных и памяти – указание переменных и областей памяти с данными, хранящих неизменную с точки зрения механизмов информацию (например, закрытые ключи шифрования);
b) инвариантность функций – указание функций, реализующих критически-важные с точки зрения выполнения механизмы (например, алгоритмы криптографического модуля).

Пункты с идентификаторами 9–11 удовлетворяют общему требованию по увеличению производительности получаемого кода, остальные – требованию безопасности данных.

Следует отметить, что большинство указанных расширений языка можно реализовать и с помощью классического языка *C*, а остальная часть – путем применения специализированных библиотек и оптимизаций компилятора. Тем не менее, это не является удовлетворительным решением: во-первых, такой исходный код будет менее удобным для написания и восприятия, а во-вторых, работа библиотек и компиля-

тора является достаточно непредсказуемой и не гарантирующей ожидаемый эффект.

Реализация новых возможностей *CPL* может ложиться, как на специфику самого процессора, так и на используемые программой библиотеки. Для разъяснения этого момента (являющегося достаточно важной особенностью языка), приведем следующие примеры.

Пример 1. Чтение байта памяти по адресу (*0xA000*) с занесением в переменную *b*.

```
addr = 0xA000;
b = (*addr).[0];
```

Данный пример может быть реализован с помощью генерации соответствующего ассемблерного кода для процессора, поддерживающего побайтный доступ к памяти. Так, в случае процессора *PowerPC* ассемблер будет иметь вид

```
lbz b, 0(addr)
```

Пример 2. Вычисление длины строки, хранящейся в защищаемой переменной *str*.

```
int strlen(str) secure;
char * secure str = “password”;
len = strlen(str);
```

Вычисление может быть реализовано с помощью библиотеки, функции которой в случае переменной с пометкой *secure* имеют специальные алгоритмы, не хранящие данные в легкодоступном виде. Так, для данного примера функция *strlen()* в процессе обработки может хранить части строки *str* в случайных адресах.

В примере пометка *secure* влияет на работу с переменной *str* на всем периоде ее жизни: блок памяти переменной должен шифроваться, а по окончанию использования (может определяться компилятором, средой выполнения или пользовательской директивой) быть принудительно «затерт».

Практика программирования

Для обоснования применимости нового языка, сравним его возможности с реализацией каждого из уточненных требований на классическом языке *C*, используя введенные ранее идентификаторы *ID*.

Сравнение проведем по следующим показателям: Л – исходный код на *CPL* иметь более лаконичный вид; Э – генерируемый на *CPL* код будет более производительным; Б – генерируемый на *CPL* код будет обладать элементами защиты данных.

Таблица сравнения с упрощенными примерами, которую также можно использовать для введения в синтаксис возможностей нового языка, приведена ниже.

ТАБЛИЦА. Сравнение возможностей языков *CPL* и *C*

ID	Реализация на языке *C*	Предлагаемая реализация на языке *CPL*	Результат сравнения
1)	получение значения 3-го бита		Л, Э
	(x & ~((1<<2))) >> 2;	*(x.3);*	
	установка 3-го бита		
	x |=(1<<2);	*(x.3)!true;*	
2)	*if((x & ~(0x4)))*	*if((x.3)?true)*	Л, Э
3)	получение значения 3-го байта		Л, Э, Б
	(x & ~(0xFF<<2)) >> 2;	*x.[3];*	
	установка значения 3-го байта в *0xAB*		
	x |= (0xAB<<2);	*x.[3] = 0xAB;*	
4)	получение значения числа, состоящего из 3-го и 4-го битов		Л, Э
	(x & ~((1<<2)|(1<<3))) >> 2;	*x.[3-4];*	
	установка 2-го и 3-го битов		
	x |= ((1<<2)|(1<<3));	*x.[3-4]!true;*	
5)			
5.a)	*while(1){}* или *do{…}while(1)*	*loop{…}*	Л
5.b)	*while(cond){…}*	*loop(cond){…}*	Л
5.c)	*while{…}(cond)*	*loop{…}(cond)*	Л
5.d)	*while(cond){* *again:* *…* *goto again;* *…* *}*	*loop(cond){* *…* *repeat;* *…* *}*	Л, Э
6)	выход из цикла на 2 уровня вверх		Л, Э
	(с помощью дополнительного флага выхода *fExit*) *int fExit = 0;* *while(cond1){* *…* *while(cond2){* *…* *fExit = 1;* *break;* *…* *}* *if(fExit)* *break;* *…* *}*	(с помощью аргумента команды *break*) *loop(cond1){* *…* *loop(cond1){* *…* *break(2);* *}* *…* *}*	

ID	Реализация на языке *C*	Предлагаемая реализация на языке *CPL*	Результат сравнения
7)	возврат пары значений из функции		Л, Э
	(с помощью структуры *S_R12*) *struct S_R12{* *int r1, r2;* *} r12, r12_ret;* *r12 = funct();* *funct() { return r12_ret; }* *или* *funct(&r12);* *funct(S_R12 *r12_) {* *(*r12_) = r12_ret;* *}*	*(r1, r2) = funct();* *funct(){* *return (r1_ret, r2_ret);* *}*	
8)			
8.a)	**Директива Отсутствует**, требование может быть частично удовлетворено оптимизацией компилятора (далее – ДО)	переменная *x* может принимать значение 10 или диапазон значений от 20 до 30 *int [10, 20-30]x;*	Л, Э
8.b)	ДО	функция *funct()* может возвращать значение 10 или диапазон значений от 20 до 30 *int [10, 20-30] funct();*	Л, Э,
9)	возможно лишь с помощью настроек линкера	принудительное размещение переменной *x* по адресу 0xA000 *int x &0xA000;*	Л, Э,
10)			
10.a)	возможно лишь с применением **Специальных Архитектур**, подходов и библиотек (далее – СА)	защита переменной *x* *int secure x;*	Л, Э, Б
10.b)	СА	защита функции *funct()* *int f() secure;*	Л, Э, Б
11)			
11.a)	СА; использование ключевого слова «const» языка *C* не решает задачу, поскольку оно имеет значение лишь для синтаксиса кода (хотя иногда и влияет на распределение переменных в бинарном коде)	установка неизменности переменной *key* *int const(secure) key;*	Л, Э, Б
11.b)	СА	установка неизменности функции *encrypt()* *int encrypt () const(secure);*	Л, Э, Б

Достоинства и недостатки нового языка

С уверенностью можно утверждать, что любой язык (кроме, естественно, группы эзотерических [2], лишенных особого смысла в применении) обладает множеством достоинств и недостатков. Это вытекает из того, что изначально любой язык предназначается для одних целей, а в меру дальнейшего распространения и отсутствия аналогов начинает использоваться и для иных. Таким образом, несмотря на то, классический язык *C* для ряда задач предпочтительнее, однако для задач написания эффективных низкоуровневых алгоритмов с поддержкой механизмов безопасности данных он будет заметно уступать языку *CPL*. И хотя, ряд особенностей языка *CPL* и может быть описано на языке *C* (впрочем, как и особенности последнего реализуемы на ассемблере), тем не менее, отсутствие в этом случае удобства при разработке программ может привести к ухудшению качества кода.

Также, нововведения в язык *CPL* не уменьшают набор возможностей *C*-языка, а лишь добавляют новые. Следовательно, *CPL* можно считать именно специализированным расширением функционала *C*.

Применение к задаче восстановления алгоритмов

Хотя изначально выдвигалось требование обеспечения возможности написания кода, имеющего высокую производительность выполнения (без потери удобства программирования), *CPL* также имеет непосредственное отношение к обратной задаче – реверс-инжинирингу.

Рассмотрим *CPL* с позиции языка для описания алгоритмов кода, восстановленных из машинного представления. Данная тема, включающая в себя метод и автоматизированную утилиту восстановления машинного кода ТКУ, неоднократно поднималась в предыдущих статьях [3, 4]. При этом, упростим язык с помощью исключения из него основной информации о типах объектов, поскольку она не сильно влияет на суть алгоритмов. Такую новую модификацию языка будем именовать как *AD* («*Algorithms Description*», означающее в переводе с английского «язык описания алгоритмов»).

Язык *AD* с точки зрения уровней абстракции (где самым низким является машинный код, а самым высоким – объекты сверхуровневых языков программирования), располагается между языком *C* и ассемблером, при этом обладая более лаконичным, чем у обоих, синтаксисом. Следовательно, описание алгоритмов машинного кода на нем предпочтительнее.

Поддержка конструкций безопасности данных позволит в процессе восстановления кода выявить потенциальные уязвимости ПО ТКУ – как собственно утилитой восстановления, так и при ручном анализе. Также,

аналитик безопасности может указать утилите наиболее приоритетные для проверки данные. В этом случае утилита пометит места в коде, в которых их безопасность может быть нарушена.

Прикладное применение

Комплексное сравнение синтаксисов и возможностей языков проведем на одном из вариантов исходного кода функции *inteN2A()* (переводящей 4-х байтную запись *IP*-адреса в текстовый формат «*dd.dd.dd.dd*»). Для введения функционала безопасности примем легенду, по которой обрабатываемая функцией переменная хранит *IP*-адрес, значение которого должно защищаться от доступа злоумышленника (например, адрес интернет-шлюза по умолчанию).

Листинг 1. Функция *inteN2A()* на языке *C*

```
char *inetN2A_ClassicC(long ina){
   static char buff[4 * sizeof("dd.")];
   unsigned char *ucp = (unsigned char *)&ina;

   sprintf(buff, "%d.%d.%d.%d",
      ucp[0] & 0xff,
      ucp[1] & 0xff,
      ucp[2] & 0xff,
      ucp[3] & 0xff);
   return buff;
}
```

Листинг 2. Функция *inteN2A()* на языке *CPL*

```
char secure *inetN2A_CPL(long secure ina){
   secure static char buff[4 * sizeof("dd.")];
   secure unsigned char *ucp = (unsigned char *)&ina;

   sprintf(buff, "%d.%d.%d.%d",
      ucp.[0],
      ucp.[1],
      ucp.[2],
      ucp.[3];
   return buff;
}
```

Листинг 3. Функция *inteN2A()* на языке *AD*

```
secure *inetN2A_AD(secure ina){
   secure buff[];
   secure *ucp = &ina;

   sprintf(buff, "%d.%d.%d.%d",
      ucp.[0],
      ucp.[1],
      ucp.[2],
      ucp.[3]);
   return buff;
}
```

Как можно увидеть из листингов, код на языке *CPL* выглядит более компактно и понятнее, чем на *C*. А ключевое слово *secure* сигнализирует о том, что генерируемый код должен быть максимально защищен от компрометации значения обрабатываемого *IP*-адреса.

Листинг же на языке *AD,* хотя и нельзя считать подходящим для компиляции (из-за отсутствия типов переменных), однако его лаконичность и «заостренность» на принципах работы алгоритмов функции (без соблюдения корректности работы с данными), позволяет использовать его для описания алгоритмов кода. В Листинге 3 сгенерированное ключевое слово *secure* сигнализирует аналитику кода о потенциальном нарушении безопасности данных в теле функции.

Заключение

Предложенный язык «среднего уровня» *CPL* позволяет разрабатывать программы, предназначением которых является эффективное и безопасное выполнение кода на ТКУ. При этом резкого снижения актуальности его применения в ближайшее время не ожидается. А удобство записи конструкций языка позволяет использовать его упрощенную версию (язык *AD*) и в обратной задаче – восстановлении алгоритмов кода ТКУ с целью поиска уязвимостей.

Библиографический список

1. **Социология** потребления: основные подходы / В. В. Радаев // Социологические Исследования. – 2005. – № 6. – С. 5–18.

2. **Основные** концепции языков программирования / У. С. Роберт. – М. : «Вильямс», 2001. – 672 с.

3. **Метод** алгоритмизации машинного кода телекоммуникационных устройств / М. В. Буйневич, К. Е. Израилов // Телекоммуникации. – 2012. – № 12. – С. 2–6.

4. **Автоматизированное** средство алгоритмизации машинного кода телекоммуникационных устройств / М. В. Буйневич, К. Е. Израилов // Телекоммуникации. – 2013. – № 6. – С. 2–9.

Аннотация

Статья посвящена оценке соответствия возможностей современных языков программирования потребностям общества. В качестве логичного продолжения результата сравнения предлагается расширение существующего С-языка с целью увеличения эффективности и безопасности кода для телекоммуникационных устройств. Также, обосновывается возможность применения расширения в обратной задаче по восстановления кода, используемой для поиска уязвимостей. Приводится комплексное сравнение синтаксисов и возможностей языков на примере кода реальной функции.

K. Izrailov

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

C-LANGUAGE EXTENSION FOR ALGORITHM DESCRIPTION OF TELECOMMUNICATION DEVICES CODE

Annotation

The article is devoted to the assessment of compliance with the capabilities of modern programming languages needs of society. As a logical continuation of the comparison proposed expansion of the existing C-language in order to increase efficiency and security code for telecommunication devices. Also, the possibility of extension is justified in the inverse problem for reconstruction of code used to find vulnerabilities. There is presents a comprehensive comparison of syntax and language capabilities at the code of real functions.

Key words: modern society, information space, telecommunication devices, the language *C*, the expansion of the language, code efficiency, restore the code, the language description of the algorithms, security, vulnerability.

References

1. **Sociologija** potreblenija: osnovnye podhody / V. V. Radaev // Sociologicheskie Issledovanija. – 2005. – № 6. – S. 5–18.
2. **Osnovnye** koncepcii jazykov programmirovanija / U. S. Robert. – M. : «Vil'jams», 2001. – 672 s.
3. **Metod** algoritmizacii mashinnogo koda telekommunikacionnyh ustrojstv / M. V. Bujnevich, K. E. Izrailov // Telekommunikacii. – 2012. – № 12. – C. 2–6.
4. **Avtomatizirovannoe** sredstvo algoritmizacii mashinnogo koda telekommunikacionnyh ustrojstv / M. V. Bujnevich, K. E. Izrailov // Telekommunikacii. – 2013. – № 6. – C. 2–9.

Израилов Константин Евгеньевич – аспирант ФГОБУ ВПО «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», konstantin.izrailov@mail.ru

УДК 621.396

К. А. Польщиков

Донбасская государственная машиностроительная академия

МЕТОД НЕЙРО-НЕЧЕТКОГО УПРАВЛЕНИЯ ИНТЕНСИВНОСТЬЮ ПОВТОРНЫХ ПЕРЕДАЧ В ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ

интенсивность повторных передач, телекоммуникационная сеть, нейро-нечеткая система, доставка данных, информационный блок

Для обеспечения достоверной доставки данных в телекоммуникационных сетях применяются повторные передачи потерянных или искаженных информационных блоков (кадров, пакетов, сегментов). В телекоммуникационных сетях, в которых часто происходят искажения передаваемых данных и перегрузки, например, в беспроводных сетях, повторные передачи потерянной информации создают существенный трафик [1]. В таких сетях эффективность управления интенсивностью повторных передач имеет большое значение. Необоснованно малая интенсивность повторных передач приводит увеличению времени восполнения недостающих информационных блоков и замедлению доставки всего сообщения (файла). Если же указанная интенсивность неоправданно высока, то лишние повторные передачи перегружают сеть и негативно влияют на характеристики доставки данных.

Управление интенсивностью повторных передач в телекоммуникационных сетях реализуется путем выбора значений тайм-аута ζ. В соответствии с протоколом высокоуровневого управления линией связи (High-level Data Link Control, HDLC) [2] в момент отправки в сеть очередного информационного блока начинается отсчет тайм-аута ζ, в течение которого источнику следует ожидать получения подтверждения успешной доставки этого информационного блока адресату, не осуществляя повторную передачу. Если подтверждение поступит к источнику раньше, чем закончится тайм-аут, то повторная передача соответствующего информационного блока не выполняется, а источник отправляет адресату следующий информационный блок. Если время тайм-аута будет исчерпано, а подтверждения к этому моменту источник не получит, то осуществляется повторная передача нужного информационного блока. Протоколом HDLC предусмотрено управление передачей данных с использованием фиксированного тайм-аута. Недостатком этого метода является отсутствие адекватной реакции источника на изменения време-

ни ожидания подтверждений, вызванные неравномерностью нагрузки сети в разные моменты времени.

Известен более эффективный метод управления интенсивностью повторных передач, который нашел применение в протоколе управления передачей (Transmission Control Protocol, TCP) и предусматривает использование адаптивного тайм-аута [3, 4]. Значение параметра ζ на передающей стороне устанавливается для каждого TCP-соединения. В сетях, работающих в соответствии с протоколом TCP, выбор значения тайм-аута осуществляется с помощью метода Джекобсона. Безусловным преимуществом этого метода является простота вычислений. Однако методу Джекобсона присущи существенные недостатки. Во-первых, данный метод основывается на приблизительном расчете тайм-аута и не обеспечивает установку в сети рациональных значений параметра ζ. Во-вторых, при расчете тайм-аута используются фиксированные значения коэффициентов, т. е. недостаточно учитываются особенности той или иной сети. Несовершенство метода Джекобсона приводит к тому, что при высокой вероятности потери данных вследствие искажений или перегрузок в сети, доставка сообщений существенно замедляется [5].

Вышеизложенные аргументы свидетельствует об актуальности научно-технической задачи, состоящей в разработке эффективного метода управления интенсивностью повторных передач в телекоммуникационной сети.

Сущность предлагаемого в статье метода заключается в прогнозировании времени ожидания подтверждений на основе применения нечеткой нейронной сети. Метод нейро-нечеткого управления интенсивностью повторных передач включает следующие этапы:

1) для каждого отправленного источником информационного блока измеряется значение времени ожидания подтверждения;

2) на входы нейро-нечеткой системы в качестве входных переменных подаются величина M_i (текущее значение времени ожидания подтверждения) и величины M_{i-1} и M_{i-2} (значения времени ожидания двух предыдущих подтверждений);

3) выполняются процедуры нечеткого вывода, в результате которых определяется значение $\tilde{M}_{i+1}$ (прогнозируемое время ожидания подтверждения на очередной информационный блок) на выходе нейро-нечеткой системы;

4) вычисляется значение тайм-аута для очередного информационного блока по формуле:

$$\zeta_{i+1} = \tilde{M}_{i+1} + \Delta\zeta, \qquad (1)$$

где $\Delta\zeta > 0$ – постоянная величина, которая прибавляется к значению $\widetilde{M}_{i+1}$ для устранения возможных недооценок тайм-аута вследствие недостаточно точного прогнозирования.

5) таймером повторной передачи начинается отсчет времени ζ_{i+1} в момент отправки источником очередного информационного блока.

Предлагаемый метод основан на функционировании нейро-нечеткой системы, служащей для прогнозирования величины $\widetilde{M}_{i+1}$. Построение указанной системы целесообразно осуществлять по критерию минимальной сложности. Данному критерию соответствуют следующие параметры синтеза нейро-нечетких систем: алгоритм нечеткого вывода – Сугено 0-го порядка [6], количество функций принадлежности для каждой входной величины – 2, форма функций принадлежности для каждой входной величины – треугольная, алгоритм обучения нейронов – обратного распространения ошибки [7].

Рассмотрим пример синтеза нейро-нечеткой системы прогнозирования величины $\widetilde{M}_{i+1}$. Для настройки весов нейронов сформированы обучающие данные в виде матрицы, содержащей 97 строк:

$$\begin{Vmatrix} M_3 & M_2 & M_1 & M_4 \\ M_4 & M_3 & M_2 & M_5 \\ . & . & . & . \\ M_{99} & M_{98} & M_{97} & M_{100} \end{Vmatrix}. \qquad (2)$$

При составлении матрицы (2) использовались данные о значениях M_1, M_2, …, M_{100}, полученные в ходе имитационного эксперимента, результаты которого представлены на рисунке 1.

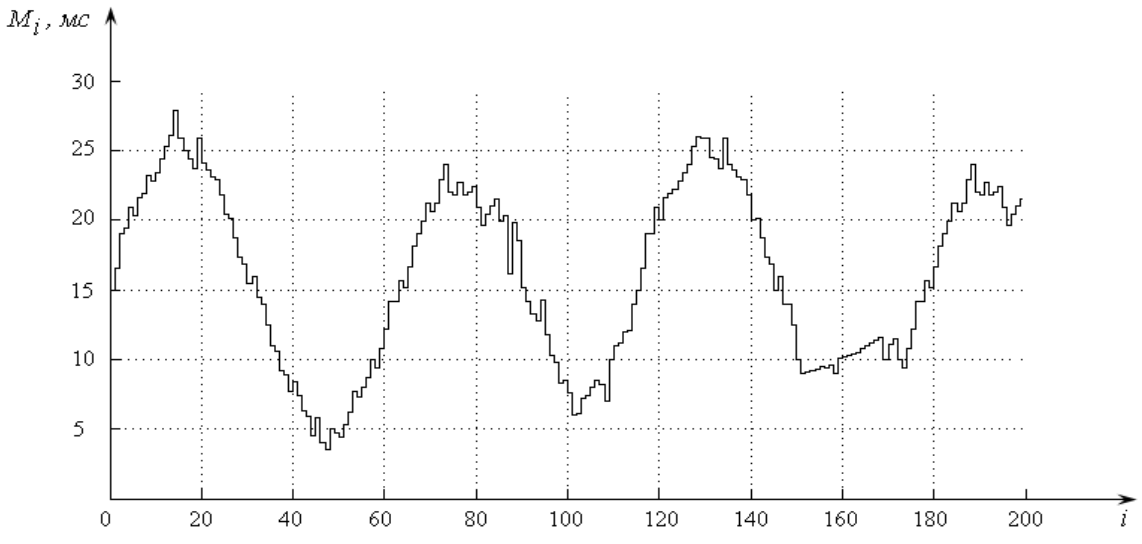

Рис. 1. Значения величины M_i

Для настройки нейро-нечеткой системы потребовалось 8 циклов обучения. В результате получены выражения для функций принадлежности входных величин:

$$\mu_1(M_i)=\begin{cases}1, & M_i<3,75;\\ \dfrac{26,74-M_i}{22,99}, & 3,75\le M_i\le 26,74;\\ 0, & M_i>26,74;\end{cases} \quad (3)$$

$$\mu_2(M_i)=\begin{cases}0, & M_i<3,63;\\ \dfrac{M_i-3,63}{24,67}, & 3,63\le M_i\le 28,3;\\ 1, & M_i>28,3;\end{cases} \quad (4)$$

$$\mu_1(M_{i-1})=\begin{cases}1, & M_{i-1}<3,566;\\ \dfrac{27,57-M_{i-1}}{23,82}, & 3,75\le M_{i-1}\le 27,57;\\ 0, & M_{i-1}>27,57;\end{cases} \quad (5)$$

$$\mu_2(M_{i-1})=\begin{cases}0, & M_{i-1}<3,535;\\ \dfrac{M_{i-1}-3,535}{24,335}, & 3,535\le M_{i-1}\le 27,87;\\ 1, & M_{i-1}>27,87;\end{cases} \quad (6)$$

$$\mu_1(M_{i-2})=\begin{cases}1, & M_{i-2}<3,594;\\ \dfrac{27,83-M_{i-2}}{24,236}, & 3,594\le M_{i-2}\le 27,83;\\ 0, & M_{i-2}>27,83;\end{cases} \quad (7)$$

$$\mu_1(M_{i-2})=\begin{cases}0, & M_{i-2}<3,508;\\ \dfrac{M_{i-2}-3,508}{24,392}, & 3,508\le M_{i-2}\le 27,9;\\ 1, & M_{i-2}>27,9;\end{cases} \quad (8)$$

а также значения коэффициентов индивидуальных выводов нечетких правил: $c_1=3,837$, $c_2=-6,012$, $c_3=7,257$, $c_4=8,733$, $c_5=31,48$, $c_6=20,88$, $c_7=29,89$, $c_8=26,36$.

Структура синтезированной нейро-нечеткой системы представлена на рисунке 2. Функционирование данной системы основано на использовании базы нечетких правил:

$$Если\ (M_i = A_1^1)\ и\ (M_{i-1} = A_2^1)\ и\ (M_{i-2} = A_3^1),\ то\ \widetilde{M}_{i+1} = c_1; \quad (9)$$

$$Если\ (M_i = A_1^1)\ и\ (M_{i-1} = A_2^1)\ и\ (M_{i-2} = A_3^2),\ то\ \widetilde{M}_{i+1} = c_2; \quad (10)$$

$$Если\ (M_i = A_1^1)\ и\ (M_{i-1} = A_2^2)\ и\ (M_{i-2} = A_3^1),\ то\ \widetilde{M}_{i+1} = c_3; \quad (11)$$

$$Если\ (M_i = A_1^1)\ и\ (M_{i-1} = A_2^2)\ и\ (M_{i-2} = A_3^2),\ то\ \widetilde{M}_{i+1} = c_4; \quad (12)$$

$$Если\ (M_i = A_1^2)\ и\ (M_{i-1} = A_2^1)\ и\ (M_{i-2} = A_3^1),\ то\ \widetilde{M}_{i+1} = c_5; \quad (13)$$

$$Если\ (M_i = A_1^2)\ и\ (M_{i-1} = A_2^1)\ и\ (M_{i-2} = A_3^2),\ то\ \widetilde{M}_{i+1} = c_6; \quad (14)$$

$$Если\ (M_i = A_1^2)\ и\ (M_{i-1} = A_2^2)\ и\ (M_{i-2} = A_3^1),\ то\ \widetilde{M}_{i+1} = c_7; \quad (15)$$

$$Если\ (M_i = A_1^2)\ и\ (M_{i-1} = A_2^2)\ и\ (M_{i-2} = A_3^2),\ то\ \widetilde{M}_{i+1} = c_8, \quad (16)$$

где A_1^1 – терм номер 1 входной величины M_i;

A_1^2 – терм номер 2 входной величины M_i;

A_2^1 – терм номер 1 входной величины M_{i-1};

A_2^2 – терм номер 2 входной величины M_{i-1};

A_3^1 – терм номер 1 входной величины M_{i-2};

A_3^2 – терм номер 2 входной величины M_{i-2}.

Нейро-нечеткая система прогнозирования величины $\widetilde{M}_{i+1}$ состоит из четырех слоев. Первый слой выполняет процедуру фаззификации, в результате которой для конкретных значений M_i^*, M_{i-1}^* и M_{i-2}^* входных величин по выражениям (3)–(8) вычисляются значения $\mu_1\left(M_i^*\right)$, $\mu_2\left(M_i^*\right)$, $\mu_1\left(M_{i-1}^*\right)$, $\mu_2\left(M_{i-1}^*\right)$, $\mu_1\left(M_{i-2}^*\right)$ и $\mu_2\left(M_{i-2}^*\right)$.

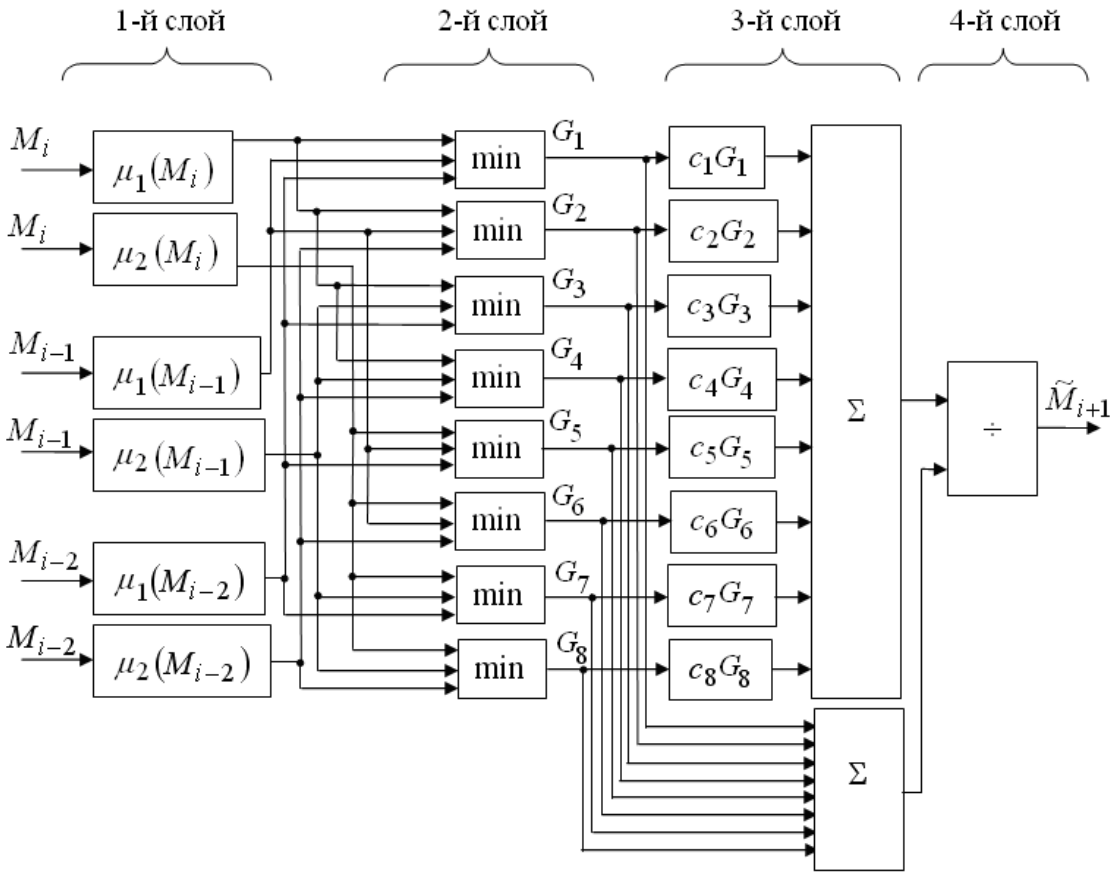

Рис. 2. Структура нейро-нечеткой системы прогнозирования величины $\tilde{M}_{i+1}$

Вторым слоем системы выполняется агрегирование, в процессе которого используются выражения:

$$G_1 = \mu_1(M_i^*) \wedge \mu_1(M_{i-1}^*) \wedge \mu_1(M_{i-2}^*); \quad (17)$$

$$G_2 = \mu_1(M_i^*) \wedge \mu_1(M_{i-1}^*) \wedge \mu_2(M_{i-2}^*); \quad (18)$$

$$G_3 = \mu_1(M_i^*) \wedge \mu_2(M_{i-1}^*) \wedge \mu_1(M_{i-2}^*); \quad (19)$$

$$G_4 = \mu_1(M_i^*) \wedge \mu_2(M_{i-1}^*) \wedge \mu_2(M_{i-2}^*); \quad (20)$$

$$G_5 = \mu_2(M_i^*) \wedge \mu_1(M_{i-1}^*) \wedge \mu_1(M_{i-2}^*); \quad (21)$$

$$G_6 = \mu_2(M_i^*) \wedge \mu_1(M_{i-1}^*) \wedge \mu_2(M_{i-2}^*); \quad (22)$$

$$G_7 = \mu_2(M_i^*) \wedge \mu_2(M_{i-1}^*) \wedge \mu_1(M_{i-2}^*); \quad (23)$$

$$G_8 = \mu_2(M_i^*) \wedge \mu_2(M_{i-1}^*) \wedge \mu_2(M_{i-2}^*). \quad (24)$$

Третьим слоем нейронов выполняется активизация, а также часть процедуры дефаззификации – вычисляются сумма результатов агрегирования $\sum_{r=1}^{8} G_r$ и взвешенная сумма результатов агрегирования $\sum_{r=1}^{8} c_r G_r$.

С помощью четвертого слоя выполняется заключительная часть процедуры дефаззификации, состоящая в вычислении искомого значения $\widetilde{M}_{i+1}^{*}$ выходной переменной по формуле:

$$\widetilde{M}_{i+1}^{*} = \frac{\sum_{r=1}^{8} c_r G_r}{\sum_{r=1}^{8} G_r}. \qquad (25)$$

Реализацию предложенного метода рассмотрим для условий, при которых номера информационных блоков, требующих повторной передачи, соответствуют тем значениям i, для которых на рисунке 3 верно равенство $N_i = 1$.

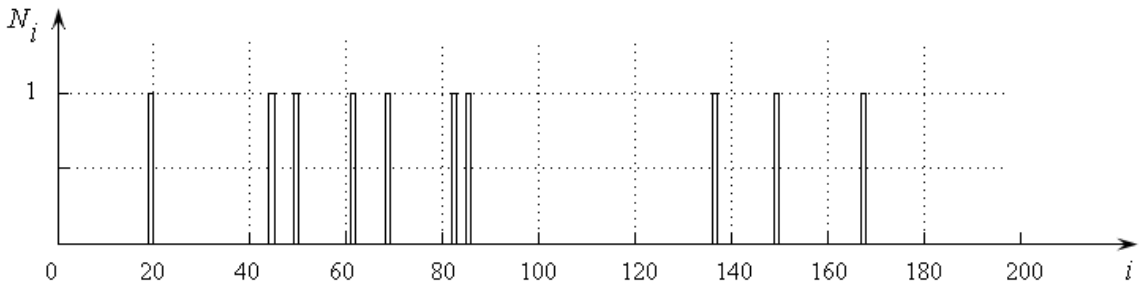

Рис. 3. Значения величины N_i

Информационные блоки, для которых $N_i = 1$, не были успешно доставлены адресату с первой попытки и передавались повторно. В каждом таком случае время доставки сообщения в телекоммуникационной сети увеличивалось на значение ζ_i.

В результате выполнения имитационного эксперимента величина времени ожидания каждого подтверждения M_i совместно с величиной тайм-аута ζ_i, установленного для каждого соответствующего информационного блока, принимали значения, представленные на рисунке 4.

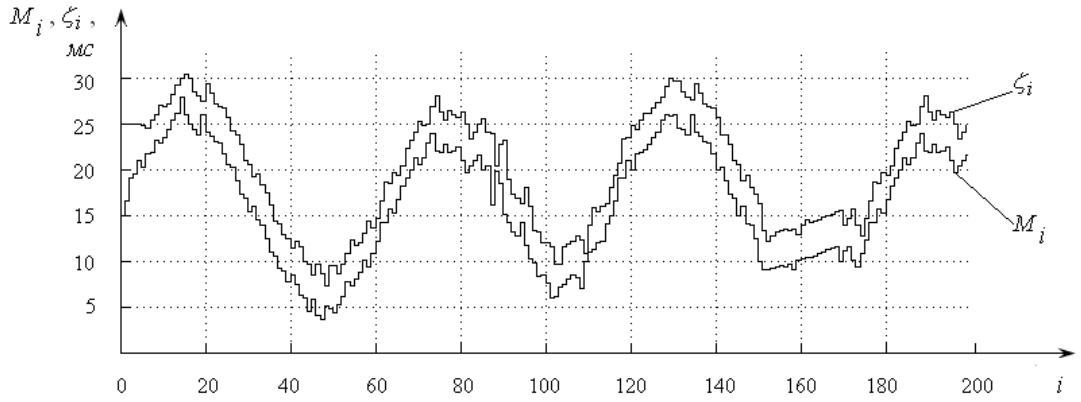

Рис. 4. Текущие значения M_i совместно со значениями ζ_i

Анализ результатов данного эксперимента показывает, что при управлении интенсивностью повторных передач на основе предложенного метода не были зафиксированы случаи лишних повторных передач, т. е. для всех i было выполнено условие $\zeta_i > M_i$. При этом разность $(\zeta_i - M_i)$ принимала относительно малые значения, не превышающие 5 мс. Кроме того, для ожидания подтверждений на потерянные информационные блоки в процессе передачи сообщения бесполезно затрачено 26,19 % времени. При управлении интенсивностью повторных передач на основе метода Джекобсона для аналогичных исходных данных значение этого показателя существенно выше (41,79 %) [5]. Следовательно, результаты имитационного эксперимента показали, что применение предложенного метода позволяет сократить время передачи сообщения на 15,6 %, то есть существенно повысить эффективность доставки данных в телекоммуникационной сети.

Проведение других многочисленных имитационных экспериментов с различными исходными данными позволило установить, что использование нейро-нечеткого управления интенсивностью повторных передач дает выигрыш во времени передачи сообщения в пределах от 8,2 до 19,4 %. Таким образом, применение предложенного метода в телекоммуникационной сети позволит осуществлять эффективное управление интенсивностью повторных передач.

С целью уменьшения среднего времени доставки сообщения в телекоммуникационной сети разработан метод управления интенсивностью повторных передач. Сущность метода заключается в применении нейро-нечеткой системы для прогнозирования времени, необходимого для получения подтверждения на отправленный источником информационный блок. Данный метод предусматривает в качестве переменных, значения которых поступают на вход нейро-нечеткой системы, использовать те-

кущую величину времени ожидания подтверждения и величины времени ожидания двух предыдущих подтверждений.

Нейро-нечеткая система состоит из четырех слоев, выполняющих процедуры фаззификации, агрегирования, активизации и дефаззификации. Для настройки весов нейронов первого и третьего слоев используются обучающие данные, полученные в ходе экспериментальных исследований. Результаты имитационных экспериментов показали, что применение разработанного метода управления тайм-аутом в телекоммуникационной сети позволяет сократить время передачи сообщения на 8,2–19,4 % и минимизировать количество лишних повторных передач.

Библиографический список

1. **Функциональная** модель управления интенсивностью потоков данных в мобильной радиосети специального назначения / К. А. Польщиков // Научный вестник ДГМА. – Краматорск: ДГМА, 2012. – № 1 (9Е). – С. 127–135.

2. **ISO TC 97/SC 6/N 3843**. Final Text of ISO 4335/DADI: Data Communication – High-Level Data Link Control Procedures – Consolidation of Elements of Procedures – Addendum 1. – 1986.

3. **Congestion** Avoidance and Control / V. Jacobson // Proceedings of ACM SIGCOMM'88. – Stanford, 1988. – PP. 314–329.

4. **Computing** TCP's Retransmission Timer / V. Paxton, M. Allman // RFC 2988. – November, 2000.

5. **Математическая** модель процесса управления таймером повторной передачи в информационной сети, функционирующей в соответствии с протоколом TCP / К. А. Польщиков // Системы управления, навигации и связи. – К. : ЦНИИНУ, 2008. – Вып. 4 (8). – С. 139–141.

6. **Fuzzy** Identification of Systems and Its Applications to Modeling and Control / T. Takagi, M. Sugeno // IEEE Transactions on Systems, Man and Cybernetics. – 1985. – Vol. 15, N. 1. – PP. 116–132.

7. **Learning** Internal Representations by Error Propagation / D. E. Rumelhart, G. E. Hilton, R. J. Williams // In Parallel Distributed Processing, t. 1. – Cambridge: M.I.T. Press, 1986.

Аннотация

В статье представлено решение актуальной научно-технической задачи, состоящей в разработке эффективного метода управления интенсивностью повторных передач в телекоммуникационной сети. Предложенный метод основан на применении нейро-нечеткой системы и позволяет существенно сократить время передачи сообщения.

K. Polschykov

Donbass State Engineering Academy

THE METHOD OF NEURO-FUZZY CONTROL OF THE REPEATED TRANSMISSIONS INTENSITY IN TELECOMMUNICATION NETWORK

Abstract

The paper presents a decision of an actual scientific and technical problem consisting in the development of an effective method of the repeated transmissions intensity control in telecommunication network. The proposed method is based on neuro-fuzzy system and can significantly reduce the transmission of messages.

Key words: intensity of repeated transmissions, telecommunications network, neuro-fuzzy system, delivery of data, information block.

References

1. **Funkcional'naja** model' upravlenija intensivnost'ju potokov dannyh v mobil'noj radioseti special'nogo naznachenija / K. A. Pol'shhikov // Nauchnyj vestnik DGMA. – Kramatorsk: DGMA, 2012. – № 1 (9E). – S. 127–135.

2. **ISO TC 97/SC 6/N 3843**. Final Text of ISO 4335/DADI: Data Communication – High-Level Data Link Control Procedures – Consolidation of Elements of Procedures – Addendum 1. – 1986.

3. **Congestion** Avoidance and Control / V. Jacobson // Proceedings of ACM SIGCOMM'88. – Stanford, 1988. – PP. 314–329.

4. **Computing** TCP's Retransmission Timer / V. Paxton, M. Allman // RFC 2988. – November, 2000.

5. **Matematicheskaja** model' processa upravlenija tajmerom povtornoj peredachi v informacionnoj seti, funkcionirujushhej v sootvetstvii s protokolom TSR / K. A. Pol'shhikov // Sistemy upravlenija, navigacii i svjazi. – K. : CNIINU, 2008. – Vyp. 4 (8). – S. 139–141.

6. **Fuzzy** Identification of Systems and Its Applications to Modeling and Control / T. Takagi, M. Sugeno // IEEE Transactions on Systems, Man and Cybernetics. – 1985. – Vol. 15, N. 1. – PP. 116–132.

7. **Learning** Internal Representations by Error Propagation / D. E. Rumelhart, G. E. Hilton, R. J. Williams // In Parallel Distributed Processing, t. 1. – Cambridge: M.I.T. Press, 1986.

Польщиков Константин Александрович – кандидат технических наук, доцент кафедры «Компьютерные информационные технологии» Донбасской государственной машиностроительной академии, konspol@rambler.ru

УДК 004.772

А. С. Солозобов

Санкт-Петербургский государственный университет

ПЕРЕДАЧА ДАННЫХ В IP-СЕТЯХ С ИСПОЛЬЗОВАНИЕМ ДИНАМИЧЕСКОЙ МАРШРУТИЗАЦИИ ОТ ИСТОЧНИКА

IP, IP-сеть, передача данных, маршрутизация от источника

Дефицит пропускной способности каналов связи между прикладными процессами, разнесёнными в пространстве на значительные расстояния, является серьёзной проблемой. Его причинами могут быть фактическое исчерпание пропускной способности физического и канального уровней соединения, нестабильность работы каналов связи и многое другое. При этом пользователь (процесс), столкнувшийся с нехваткой пропускной способности установленного соединения, как правило, оказывается не в состоянии решить её в одиночку.

Первым этапом на пути к решению проблемы становится поиск её причин. Для этого, используя различные системы мониторинга трафика, собирают и анализируют статистические данные работы сети, используя специальные генераторы трафика, проводят нагрузочные тесты отдельных её участков. На втором этапе, когда проблемный участок сети уже обнаружен, производится перенастройка и обновление его программной и аппаратной составляющих с целью увеличения пропускной способности, задействуются дополнительные каналы передачи данных для снятия части нагрузки с «проблемного» участка. Может потребоваться не одна итерация такого процесса поиска и устранения причин низкой пропускной способности соединения, до того, как будут получены удовлетворительные результаты.

Для определения причин проблемы необходимо обладать комплексным взглядом на принципы построения используемых сетей, знанием протоколистики их функционирования, топологии и характеристик связей. Даже локализованный и классифицированный источник ограничения пропускной способности может оказаться физически недосягаемым. Решение проблемы часто требует административного и финансового ресурса, согласования со сторонними лицами и организациями, отчего затягивается, а иногда и останавливается, будучи осложнённым бюрократическими, экономическими и другими причинами. А до тех пор, пока проблема не будет решена, неподготовленному пользователю приходится с ней смириться.

Более подготовленный пользователь может воспользоваться особенностями протокола IP и современных IP-сетей для самостоятельного разрешения возникшей проблемы.

При проектировании IP-сетей принимаются меры для повышения их отказоустойчивости. Используются приёмы создания запасных каналов связи: дублирование связей, топологические кольца, утолщённые деревья. Если представить такую сеть в виде графа связности узлов, то в нём обнаружится множество циклов. На рисунке 1 приведен пример участка IP-сети, просканированного на предмет связности узлов сети.

Данный граф отражает связность узлов сети именно на уровне протокола IP, и репрезентативность представленных связей может страдать от работы механизмов туннелирования при передаче данных. Всё же, во многих случаях данные от источника к получателю могут быть переданы по множеству альтернативных маршрутов, минимально пересекающемуся по узлам и рёбрам графа со стандартным маршрутом передачи данных, определяемым сетью.

В протоколе IP заложена возможность передачи данных по нестандартным маршрутам. Для этого в IP-пакете можно указать специальную опцию маршрутизации от источника в виде списка IP-адресов маршрутизирующих узлов сети, через которые данный пакет должен пересылаться при доставке адресату (в протоколе IPv4 они включаются в дополнительное поле опций заголовка пакета, в протоколе IPv6 опции указываются в отдельном заголовке пакета) [1, 2]. К сожалению, использование данной опции протокола IP может административно быть запрещено из-за опасений её недобросовестного использованияс целью взлома некорректно реализованных программных продуктов, обрабатывающих IP-пакеты.

Действительно, в ходе работ по реализации описываемого ниже подхода использования опций маршрутизации от источника было выявлено, что операционные системы Windows 7 и Ubuntu 12.10 (Linux 3.5.0) обрабатывают их некорректно. Разработчикам этих операционных систем были отправлены подробные отчёты о несоответствующей описанию протокола IP работе сетевого стека. К сожалению, автору статьи не известно могут ли и насколько данные ошибки разработчиков быть использованы злонамеренными участниками сети.

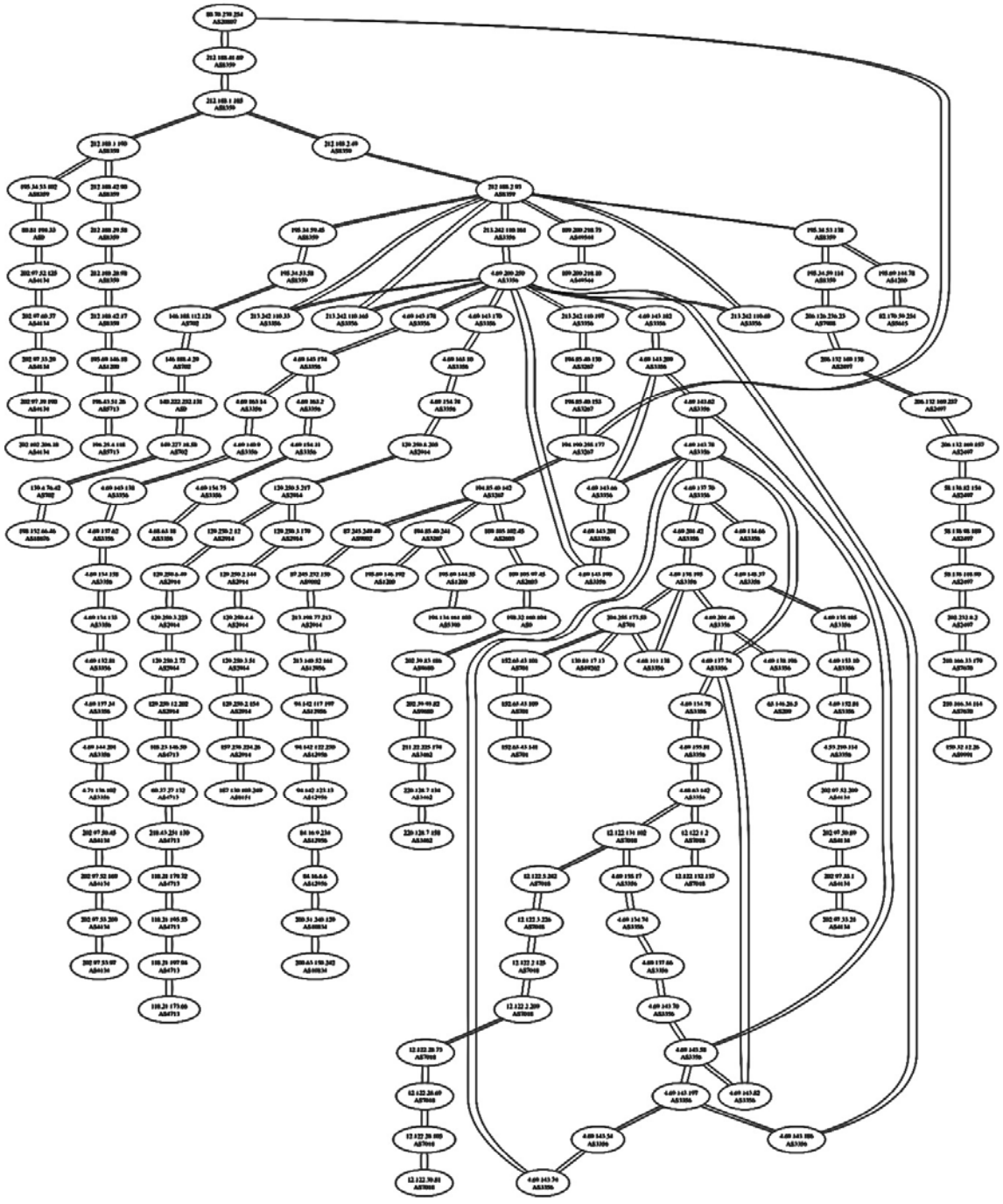

Рис. 1. Пример графа связности узлов Интернета

Описание подхода

Если в сети разрешено использование опций маршрутизации от источника, то при обнаружении недостаточной пропускной способности соединения пользователь может провести сканирование связности узлов IP-сети в направлении целевого узла соединения, выявить в полученном графе, по возможности, наиболее различные альтернативные маршруты передачи данных и использовать их для распределения генерируемой при передаче данных нагрузки, и тем самым повышая пропускную спо-

собность соединения. Такой подход окажется особенно выигрышным в случае перегрузки каналов стандартного маршрута передачи данных, когда низкая пропускная способность соединения обуславливается большой задержкой при передаче пакетов, превышением времени ожидания на них ответа и многократной их переотправкой. Распределение передаваемых данных между стандартным и альтернативным маршрутами позволит снизить частоту переотправки пакетов по таймауту из-за снижения нагрузки на стандартный маршрут передачи данных и задействования, возможно не загруженных, альтернативных маршрутов.

Суть такого способа организации передачи данных в IP-сетях близка к концепции построения сетей, в которой функцию маршрутизатора выполняет сам отправитель на основе информации о топологии сети, получаемой от других узлов. Воплощением этой концепции является протокол WRAP [3], работающий поверх протокола IP между транспортным и сетевым уровнями модели OSI. Заголовок WRAP-пакета несёт в себе записанную отправителем последовательность IP-адресов, через которые пакет уже прошел и ещё должен будет пройти. Этот подход позволяет пользователям выбирать каналы движения трафика и тем самым достигать необходимых характеристик виртуального канала связи, а сетевым администраторам реализовывать более сложные механизмы фильтрации трафика.

Реализация

Программная реализация описанного подхода была выполнена на языке C++ с использованием библиотеки подпрограмм BOOST ASIO.

Для наиболее корректного выбора различных маршрутов передачи данных определение связности узлов сети производилось сразу на двух уровнях: на уровне протокола IP (IP-адреса), и на уровне протокола BGP (номера автономных систем). Отображение IP-адресов в номера автономных систем было выполнено на основе обновляемой раз в три месяца базы IP – ASN соответствий GeoLite ASN с исправленными коллизиями пространств IP-адресов.

Определение связности узлов сети производилось методом трассировки. Через различные промежуточные узлы сети целевому узлу отправлялись IP-пакеты с различными значениями поля временем жизни, и анализировались ответы узлов. В качестве содержимого IP-пакетов использовались пакеты протокола UDP, как не требующие привилегий суперпользователя для отправки [4].

Для определения оптимальных с точки зрения производительности и полноты параметров трассировки маршрутов была проведена экспериментальная трассировка маршрутов до случайных IP-адресов Интернета.ICMP-ответы на трассировочные UDP-пакеты ожидались не более двух секунд. В случае отсутствия ответа пакет переотправлялся до пяти

раз. После последовательности из семи подряд идущих не ответивших узлов трассировка данного IP-адреса прекращалась. Всего в ходе трассировки было обнаружено 2090 узлов.

Параметр максимального времени ожидания ответа трассируемого узла был выбран равным 800 миллисекундам. В него в среднем уложилось 99 % узлов. График распределения количества узлов по времени ответа, в которое они уложились, приведён на рисунке 2.

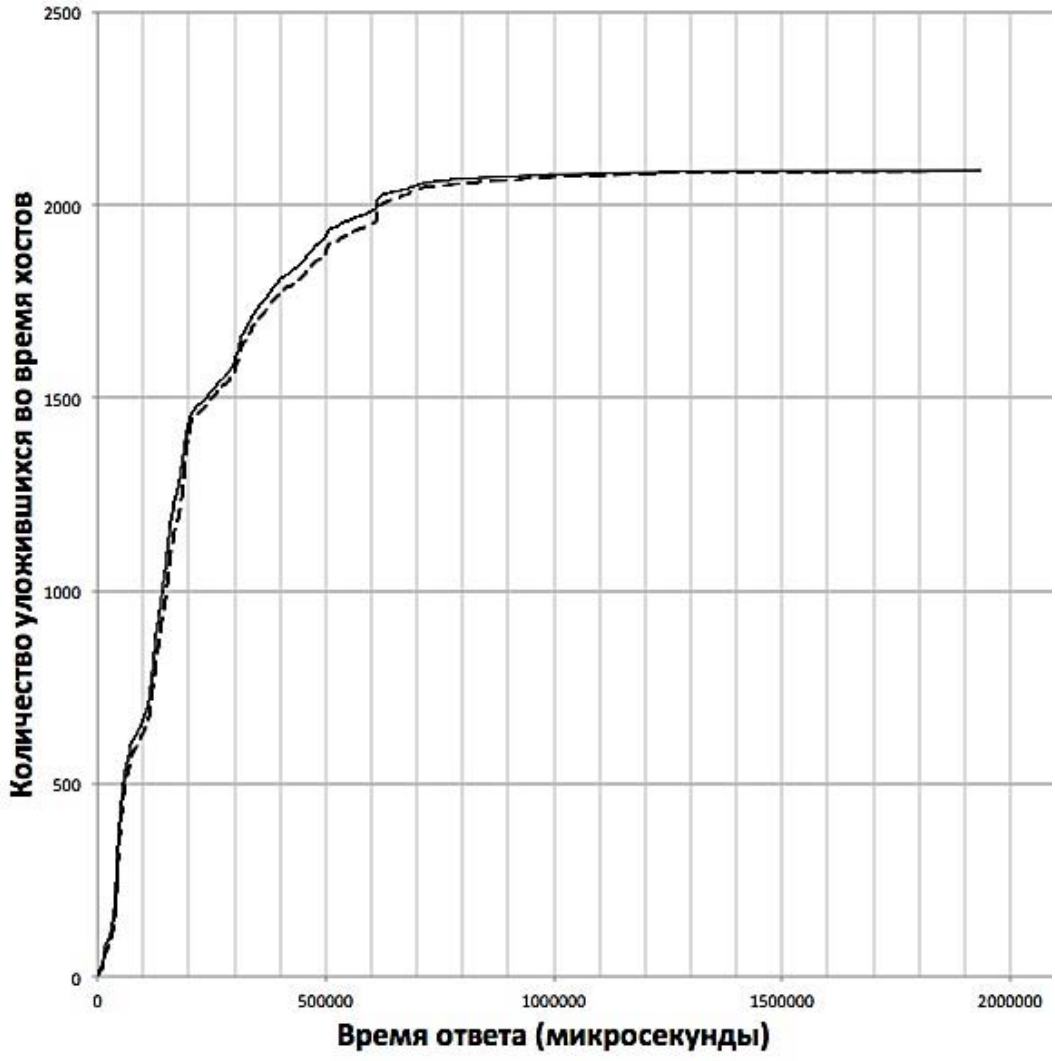

Рис. 2. Распределение количества узлов по времени ответа, в которое они уложились (сплошная линия – в среднем, пунктирная – в худшем случае)

В полученных трассах были обнаружены 134 различные последовательности узлов сети, намеренно не отвечавших на запросы, завершавшиеся узлом сети, отправлявшим ответ. Распределение количества последовательностей в зависимости от их длины приведено на рисунке 3.

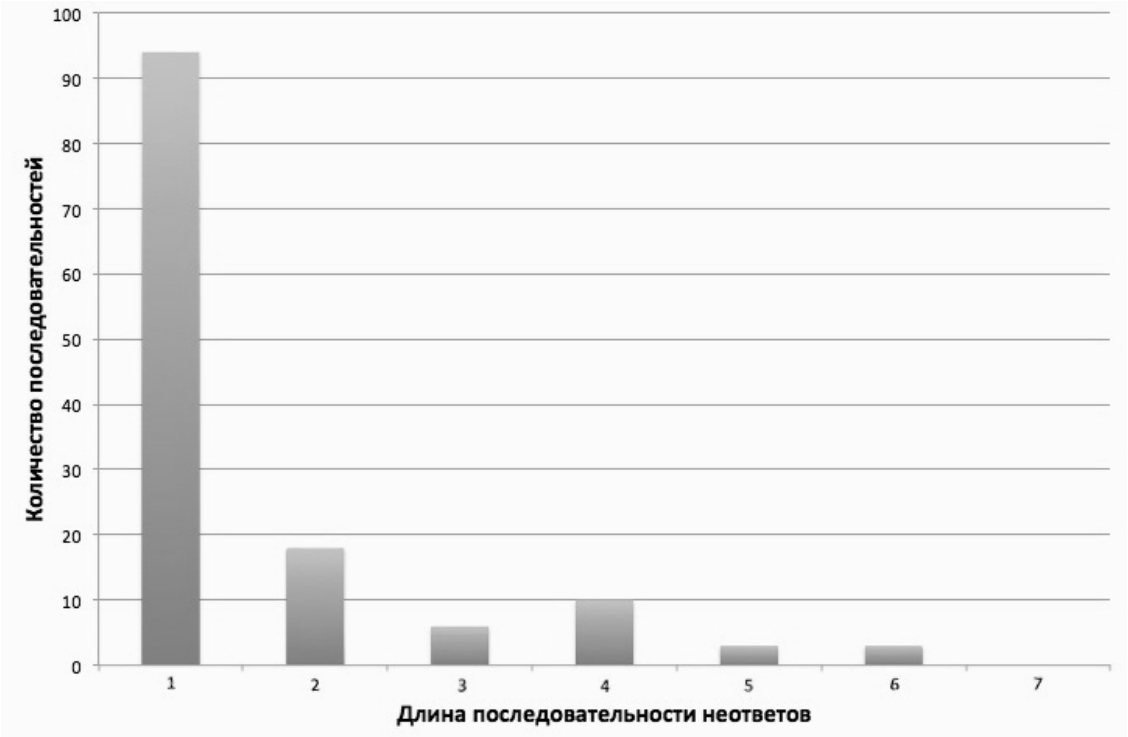

Рис. 3. Распределение количества последовательностей «неответов» по их длине

Параметр длины последовательности не отвечающих узлов, при котором трассировка останавливалась, был выбран равным 4. В него уложились 95 % всех полученных последовательностей «неответов».

Поиск на графе альтернативных путей передачи данных осуществлялся алгоритмом двунаправленного поиска в ширину от источника и получателя до получения необходимого количества маршрутов.

Эксперимент

В рамках эксперимента между двумя целевыми узлами (отправителем и получателем данных) хотелось получить избыточную в смысле количества маршрутов обмена данными сеть передачи данных с перегруженным основным маршрутом и возможностью передачи по сети IP-пакетов с опцией маршрутизации от источника.

Для эксперимента были построены две локальные сети, имеющие продублированную связь через два осуществляющих маршрутизацию компьютера (A и B). В одну локальную сеть был подключен отправитель данных, во вторую – получатель. Схема соединения сетевого оборудования приведена на рисунке 4.

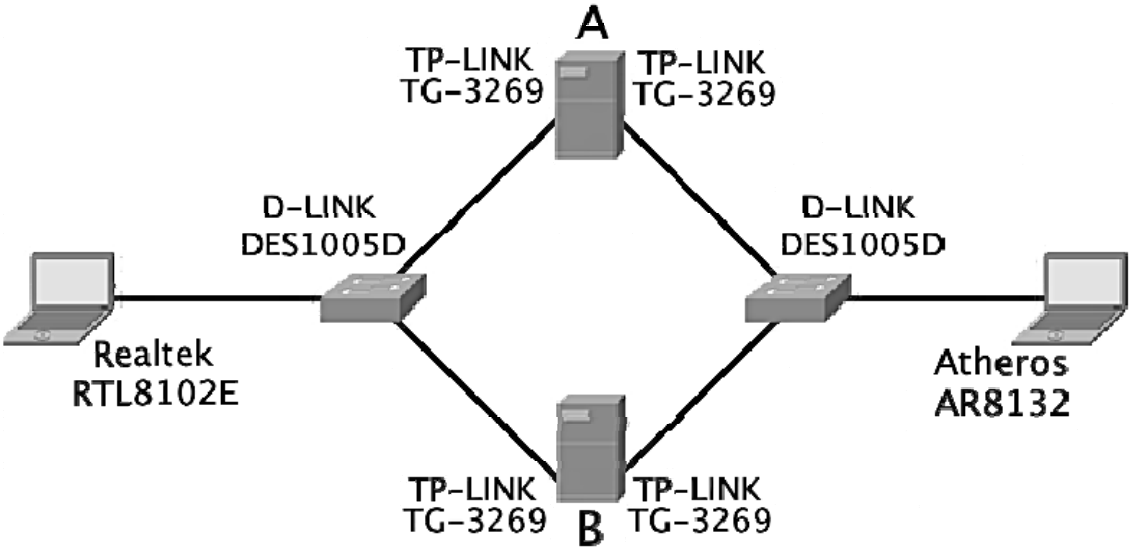

Рис. 4. Схема построения сетей в рамках эксперимента

Оба маршрутизирующих компьютера имели по две PCI Gigabit Ethernet сетевые карты, соединённые с разными локальными сетями, и работали под управлением операционной системы Windows XP Professional SP3 32bit 5.1.2600 в режиме маршрутизации поступающих пакетов.

У отправителя и получателя данных были установлены 100 Fast Ethernet сетевые. Компьютер A был настроен как шлюз по умолчанию у отправителя и получателя.

Эксперимент состоял в передаче одного гигабайта данных по протоколу TCP от отправителя к получателю. Передача выполнялась тремя разными способами:

– стандартный: данные отправлялись без использования опций маршрутизации от источника и передавались через компьютер A;

– альтернативный: данные отправлялись с использованием опций маршрутизации от источника и передавались через компьютер B;

– смешанный: данные отправлялись с использованием опций маршрутизации от источника и передавались одновременно через компьютеры A и B.

Передача с настройками каждого типа выполнялась десять раз. Полученные средние скорости передачи данных приведены в таблице.

ТАБЛИЦА. Средние скорости передачи данных различными способами

Способ передачи	Среднее время передачи	Средняя скорость передачи
стандартный	169 ± 4 секунды	6.0± 0.2 МБ/сек
альтернативный	172 ± 4 секунды	5.9 ± 0.2 МБ/сек
смешанный	120 ± 7 секунд	8.5 ± 0.5 МБ/сек

Скорость работы Fast Ethernet интерфейсов отправителя и получателя физически имеет ограничение сверху в 100 Мбит/сек. Был проведён тест передачи данных от отправителя к получателю по прямому соединению кабелем. Средняя скорость передачи данных составила 11,3 МБайт/сек, что означает, что максимальное значение пропускной способности сетевых карт отправителя и получателя в ходе эксперимента не было достигнуто. А значит, «узким» местом в сети передачи данных являлись маршрутизирующие компьютеры. С другой стороны, было замечено, что во время передачи данных ни один из компьютеров, участвовавших в передаче, не загружал свои процессоры больше чем на 80 %.Такое поведение может быть обусловлено медленной работой драйверов сетевых карт.

Из небольшого различия в пропускной способности стандартного и альтернативного способа передачи данных, можно сделать вывод, что программная обработка сетевого трафика центральным процессором не сильно замедляется обработкой опций маршрутизации от источника. Так как аппаратная обработка сетевого трафика выполняется в разы быстрее программной, а опции маршрутизации от источника из-за сложности своего формата требуют именно процессорной обработки, скорее всего характер их влияния на пропускную способность канала при аппаратной маршрутизации изменится не в лучшую сторону.

Скорость передачи данных смешанным способом в 1,4 раза больше скорости стандартного способа. Такой сравнительно небольшой прирост пропускной способности соединения связан со слишком медленной работой модуля, конфигурирующего соединения, и говорит о том, что в этом месте разработанную программу требуется усовершенствовать.

В целом, эксперимент показал, что в избыточной в смысле связей IP-сети с разрешенными опциями маршрутизации от источника предложенный подход, использующий исключительно «пользовательские» возможности сети, позволяет пользователю повысить пропускную способность соединения с целевым узлом сверх того, что ему по умолчанию предоставляет сеть.

Библиографический список

1. **RFC 791**: Internet Protocol [On-line] // Information Sciences Institute: [web]. [1981] // URL: http://www.rfc-editor.org/rfc/rfc791.txt.

2. **RFC 2460**: Internet Protocol, Version 6 [On-line] // S. Deering, Cisco, R. Hinden, Nokia: [web]. [1998] // URL: http://tools.ietf.org/html/rfc2460.

3. **Loose** Source Routing as a Mechanism for Traffic Policies [On-line] // Katerina Argyraki, David R. Cheriton: [web]. [2004] // URL: http://www.cs.umd.edu/class/spring2007/cmsc711/papers/lsrr-fdna04-paper.pdf.

4. **The Art** of Port Scanning [On-line] // Gordon Lyon: [web]. [1997] // URL: http://netsec.cs.northwestern.edu/media/readings/port_scanning.pdf.

Аннотация

В данной статье предложен подход увеличения пропускной способности виртуального канала передачи данных в IP-сетях с использованием опций маршрутизации от источника, описаны детали программной реализации данного подхода.

A. Solozobov

Saint-Petersburg State University

DATA TRANSMISSION IN IP NETWORKS USING LOOSE SOURCE ROUTING

Annotation

The approach of data transmission virtual channel capacity increase in IP networks using loose source routing and approach software implementation details are given in this article.

Key words: IP, IP network, data transmission, loose source routing.

References

1. **RFC 791**: Internet Protocol [On-line] // Information Sciences Institute: [web]. [1981] // URL: http://www.rfc-editor.org/rfc/rfc791.txt.

2. **RFC 2460**: Internet Protocol, Version 6 [On-line] // S. Deering, Cisco, R. Hinden, Nokia: [web]. [1998] // URL: http://tools.ietf.org/html/rfc2460.

3. **Loose** Source Routing as a Mechanism for Traffic Policies [On-line] // Katerina Argyraki, David R. Cheriton: [web]. [2004] // URL: http://www.cs.umd.edu/class/spring2007/cmsc711/papers/lsrr-fdna04-paper.pdf.

4. **The Art** of Port Scanning [On-line] // Gordon Lyon: [web]. [1997] // URL: http://netsec.cs.northwestern.edu/media/readings/port_scanning.pdf.

Солозобов Андрей Сергеевич – выпускник кафедры системного программирования математико-механического факультета ФГБОУ ВПО «Санкт-Петербургский государственный университет», Andrei.Solozobov@gmail.com

СИСТЕМНЫЙ АНАЛИЗ И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 681.883

С. М. Доценко, А. Г. Владыко

Санкт-Петербургский государственный университет телекоммуникаций им. М. А. Бонч-Бруевича

М. Б. Гладких

ЗАО «ТЕЛРОС»

ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА КОРРЕЛЯЦИОННОГО ОТКЛИКА ПРОРЕЖЕННОГО ГЧМ СИГНАЛА

ГЧМ сигнал, корреляционный отклик, мультипликативная гребневая функция

Естественный ход эволюции создал условия, при которых млекопитающие, вернувшиеся в водную среду (дельфины, киты и т. п.), должны были искать способ определять положение цели. Модель, выбранная эволюцией для наземного мира, не соответствовала реальностям водной среды обитания. Но после многих циклов эволюции был выработан механизм освещения подводной обстановки. В качестве основы такого механизма выступили сигналы с гиперболической частотной модуляцией (ГЧМ сигналы).

В данной статье будет рассмотрена модель ГЧМ сигнала[1] соответствующая используемой дельфинами. В ходе изучения данной модели будет вычислен корреляционный отклик и описаны свойства данной модели.

В действительности двух симметрий недостаточно для описания реальных воздействий океана, для введения дополнительных воздействий можно использовать прореженный сигнал.

Сначала построим аддитивную гребневую функцию. Введем переменные для описания этой функции:

KW – количество волн в одном элементарном импульсе;

T – период одного элементарного импульса;

D – сдвиг на количество волн;

KPS – количество волн в одном импульсе.

[1] Применение гармонического анализа для описания доплеровских и сдвиговых преобразований сигнала / В. А. Сапрыкин, Л. Н. Пронин. – М., 1985. – С. 4.

Для имитационного моделирования была принята следующая модель гребневой функции:

$$s(t) = \sum_{k=0}^{KPS-1} rect\left(\frac{t - k \cdot D \cdot T}{T \cdot KW}\right) \sin\left(2\pi \frac{t}{T}\right). \qquad (1)$$

На рисунке 1 представлена пара элементарных сигналов. Параметры сигналов определены в обозначениях.

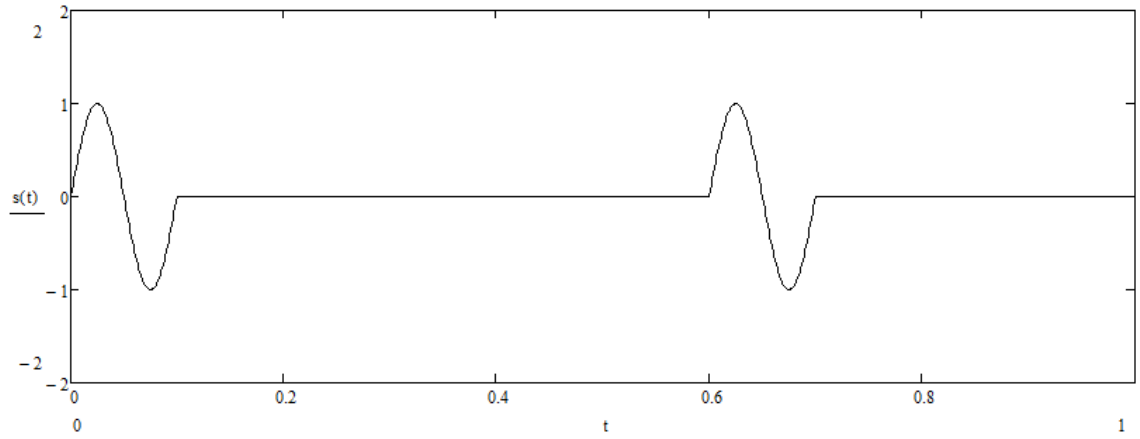

Рис. 1. Элементарные сигналы при численных значениях параметров $T = 0{,}01$, $KW = 1$, $KPS = 3$, $D = 6$

На рисунке 2 представлен сигнал, состоявший из трех элементарных сигналов. Параметры сигналов определены в обозначениях.

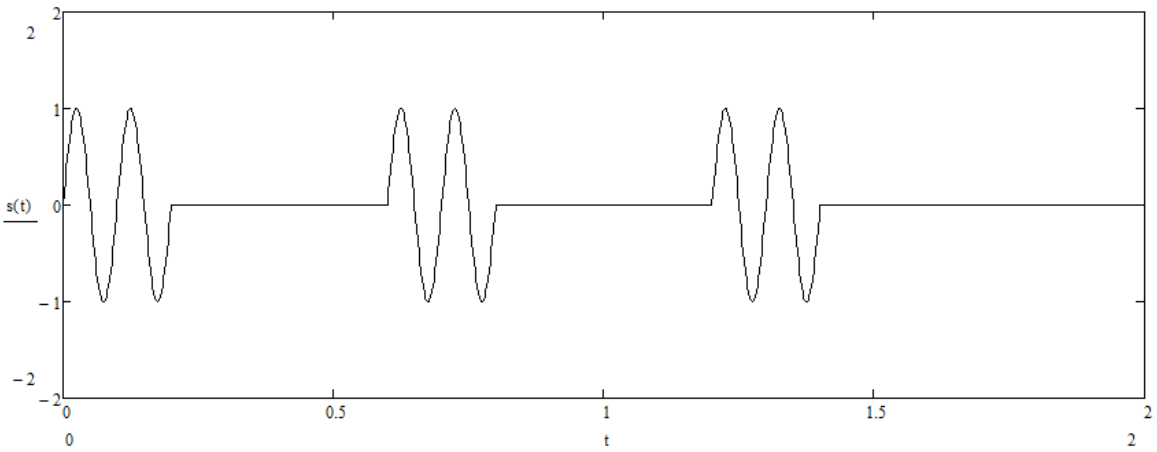

Рис. 2. Элементарные сигналы при численных значениях параметров $T = 0{,}01$, $KW = 2$, $KPS = 2$, $D = 6$

Теперь запишем аналогичную формулу для ГМЧ сигнала $\sin\left(2\pi\frac{\ln(t)}{T}\right)$ и гребневой функции $\sum_{k=0}^{KPS-1} rect\left(\frac{\ln(t)-k\cdot D\cdot T}{T\cdot KW}\right)$, взаимодействие этих сигналов, для лучшего восприятия, приведено на рисунке 3.

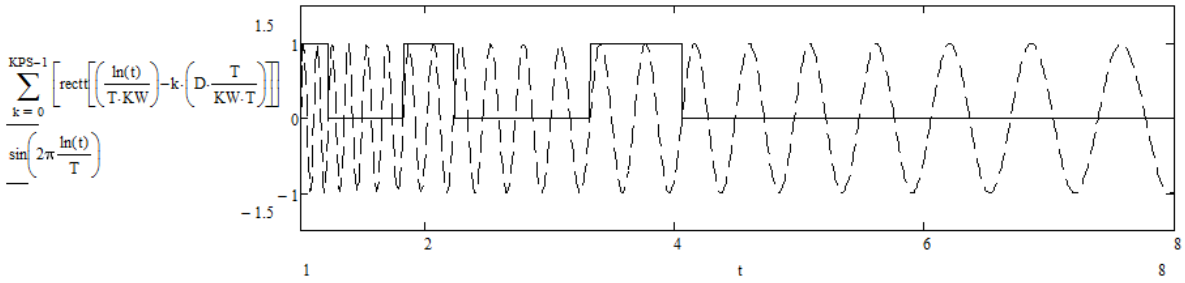

Рис. 3. Наложение сигналов при численных значениях параметров $T=0{,}01$, $KW=2$, $KPS=2$, $D=6$

В итоге для имитационного моделирования принята следующая модель мультипликативной гребневой функции

$$sm(t)=\sum_{k=0}^{KPS-1} rect\left(\frac{\ln(t)-k\cdot D\cdot T}{T\cdot KW}\right)\frac{\sin\left(2\pi\frac{\ln(t)}{T}\right)}{t}. \qquad (2)$$

На рисунке 4 представлен сигнал. Параметры сигналов определены в обозначениях.

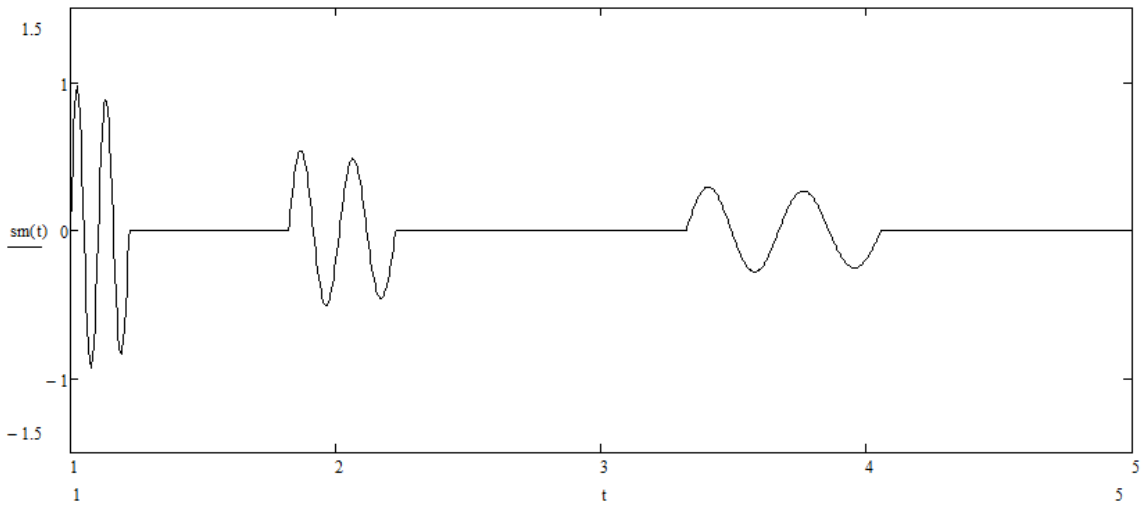

Рис. 4. Элементарные сигналы при численных значениях параметров $T=0{,}01$, $KW=2$, $KPS=2$, $D=6$

Сигнал (2) можно трактовать как гиперболический сигнал, у которого прореживаются волны в гиперболическом масштабе. Для сигнала, представленного на рисунке 4 мультипликативный период составляет 6 волн.

Будем называть мультипликативным периодом функции $s(t)$ положительное вещественное число T, такое, что $s(t \cdot T) = s(t)$, число 1 всегда является периодом. Квадрат периода является периодом, произведение двух периодом снова является периодом и обратное к периоду так же является периодом. Периоды образуют группу периодов функции.

Предположим, что полоса сигнала составляет одну октаву, то есть частотного интервала между двумя частотами, что в нашем случае эквивалентно отношению времени начала сигнала к времени окончания. В предположении, что мы испускаем сигнал в момент времени 1 мы получим, что наш сигнал должен содержаться в интервале от 1 до 2.

Для реализации когерентной обработки пачки необходимо выбрать эталон, который является сигналом ГЧМ с прореживанием элементарных сигналов, например, с требуемым мультипликативным периодом. Выберем в качестве эталона комплексный сигнал

$$s\_comp(t) = \sum_{k=0}^{KPS-1} rect\left(\frac{\ln(t) - k \cdot D \cdot T}{T \cdot KW}\right) \frac{\exp\left(2\pi i \frac{\ln(t)}{T}\right)}{t}.$$

При распространении исходный сигнал изменяется

$$sm(t) = \sum_{k=0}^{KPS-1} rect\left(\frac{\ln(t) - k \cdot D \cdot T}{T \cdot KW}\right) \frac{\sin\left(2\pi \frac{\ln(\alpha(t-\tau))}{T}\right)}{t},$$

где $\tau = 0{,}01$ и $\alpha = 1{,}01$.

Мы будем проводить оценку числовых характеристик корреляционной функции сигналов при количестве отсчетов равном 2048.

Рассмотрим значение параметров $T = 0{,}01$, $KW = 2$, $KPS = 100$, $D = 3$ (рис. 5), при этих значениях отношения сигнал шум будет 41.

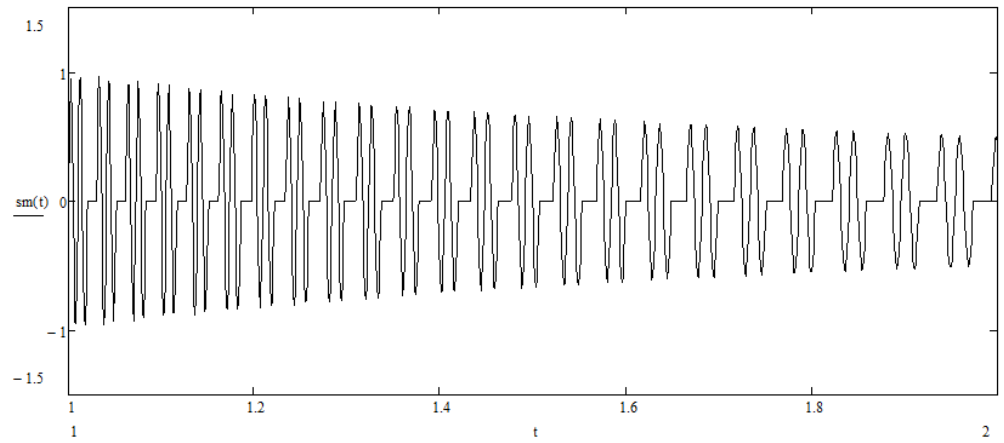

Рис. 5. Пачка импульсов

И определим для него отклик (рис. 6).

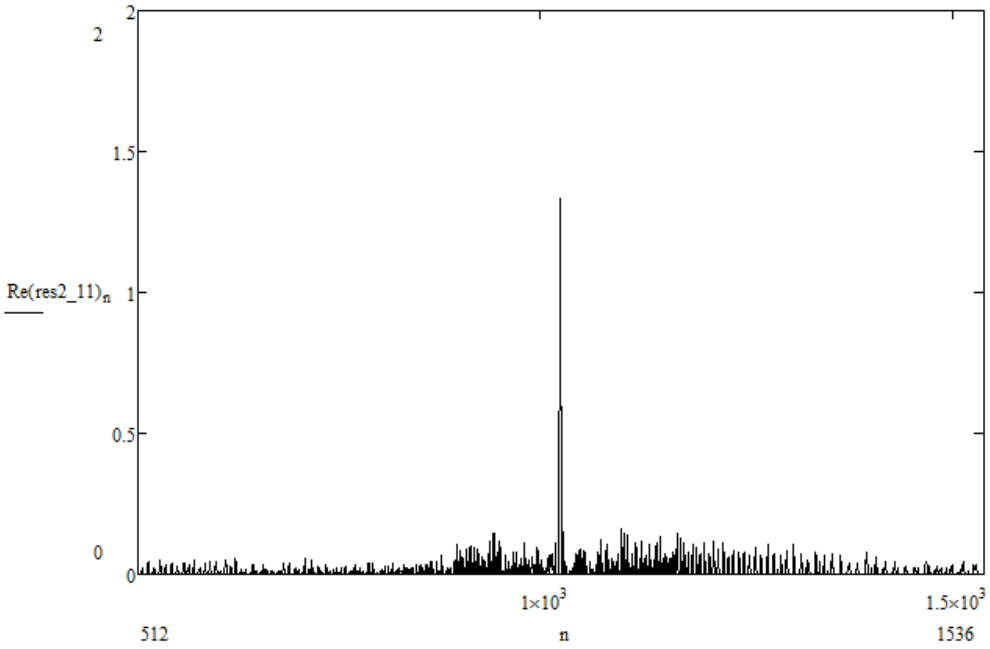

Рис. 6. Отклик согласованного фильтра

Заметим, что мы рассмотрели ГЧМ сигнал и получили для него очень хороший отклик. Теперь немного изменим наш сигнал, приняв $D=4$ и $KPS=17$ (рис. 7), при этих значениях отношения сигнал шум будет 37.

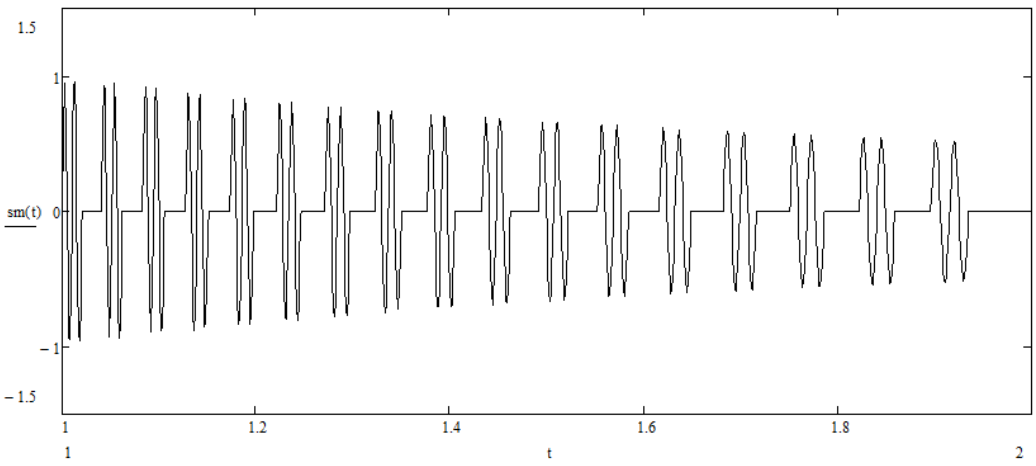

Рис. 7. Пачка импульсов

При этом отклик ухудшится (рис. 8).

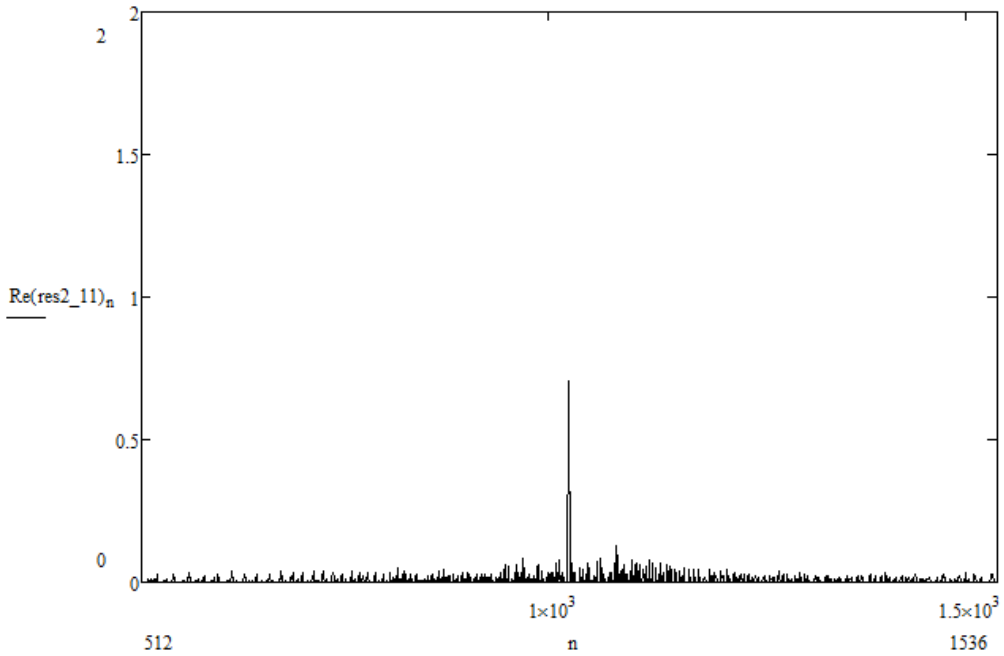

Рис. 8.Отклик согласованного фильтра

Что означает, что при выборе параметров нам нужно стремиться к ГЧМ сигналу. Теперь изменим параметры $KW = 4$, $D = 5$ (рис. 9), при этих значениях отношения сигнал шум будет 59.

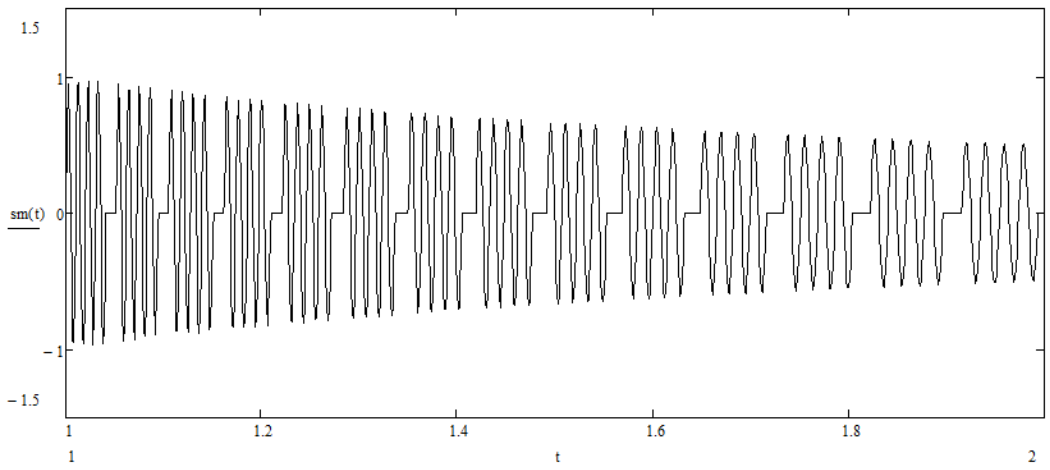

Рис. 9. Пачка импульсов

При этом отклик улучшится (рис. 10).

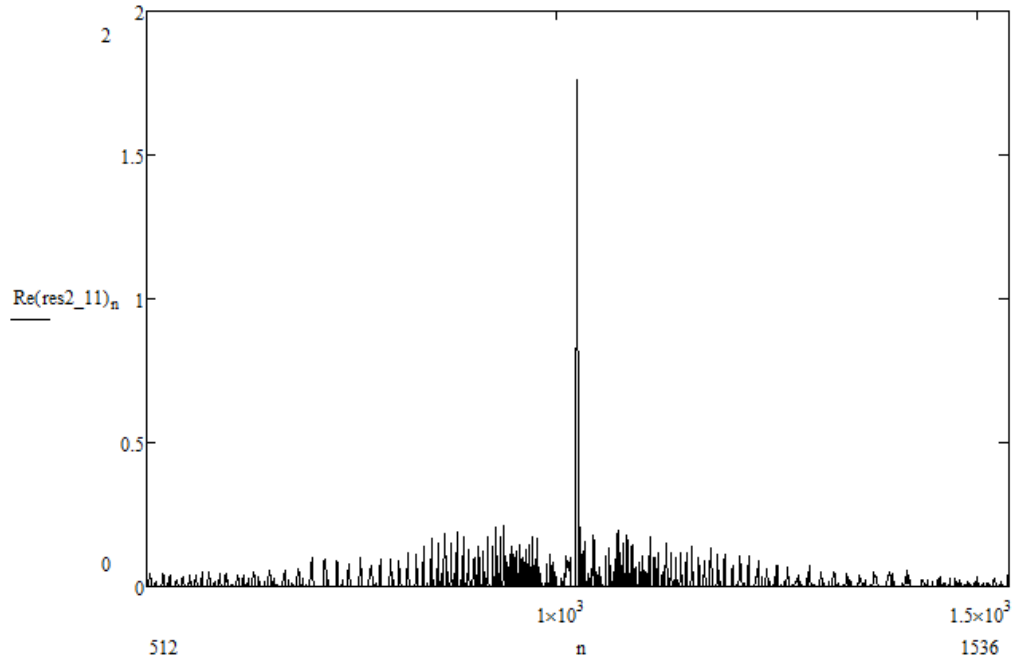

Рис. 10. Отклик согласованного фильтра

Таким образом, можно сделать следующие выводы для сигнала без шума:

1. Для улучшения отклика необходимо приближение мультипликативной гребневой функции к ГЧМ сигналу, для чего величины KW и D должны мало отличаться. Прореживание уменьшает отношение сигнал/шум.

2. Для улучшения отклика лучше увеличивать KW, однако при этом необходимо снижать KPS, что бы сигнал укладывался в одну октаву.

При выполнении этих условий отношение сигнал/шум достигает максимальных значений.

Однако, при введение «белого» шума ситуация меняется (рис. 11).

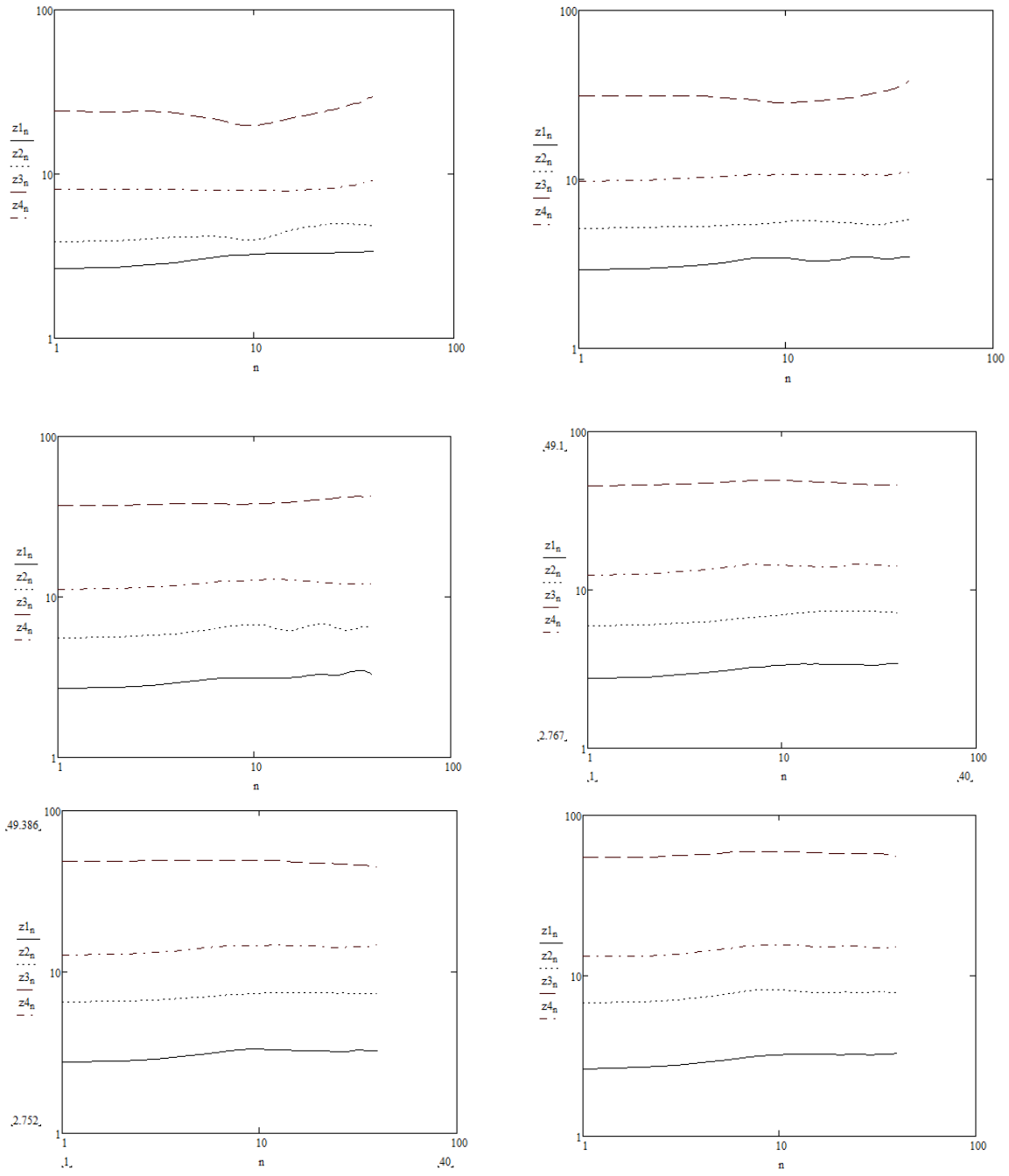

Рис. 11. Зависимости отношения сигнал шум от величины стандартного отклонения «белого» шума (сплошная линия $\sigma = 1$, мелкий пунктир $\sigma = 0{,}1$, штрих пунктир $\sigma = 0{,}05$, крупный пунктир $\sigma = 0{,}01$) и величины сдвига между волнами (мультипликативного периода)

Сравнительный анализ отношения сигнал шум от мультипликативного периода, при условии нормирования энергии сигнала показывает:

1. Отношение сигнал шум меняется мало при изменении мультипликативного периода.
2. Можно использовать различные интервалы прореживания;
3. Нет накопления энергетического вклада обусловленного реверберацией.

Аннотация

Одна из определяющих задач представления акустических сигналов – исследование функционалов при помощи «экспонент», являющихся неприводимыми представлениями группы, которая описывает физические явления, связанные с преобразованием сигнала. Наибольший интерес представляет группа линейных преобразований времени. Данная группа характеризует движение приемника и источника на прямой. Неприводимыми представлениями являются операторы, заданные на полуоси. ГЧМ сигнал является неприводимым относительно этих операторов и поэтому представляет особый интерес.

Аналитическое решение задачи корреляционной оценки имеет высокую трудоемкость. Поэтому ниже будет предложена экспериментальная модель для вычисления корреляционной функции. В ходе эксперимента будет построена модель мультипликативной гребневой функции сначала в аддитивной, а потом в мультипликативной форме.

S. Dotsenko, A. Vladyko

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

M. Gladkikh

CJSC "TELROS"

EXPERIMENTAL EVALUATION OF THE CORRELATION RESPONSE DECIMATED HYPERBOLIC FM SIGNAL

Annotation

One of the key problems of representation of acoustic signals - research of functionals by means of the "exponential curves", they are the irreducible representations of the group, which describes physical phenomena associated with the transformation of the signal. Greatest interest is the group of the linear transformations of time. This group characterizes movement of the source and the receiver on the line. Irreducible representations are the operators defined on semiaxis. Hyperbolic FM signal is irreducible concerning these operators, and therefore of special interest.

Analytical solution of the correlation evaluation is high difficulty. Therefore, in this paper will be offered an experimental model for the calculation of the correlation function. During the experiment will be constructed a model of the multiplicative ridge function, first in the additive form, and then in the multiplicative form.

Key words: hyperbolic FM signal, correlation response, multiplicative ridge function.

Доценко Сергей Михайлович – доктор технических наук, профессор, заслуженный деятель науки РФ, проректор по научной работе ФГОБУ ВПО «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

Владыко Андрей Геннадьевич – кандидат технических наук, начальник Управления научных исследований и подготовки научных кадров ФГОБУ ВПО «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», vladyko@bk.ru

Гладких Михаил Борисович – старший специалист отдела разработки методического и программного обеспечения КЦКБ ЗАО «ТЕЛРОС», gladkikh@telros.ru

УДК 004.896

Л. М. Макаров

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича

СТРУКТУРИРОВАННЫЙ ЛИНГВИСТИЧЕСКИЙ АНАЛИЗ

модель документа, размерность пространства признаков, компьютерный анализ

Введение

Развитие сети телекоммуникаций в значительной мере способствует формированию многочисленных по тематическому содержанию информационных ресурсов, где наибольшее количество представляют текстовые документы. Запрос на поиск, организация поиска интересующего документа, соотнесение полученной подборки документов с той или иной рубрикой осуществляется посредством использования математических моделей. Обращение к математическим аспектам при создании информационных систем и выделение специфических методов формирования решающих правил с учетом языковедения – лингвистических правил, формируется с учетом двух аспектах.

Во-первых, развитие языковедческой теории и практики требует введения все более точных и объективных методов для анализа языка и текста. Одновременно использование математических приемов при систематизации, измерении и обобщении лингвистического материала в сочетании с качественной интерпретацией результатов позволяет языковедам глубже проникнуть в тайны построения языка и образования текста.

Во-вторых, все расширяющиеся контакты языкознания с другими науками, например с акустикой, физиологией высшей нервной деятельности, кибернетикой и вычислительной техникой, могут осуществляться только при использовании математического языка, обладающего высокой степенью общности и универсальности для различных отраслей знаний. Именно математика в этой области исследования становится связующим звеном, позволяющим заглянуть за горизонт предстоящих событий, постоянно формируемый прогрессом в области телекоммуникаций. Особенно актуализируются математические задачи языкознания в связи с использованием естественного языка в информационных и управленческих системах человек–машина–человек.

При обсуждении смежных задач языка и математики следует иметь в виду, что как естественный язык, так и язык математики являются знаковыми (семиотическими) системами передачи информации. Основные

расхождения между этими языками связаны с различным построением языкового знака и знака математического, которые можно выделить в такой последовательности:

1. Лингвистический знак (слово, словосочетание, предложение) обычно включает в себя четыре компонента – *имя* (материальный носитель информации), *денотат* (отражение предмета из внешнего мира), *десигнат* (понятие о предмете) и *коннотат* (комплекс чувственно-оценочных оттенков, связанных с предметом и понятием о нем).

2. Знак математического языка включает только имя и десигнат (математическое понятие).

3. Лингвистический знак многозначен, в то время как математический знак имеет, как правило, одно концептуальное значение.

4. Лингвистический знак потенциально метафоричен, у знака математического метафоричность полностью отсутствует.

Зафиксировать высказанные суждения представляется возможным на иллюстративном примере, сравнив значения математического знака 7 и слова семерка. Если 7 имеет единственное десигнативное математическое значение – «семь любых объектов», то слово семерка имеет несколько значений: «цифра 7», «карта в семь очков», «группа из семи человек» и т. п. При этом в значении слова семерка содержатся не только указанные десигнативные понятия, но оно может указывать на конкретный предмет, например на вполне определенную группу в семь человек. Одновременно это слово несет дополнительные коннотативные метафорические оттенки, связанные с такими словосочетаниями как «великолепная семерка», «семь чудес света», «семь смертных грехов», «семь дочерей Атланта (Плеяды)» и т. д. Из всего сказанного можно создать логическое суждение, которое подчеркивает различие между десигнативными значениями математического и лингвистического знаков.

Значение каждого математического знака легко представить в качестве множества элементов, причем такое множество имеет вполне четкие границы: значение знака 7 является множеством, охватывающим такие конкретные совокупности, которые включают только семь счетных объектов.

Иначе организовано десигнативное значение лингвистического знака – оно также может рассматриваться как множество денотатов, однако это множество не всегда имеет четкие границы. Так, например, не удается определить смысловые границы слов голубой и синий, голубой и зеленый. Разные люди в зависимости от особенностей своего хроматического зрения будут называть показываемые им конкретные сине-голубые оттенки то синим, то голубым цветом. Нельзя также указать точную временную границу, разделяющую значения слов ночь и утро. Иными словами, значения лингвистических знаков представляют собой

нечеткие множества с размытыми границами. Расширяя эти представления на группу слов, позиционирующих произвольный текст, можно сделать заключение о размытости десигнативного образа.

Особенности построения лингвистического языка приводят к тому, что естественный язык представляет собой нежестко организованную диффузную систему, которая воспринимается и используется человеком в значительной мере интуитивно.

Напротив, язык математики является хорошо организованной системой, существующей и функционирующей в виде логического построения, каждый элемент которого имеет осознанную значимость.

Конфронтация естественного языка и языка математики требует, чтобы каждому лингвистическому объекту был поставлен в соответствие некоторый математический объект. Лингвистический знак, например, словосочетание или слово и составляющие этот знак фигуры – фонемы, буквы, слоги – должны интерпретироваться с помощью знаков математических. Эта математическая интерпретация связана с расчленением лингвистического объекта и выделением в нем одного смыслового или сигнального компонента, который становится предметом дальнейшего исследования. Остальные сигнальные и смысловые элементы лингвистического объекта, а также разного рода метафорические коннотативные оттенки из рассмотрения исключаются.

Математическая экспликация

Применение математических методов в языкознании имеет своей целью заменить обычно диффузную, интуитивно сформулированную и не имеющую полного решения лингвистическую задачу одной или несколькими более простыми, логически сформулированными и имеющими алгоритмическое решение математическими задачами. Такое расчленение сложной лингвистической проблемы на более простые задачи относят к области математической экспликации лингвистического объекта или явления.

Математическая экспликация интересна не только с чисто познавательной и теоретической точки зрения. Она совершенно необходима при решении прикладных вопросов, связанных с анализом и синтезом устной речи или информационной переработкой текста на компьютере. Математическая экспликация лингвистических объектов оказывается чрезвычайно полезной не только при решении типовых задач на компьютере, но также при составлении и реализации эвристических алгоритмов искусственного интеллекта.

Надо признать, что выбор математического аппарата в лингвистических исследованиях является сложной задачей, требующей привлечения большого количества позитивных аргументов в пользу того или иного привлекаемого для исследования метода. В этом вопросе

важно составить представление как определяется объект исследования, основные понятия языкознания и его теоретического ядра в контексте структурной математической лингвистики.

В этом направлении создания рекомендаций по выбору метода исследования существует много мнений. Наиболее распространено представление о том, что предметом математической и структурной лингвистики должно быть изучение грамматики, порождающей текст. При этом грамматика понимается как конечное множество детерминированных правил, в том числе неграмматических, а язык рассматривается как бесконечное число регулярных цепочек слов, порождаемых этой грамматикой. В этой концепции экспликация лингвистических объектов рассматривается с позиции математики, где выделяются разделы: теории множеств, математической логики, теории рекурсивных функций, бинарных отношений, теории алгоритмов.

Развитие этих представлений формирует дополнительные разделы, аргументация рассмотрения которых формируется на представлениях количественного исследования текстового материала. В качестве таких разделов рассматриваются: математическая статистика, теория вероятностей, теория информации, математический анализ. Сочетание направлений комбинаторной лингвистики и квантитативной (количественной) лингвистики формирует общий вектор исследований в этой области знаний.

Для того чтобы провести адекватный выбор методов исследования следует общую процедуру образования текста. Порождение текста определяется, с одной стороны, системой языка и ограничивающей ее действие нормой, а, с другой – совершенно независимой от языка внешней ситуацией, иначе говоря, условиями внешней среды в определенной мере влияющей на семантическую составляющую текста.

Рассматривая язык как неколичественную систему, комбинаторная лингвистика пытается описать механизм перехода от языка к речи с помощью тех же приемов «неколичественной» математики. Такие описания представляют собой контекстно-свободные грамматики, т. е. грамматики, не учитывающие контекстных ограничений на употребление отдельных лингвистических единиц и их сочетаний. В связи с этим контекстно-свободные грамматики порождают много цепочек, не являющихся реальными предложениями данного языка. Чтобы добиться порождения реальных текстов, необходимо перейти от контекстно-свободных грамматик к более сильным контекстно-зависимым грамматикам. Такие грамматики можно построить при условии, что к элементам системы языка применяются вероятностные оценки, а сам язык рассматривается как неколичественная производящая система, функционирование которой регулируется вероятностными ограничениями, заложенными в норме.

Эти суждения следует расширить при рассмотрении текстовых документов и речевых актов. Действительно, при рассмотрении текста или речевой записи текста имеем линейную цепочку отграниченных друг от друга (дискретных) символов, (фонем, букв, слогов, слов). Каждый из символов встречается в тексте с определенной частотой и обладает особыми валентностями, т. е. лингвистическими способностями сочетаться с другими символами. Эти свойства лингвистических единиц в тексте эксплицируются в терминах теории вероятностей и математической статистики. К результатам вероятностно-статистического описания, взятым в сочетании с данными лингво-психологического эксперимента, может быть применен аппарат теории информации, с помощью которого удается количественно оценить как структурную организацию текста, так и заключенною в нем смысловую информацию. Из всего сказанного следует, что математическая экспликация центральной проблемы современного языкознания «система языка – норма – текст» может быть осуществлена на основе применения методов как «качественной», так и «количественной» математики.

Известно, что для решения хорошо структурированных количественно выражаемых проблем используется известная методология исследования операций, которая состоит в построении адекватной математической модели. Например, задачи линейного, нелинейного, динамического программирования, задачи теории массового обслуживания, теории игр и применении методов для отыскания оптимальной стратегии управления целенаправленными действиями.

Напротив, для решения слабо структурированных проблем используется методология системного анализа, системы поддержки принятия решений (СППР). Рассмотрим технологию применения системного анализа к решению сложных задач. В этом направлении большое внимание уделяется построению моделей для обоснования решений. Формирование моделей принятия решений происходит с учетом накопленного опыта в этой области знаний, а также требованиями адекватного сопряжения результатов с эксплицитными суждениями.

Выделяя аспект о наличии эксплицитных суждений, формируемых человеком, например, при оценке результатов компьютерного анализа текстового массива, следует указать на наличие большого количества концепций, рассматривающих это уникальное проявление деятельности человека. Именно большое разнообразие взглядов и выделение реперных точек запуска исследовательской процедуры постоянно стимулирует разработку новых алгоритмов и правил формирования адекватных суждений на основе формализованных компьютерных конструкций.

Современные информационные ресурсы представлены большими массивами текстов. Смысловая часть текстового документа представлена словами, которые сформированы из букв. Важными характеристика-

ми текста являются повторяемость букв, пар букв (биграмм) и вообще m-ок (m-грамм), сочетаемость букв друг с другом, чередование гласных и согласных и некоторые другие. В первом приближении можно сказать, что эти характеристики являются достаточно устойчивыми. Используя методы статистики можно вычислить частоты вхождения в текст отдельных слов.

Текст можно позиционировать в качестве временного ряда значений. Для этого производят исполнение серии процедур по воспроизведению матричной структуры текста, где в качестве элементов рассматриваются отдельные элементы, как правило, слова или элементы слов (морфемы). Аналогичный подход используется при рассмотрении графического материала, в качестве которого рассматриваются графики, образы сигналов, фото и виде фреймы. Исследование временных рядов представляет большую серию разнообразных практических задач, среди которых выделяется проблема оценки структурных свойств информационных массивов, например, представленных в формате текста или графики. Так, в частности, расширяя эти понятия можно указать на задачи анализа сигналов, позиционируемых временным рядом амплитудных значений.

Известно, что любой временной ряд может позиционироваться фракталом. Все эти ряды обычно порождаются сложными нелинейными процессами, описание которых в виде дифференциальных уравнений или дискретных отображений обычно связано с большими трудностями. Отмечается что, несмотря на крайнюю нерегулярность сигнальных структур, характер их поведения остается неизменным на всех масштабах, вплоть до минимального структурного образа. Другими словами, с точностью до масштабного фактора, структурный образ сигнала, позиционируемого временным рядом значений, на разных масштабах выглядит примерно одинаково.

В образном выражении это характеризуется «вложением» множества образов в один, который рассматривается как фрактальный. Такое широкое распространение фрактальной структуры, по-видимому, связано с тем, что в реальности любые достаточно нерегулярные процессы стремятся обрести самоподобную структуру. Поэтому совсем не удивительно, что нелинейные динамические системы, в качестве которых могут рассматриваться отдельные фрагменты текста и текст в целом, обычно имеют фрактальные аттракторы.

Подобие текстовых структур

Основной характеристикой самоподобных структур, как известно, является размерность D, введенная Хаусдорфом в 1919 году для компактного множества в произвольном метрическом пространстве [1]. Такая оценка оказывается полезной при сопоставлении разных текстовых

документов. Если в качестве аппроксимации графиков временных рядов рассматривать комплексы, состоящие из двумерных выпуклых фигур, с геометрическим фактором δ, то, как следует из определения Хаусдорфа, фрактальная размерность D интерпретируется в соответствии с выражение:

$$S(\delta) \sim \delta^{2-D} \text{ при } \delta \to 0, \quad (1)$$

где $S(\delta)$ – характеризует полную площадь комплекса, с масштабом разбиения δ. Между тем, на практике, при попытке вычислить D непосредственно из формулы (1), возникает серьезная проблема. Она связана с тем, что с одной стороны, реальные временные ряды всегда имеют минимальным масштаб структуры δ_0, с другой же стороны, для всех известных аппроксимаций приближение к асимптотическому режиму (1) обычно является слишком медленным.

По этой причине, для определения D обычно вычисляют показатель Херста H, который для гауссовых процессов связан с D соотношением $H = 2\text{-}D$. Однако, для надежного вычисления H требуется слишком большой репрезентативный масштаб, содержащий несколько тысяч данных. Внутри этого масштаба временной ряд много раз меняет характер своего поведения. Чтобы связать локальную динамику соответствующего процесса с фрактальной размерностью временного ряда необходимо определить размерность D локально. Для этого необходимо найти процедуру реализации аппроксимации, которая реализовала быстрое вычисление искомой функции $S(\delta)$ по короткому ряду исходных значений.

По Д. Хаусдорфу определение размерности для компактного множества в произвольном метрическом пространстве производится по выражению:

$$N(\delta) \sim (\delta)^{-D}, \quad (2)$$

где параметр δ соотносится с минимальным размером элементарного элемента множества.

При рассмотрении текстовых или графических материалов в качестве элементарного элемента принимается: элемент алфавита (буква) морфема – для текста и пиксель – графического документа. Следуя этим представлениям, размерность фрактального пространства позиционирования текста или графического документа определяется по выражению:

$$D = \lim_{\delta \to 0} \frac{\ln N(\delta)}{\ln(\delta)}. \quad (3)$$

Показатель D называют предельной емкостью пространства позиционирования, понимая при этом под размерностью Хаусдорфа d_H критическое значение аргумента функции $m(p) = \sup\limits_{\varepsilon>0} \inf\limits_{\{A_i^\varepsilon\}} \sum\limits_i (diam A_i^\varepsilon)^p$, обладающее тем свойством, что при $p < d_H$ $m(p) = \infty$, а при $p > d_H$ $m(p) = 0$. Здесь A_i^ε – характеризует покрытие исходного множества семейством множеств A_i^ε, имеющих размер меньше установленного значения ε.

Для вычисления размерности модельных конструкций фракталов, рассчитываемой воспользуемся данными о количестве словоформ и статистически вычисляемой оценкой встречаемости минимальной конструкции словоформы на избранном фрагменте текста. Тогда в качестве фрактальной оценки следует использовать значение, вычисляемое по выражению (3). Данное значение, представленное в серии вычислений, по всем фрагментам исследуемого текстового материала позволяет воспроизвести фрактальный портрет, который является основой сопоставления различных текстов, с целью уточнения их близости.

Другая оценка, отражающая сходство (близость) текстовых материалов, формируется также на основе статистического материала и постулирует необходимость использования генетического алгоритма (ГА).

Формально полагаем, что имеется информационное множество элементов, представленных текстовым документом A, где выделен набор фонем, расположенных построчно. Длина строки фиксирована.

Тогда на поле информационного массива можно выделить массивы: $M = \{1, 2, ...m]$ и массив $N = \{1, 2, ...n]$, такие, что $m, n \in N$ позиционируют два конечных множества.

Определим в качестве матрицы массив размера mxn, элементами из некоторого кольца или поля K, отображаемого в виде:

$$A : MxN \to K.$$

Если индекс i пробегает множество M, а j пробегает множество N, то элемент $A(i, j)$ оказывается элементом матрицы, находящемся на пересечении i-й строки и j-го столбца:

– i-я строка матрицы состоит из элементов вида $A(i, j)$, где j пробегает все множество N;

– j-й столбец матрицы состоит из элементов вида $A(i, j)$, где i пробегает все множество M.

В соответствии с этим имеем:

– каждая строка матрицы интерпретируется как вектор в n-мерном координатном пространстве K^n;

– каждый столбец матрицы интерпретируется как вектор в m-мерном координатном пространстве K^m.

Полная конструкция матрицы естественным образом интерпретируется как вектор в пространстве K^{nm} имеющим размерность $m \times n$.

Так, например, для текстового массива документа A можно указать процедуру формального позиционирования, представленную в виде:

$$\begin{matrix} a_{11,} \ldots\ldots\ldots\ldots\ldots, a_{1n} \\ a_{21,} \ldots\ldots\ldots\ldots\ldots, a_{2n} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1,} \ldots\ldots\ldots\ldots\ldots, a_{mn} \end{matrix} \qquad A = \left[a_{mn} \right]$$

Применение этой процедуры к массиву текста позиционирует каждый элемент текста в элемент матрицы, закрепляя этот элемент в строго определенной последовательности развертки событий, осуществляемых в модели преобразования. Основой для аналитической процедуры избирается ГА, реализуемый поэтапно:

1. Исходный текстовый массив преобразуется в набор словоформ.
2. Набор словоформ позиционируется в виде матрицы A, с установленным размером строки (количеством элементов в строке).
3. По массиву элементов матрицы A формируется показатель – количество слов.
4. По массиву элементов матрицы A формируется показатель – количество знаков.
5. Массив матрицы A, с отображением словоформ, заменяется бинарным транспарантом – бинарным массивом данных A', по выражению:

$$A = \left[a_{m,n} \right] \Rightarrow A'[a'_{m,n}].$$

Далее по массиву бинарных данных, зафиксированных в структуре матрицы, реализуется ГА. Известно, что реализация ГА требует начальной введения начальной модели инициализации событий. В основе такой модели полагается, что при наличии двух источников событий представляется возможным произвести отдаленное событие, эволюционно связанное с первичными источниками. Такая процедура формирования будущего события на основе двух базовых событий реализуется в серии синтеза промежуточных событий, рассматриваемых в качестве потомков. Принимая за основу такую схему реализации ГА, установим:

1. В качестве начальных источников событий – первые две строки текстового массива, представленные в векторной форме $A^{1,m}$, $A^{2,m}$.

2. Формирование 1 и 2 потомка осуществляется по процедуре:

$$A^{A1} = \left| A^{1,m} - A^{2,m} \right|, \qquad A^{A2} = \left| A^{A1} - A^{3,m} \right|.$$

3. Формирование *i*-го потомка осуществляется по процедуре:

$$A^{Ai} = \left| A^{Ai} - A^{i+1,m} \right|.$$

Иллюстративные примеры использования ГА представлены на рисунках 1 и 2.

Рис. 1. Образ рассказа А. П. Чехова

Рис. 2. Образ рассказа А. Азимова

Использование метода ГА позволяет достаточно быстро воспроизвести индивидуальный образ исследуемого документа и провести сравнение по массиву документов.

Меру близости информационных массивов – текстовых документов можно вычислить статистически с использованием алгоритмов интеллектуального анализа данных (Data Mining). Следует отметить, что выбор наилучшей меры является сложной задачей и определяется целым

рядом обстоятельств. Во-первых, существует большое разнообразие подходов к расчету мер, что само по себе затрудняет выбор удачной формулы или алгоритма. Во-вторых, язык, а точнее, структура грамматического строя языков представляет собой чрезвычайно сложный информационный объект. Это нашло свое отражение в структуре модели описания языка в базе данных. В-третьих, большой объем данных в базы данных (БД), накладывает существенные ограничения на прямые эксперименты по выбору мер близости. В-четвертых, надо хорошо понимать, что численная характеристика близости, рассчитанная по выбранной формуле, отражает не только следы родства, но и следы заимствований в языке. Таким образом, проблема устранения типологического фона становится чуть ли не главной при выборе наилучшей меры близости.

Именно поэтому выбор, или, возможно, конструирование новой меры близости, которая удачно передаст генетическое родство текстов, является непростой экспертной задачей, в которой проявляются не только знания, но и интуиция исследователя.

Выделяя конечную фазу исследования, обратимся к рассмотрению критерия качества сопоставления текстовых документов.

Подходы к расчету мер близости

В самом общем понимании этой задачи можно указать, что меры близости объектов:

$$X = \{x_1, x_2, ..., x_n\} \quad \text{и} \quad Y = \{y_1, y_2, ..., y_n\}$$

позиционируемых дихотомическими признаками x_i и y_i, могут быть выражены в виде функции четырех переменных [2, 3]:

$$S = F(a, b, c, d),$$

где a – число признаков, отсутствующих у X и Y одновременно, d – число совпадающих признаков, b (или c) – число признаков, присутствующих у X, но отсутствующих у Y (или наоборот). Сумма $(a + b + c + d)$ равна общему числу признаков в дихотомическом наборе размерности n. Выделим следующие критерии качества сопоставления документов, представленные в виде набора выражений [3]:

$$S_1 = \frac{a+d}{a+b+c+d}; \tag{4.1}$$

$$S_2 = \frac{2(a+d)}{2(a+b)+c+d}; \tag{4.2}$$

$$S_3 = \frac{(a+d)}{a+b+2(c+d)}; \tag{4.3}$$

$$S_4 = \frac{d}{a+b+c+d}; \quad (4.4)$$

$$S_5 = \frac{d}{b+c+d}; \quad (4.5)$$

$$S_6 = \frac{a}{a+b+c}. \quad (4.6)$$

Меры близости S_1-S_6 могут быть представлены в виде функций от $\upsilon(X)$, $\upsilon(Y)$, $\upsilon(X.\cap Y)$ или $\upsilon(X \cup Y)$. Это можно объяснить тем, что близость объектов X и Y определяется наличием у них общих признаков, информационная ценность которых ($\upsilon(X. \cap Y)$) должна быть неким образом соотнесена с информацией об объектах, рассматриваемых порознь ($\upsilon(X)$, $\upsilon(Y)$) или в совокупности ($\upsilon(X. \cup Y)$). Сходство объектов может быть измерено не только по наличию, но и по отсутствию у них одних и тех же признаков. Можно сконструировать общее выражение для меры близости, в виде:

$$S_7 = \frac{R_1 a + R_4 d}{R_1 a + R_2 b + R_3 c + R_4 d}, \quad (4.7)$$

где $R_1 \ldots R_4$ – коэффициенты, определяющие вклад в общую меру одновременного присутствия признаков (параметр a), одновременного отсутствия признаков (d) или противоречий (b и c) у сравниваемых объектов.

Формулы (4.1–4.6) описывают сравнительно простые меры близости, построенные по аддитивному принципу. Коэффициенты, используемые в формулах (4.2), (4.3) и (4.6), структурируют признаковое пространство модели по принципу наличия/отсутствия и совпадения/несовпадения признаков у объектов X и Y. Эти выражения можно усложнить, если каждому признаку поставить в соответствие вес w, который в свою очередь может являться функцией от частоты f или уровня L в иерархии признаков.

$$a = \sum_{i=1}^{n} w_i p_i, \quad (4.8)$$

где $w_i^f = W(f)$ если используется весовая частотная мера, или $w_i^L = W(f)$ если используется весовая мера иерархическая.

Информация о частоте того ли иного признака или его уровне в иерархии в БД в явном виде не представлена, однако без труда может быть эксплицирована. Частота f для признака i задается как отношение

$$f = \frac{N}{N_p}, \quad (4.9)$$

где N – количество элементов анализируемого массива, в которых встречается признак i; N_p – общее количество элементов.

Использование такой континуальной переменной, как частота признака и веса, построенного на ее основе, расширяет число вариантов мер близости до бесконечности. Вариативность параметрического пространства определяется тремя возможными способами задания веса, а также их комбинациями.

Заключение

Структурированный анализ текстовых документов, осуществляемый с использованием математических моделей позиционирования элементов текста, позволяет выявлять различия, которые следует относить к авторскому стилю изложения материала. Развитие этих представлений, сформулированное на представительных выборках реального макета документа можно рассматривать как самостоятельную отдельную задачу автоматического сопоставления близких по тематическому содержанию документов, но принадлежащих разным авторам.

Библиографический список

1. **Семиотика** и лингвистика / З. Д. Попова. – М., 2005.
2. **Логика** и логическая семантика / Г. Фреге. – М., 2000.
3. **Введение** в лингвистическую компаративистику / С. А. Бурлак, С. А. Старостин. – М. : УРСС, 2001.

Аннотация

В статье излагаются принципы структурированного анализа текстовых документов с использованием математических моделей позиционирования элементов текста. Представлена отдельная задача автоматического сопоставления близких по тематическому содержанию документов, но принадлежащих разным авторам.

L. Makarov

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

STRUCTURED LINGUISTIC ANALYSIS

Annotation

This article states about principles of structured analysis of text documents using mathematical models positioning elements of the text. Represented by a separate task of automatic mapping of similar thematic content of the documents with various authors.

Key words: document model, the dimension of the feature space, a computer analysis.

References

1. **Semiotika** i lingvistika / Z. D. Popova. – M., 2005.
2. **Logika** i logicheskaja semantika / G. Frege. – M., 2000.
3. **Vvedenie** v lingvisticheskuju komparativistiku / S. A. Burlak, S. A. Starostin. – M. : URSS, 2001.

Макаров Леонид Михайлович – кандидат технических наук, доцент, профессор кафедры «Автоматизация предприятий связи» ФГОБУ ВПО «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

ISSN 2307-1303
ЭЛЕКТРОННЫЙ НАУЧНЫЙ ЖУРНАЛ СПбГУТ
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ

ТРЕБОВАНИЯ К РУКОПИСЯМ НАУЧНЫХ СТАТЕЙ, ПОДАВАЕМЫМ ДЛЯ ПУБЛИКАЦИИ

ОБЩИЕ ТРЕБОВАНИЯ И УСЛОВИЯ

1.1 Редакция принимает материалы (рукописи научных статей на русском или английском языках и сопроводительные документы), в которых приводятся результаты научной деятельности автора(ов).
1.2 Прием материалов осуществляется лично от авторов (наб. р. Мойки 61, каб. 269)**, через сайт журнала www.itt.sut.ru**.
1.3 В журнале не публикуются ранее опубликованные результаты научной деятельности автора(ов).
1.4 Подаваемые для публикации рукописи должны быть законченными работами, однако могут быть продолжающимися (разбитые на части и имеющие выводы по частям).
1.5 Результаты исследований должны соответствовать одной из научной отрасли: физико-математической (01.02.00, 01.04.00); технической (05.12.00, 05.13.00, 05.27.00), экономической (08.00.05 - экономика и управление в сфере IT и телекоммуникаций).
1.6 Материалы представляются в электронном виде (архив rar/zip) в следующем комплекте:

Файл 1 – рукопись научной статьи в формате MS Word (.doc). Если в рукописи присутствуют рисунки, необходимо приложить их в исходных форматах.
Файл 2 – скан первой страницы рукописи, подписанной автором(ами) - не предоставляется при личной подаче.
Файл 3 – информация об авторах, включая персональные данные и согласие на их обработку (фамилия, имя отчество полностью, дата рождения, место работы, ученая степень и звание, должность, паспортные сведения (серия, номер, дата выдачи, кем выдан), контактный е-мейл, телефон) в формате MS Word (.doc).
Файл 4 – информация об авторах (без персональных данных) на английском языке в формате MS Word (.doc).
Файл 5 – экспертное заключение о возможности опубликования рукописи в открытом доступе, заверенное по месту обучения или работы в сканированном виде. Авторам следует предусмотреть наличие письменного согласия на публикацию от лиц или организаций, по заказу которых производились те или иные исследования, прямые или косвенные результаты которых планируются к публикации в журнале. Отказ в представлении такого согласия может послужить отказом в публикации рукописи.
Файл 6 (для студентов-исследователей, магистров, аспирантов и соискателей ученых степеней) – рецензия научного руководителя, подпись научного руководителя заверяется по месту его работы в сканированном виде.
Рецензия должна содержать:
- актуальность рассматриваемых результатов;
- научную новизну предполагаемых решений;
- критический обзор статьи (включая замечания и предложения по их устранению);
- определение возможности публикации статьи в журнале.
- подпись и дату рецензирования, заверение подписи.

1.7 После получения материалов в полном объеме (п. 1.6) автору отправляется для заключения лицензионный договор о передаче неисключительных прав на использование произведения (может быть в виде оферты) и квитанция на оплату организационного взноса.
Редакция оставляет за собой право отбора рукописей для журнала и затребовать дополнительные документы.
1.8 Рукописи и носители информации авторам не возвращаются, гонорар не выплачивается.
1.9 Все рукописи проходят внешнюю экспертную оценку – рецензирование. При отрицательном заключении рукопись отклоняется для публикации.
1.10 При наличии положительной рецензии с рекомендацией рукописи к публикации или рекомендацией доработать рукопись (устранить замечания) и оплаченного организационного взноса, рукопись передается на предпечатную подготовку. При передаче на предпечатную подготовку рукописи, устанавливается срок ее выхода из печати. При отрицательной повторной рецензии рукопись отклоняется для публикации.
1.11 В процессе предпечатной подготовки рукопись вычитывается редактором/корректором. В это время редактор/корректор будет обращаться к автору(ам) для исключения неточностей, ошибок, опечаток и пр. по указанным автором(амии) е-мэйлу/телефону.
1.12 После второй корректуры рукопись ставится в выпуск, подписывается в печать. Внесение авторских исправлений после подписания рукописи/выпуска в печать невозможно.

ТРЕБОВАНИЯ К ФАЙЛУ РУКОПИСИ

2.1 Объем текста научной статьи – не менее 8 и не более 12-ти машинописных страниц (с рисунками, таблицами, библиографией). Формат страницы – А4, при этом каждое поле должно быть 25 мм, за исключением левого - 30 мм. Абзацный отступ 10 мм.
2.2 На первой странице рукописи до текста указываются УДК, фамилии, инициалы авторов, название статьи, ключевые слова.
2.3 Структура рукописи следующая:
УДК (размер шрифта 14, расположение текста по левому краю); для определения УДК используйте on-line классификатор www.udcc.org;
фамилия, инициалы автора (авторов) (с расстановкой по алфавиту) с указанием места обучения или работы (размер шрифта 14, расположение текста по левому краю);
название рукописи (размер шрифта 14, заглавные буквы, расположение текста по левому краю);
ключевые слова (размер шрифта 12, расположение текста по ширине);
текст рукописи (размер шрифта 14, расположение текста по ширине):
- введение;
- пункты и подпункты;
- заключение (выводы);
- библиографический список (размер шрифта 14, расположение текста по ширине);
- аннотация (размер шрифта 14, расположение текста по ширине) приводится на отдельной странице после библиографического списка.

На английском языке (на русском, если язык рукописи английский) представляются: фамилия и имя автора, место работы или учебы, название статьи, аннотация (абстракт), ключевые слова, библиографический список, сведения об авторе(ах).
При написании аннотации рекомендуем использовать статью «Сысоев, П. В. Правила написания аннотации / П. В. Сысоев // Иностранные языки в школе / гл. ред. Н. П. Каменецкая.– 2009.– N4.– С.81-83».
2.4 Текст рукописи должен быть тщательно вычитан и подписан всеми авторами на первой странице, правка текста и исправление рисунков, корректировка аннотации выполняются редактором журнала совместно с автором(ами).
2.5 Верстку производить с межстрочным интервалом «1» по приведенным требованиям, стили и макросы не применять.
2.6 Буквы в тексте и формулах латинского алфавита набираются курсивом, буквы греческого и русского алфавитов – прямым шрифтом. Математические символы lim, lg, ln, arg, sin, min и т. д. набираются прямым шрифтом.
2.7 Не следует применять сходные по начертанию буквы латинского, греческого и русского алфавитов, использовать собственные макросы и рисунки для букв.
2.8 Следует различать букву О и ноль 0; дефис «-», знак минуса и тире «–».
2.9 Формулы должны быть набраны только в редакторе MS Equation, а отдельные символы и буквы формул в тексте статьи должны быть набраны в редакторе MS Word (не в Equation!). Длинные формулы следует разбивать на независимые фрагменты (каждая строка – отдельный объект). Нумеровать нужно только те формулы, на которые есть ссылки в тексте.
2.10 Нельзя использовать рисунки и таблицы для размещения формул.
2.11 Рисунки и фотографии располагаются в тексте.
2.12 Ширина таблиц (шрифт 12 pt) не должна превышать ширину страницы.
2.13 Все таблицы, графики, схемы и рисунки должны быть подписаны и обязательно оформлены с переводом в формат MS Word.
2.14 На рисунках буквы латинского алфавита в сканированном виде также набираются курсивом, а буквы греческого и русского алфавитов – прямым шрифтом.
2.15 Перечень источников приводится общим списком в конце статьи. Составляется в соответствии с последовательностью ссылок в тексте. Ссылки на источники в тексте приводятся в квадратных скобках.
2.16 Примеры оформления библиографических описаний различных изданий приведены в ГОСТ Р 7.0.5 2008 «Библиографическая ссылка. Общие требования и правила составления».

НАУЧНОЕ ИЗДАНИЕ

Информационные технологии и телекоммуникации

Электронный научный журнал

Выпуск 2-2013

Минимальные системные требования для просмотра издания:
Тип компьютера, процессор, сопроцессор, частота: Pentium IV и выше / аналогичное; оперативная память (RAM): 256 Мб и выше; необходимо на винчестере: не менее 64 Мб; ОС MacOS, Windows (XP, Vista, 7) / аналогичное; видеосистема: встроенная; дополнительное ПО: Adobe Reader версия от 7.X или аналогичное. Защита от незаконного распространения: реализуется встроенными средствами Adobe Acrobat.

Идея создания журнала состоит в полноформатном информировании сообщества о тенденциях развития IT и телекоммуникационной отраслей, в сочетании новейших достижений науки и их внедрения в производство. Наш журнал это не только окно информации, но и площадка для научных дискуссий. Мы готовы предоставить площадку как для ученых с именем, так и для только делающих первые шаги в науке. Журнал ориентирован на статьи в области физико-математических; технических (05.12.00, 05.13.00, 05.27.00), исторических (07.00.10), экономических (08.00.05 - управление инновациями, связь, экономические аспекты IT и телекоммуникаций) наук.

Подписано в печать 26.08.2013
Вышло в свет 30.08.2013
Уст. объем 4,375 печ. л. Заказ № 003-ИТТ-2013.
191186, СПб, наб. р. Мойки, 61
E-mail: telecomsut@gmail.com
www.itt.sut.ru