ISSN 2307-1303

ЭЛЕКТРОННЫЙ НАУЧНЫЙ ЖУРНАЛ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ

ВЫПУСК 3

Информационные технологии и телекоммуникации

электронный научный журнал, выпуск 3 за 2013 год

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

UDK 004.72; 004.715

Tiamiyu A. Osuolale

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

TRUSTED ROUTING VS. MPLS: DATA SECURITY, QoS AND NETWORK SCALABILITY

MPLS, trusted routing, data security, QoS, data integrity, network scalability

1. Introduction

1.1. MPLS

MPLS is a high-performance telecommunication network [1] that allows data transfer within a network using labels, which are the information attached to the packet that tells every intermediate router (label-switch router – LSR), to which egress edge label-switched router (E-LSR) it must be forwarded. Being used as a backbone for VPN, MPLS' popularity is growing. It is gradually replacing frame relay service because of its improved connectivity and QoS.

MPLS is not a routing protocol but a flexible mechanism that incorporates concepts and protocols to achieve functions that improve the current Layer 3 and Layer 2 technologies [1–8]. MPLS brings the label switching and traffic engineering functions of ATM to packet-based networks; but unlike ATM, MPLS runs over any Layer 2 infrastructure. MPLS is IP-compatible, thus it integrates easily with traditional IP networks. In MPLS, packets are routed along pre-configured Label Switched Paths (LSPs), and packet flows are connection-oriented [2]. In MPLS, QoS implementation mechanism enables the creation of LSPs with guaranteed bandwidth. Figure 1 shows an MPLS domain.

1.2. Trusted Routing (TR)

TR concept refers to the process of planning the transfer of information flow on the calculated route through telecommunication network nodes, excluding the possibility of substitution, modification or inclusion of any form of information into the data stream passing through those nodes [6].

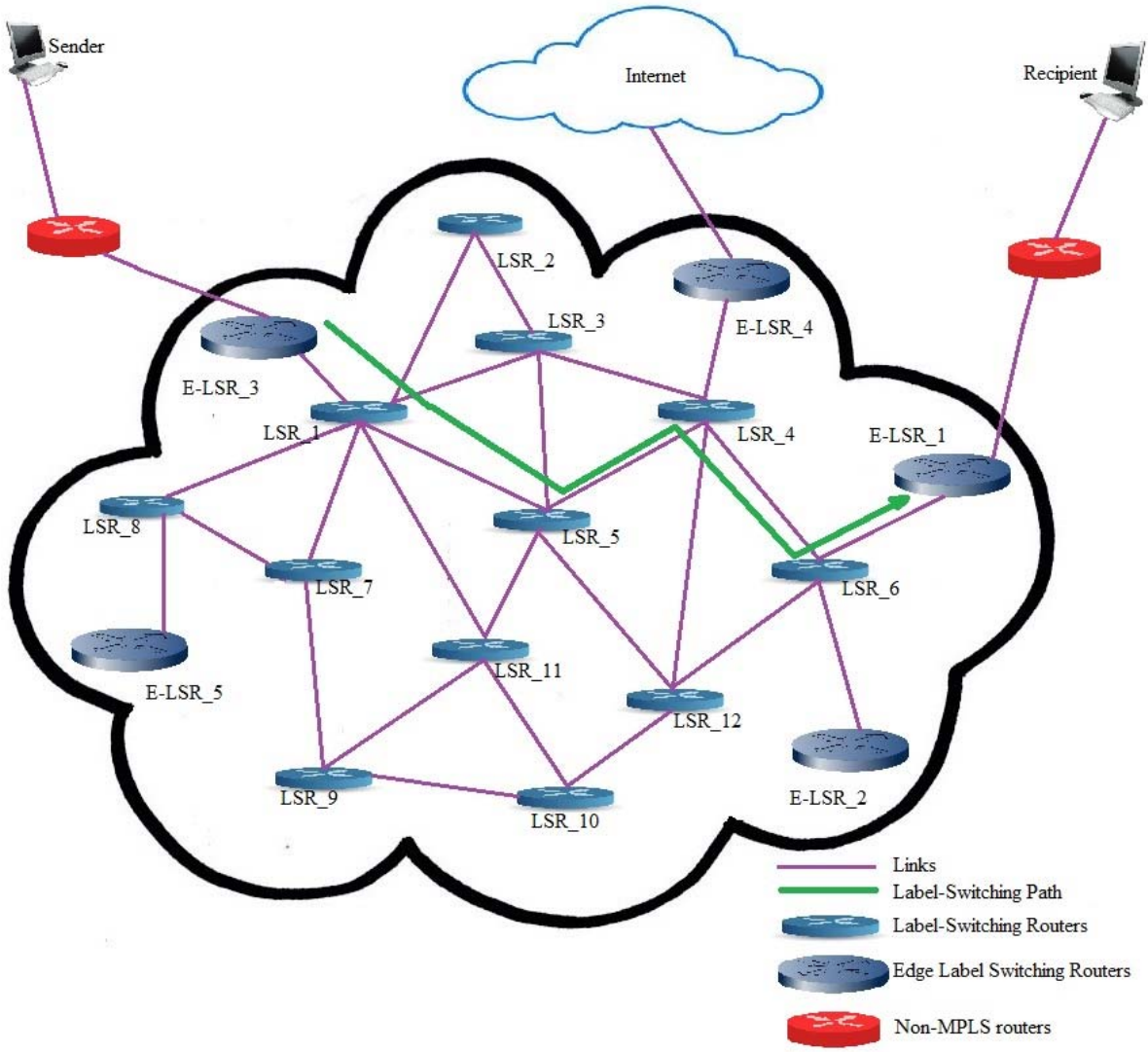

Figure 1. An MPLS Domain

TR consists of the following stages:

1. Defining network topologies and identify trusted routes.

2. Preparing trusted routes, and-"zombifying" (i. e. make trusted by gaining full control on) untrusted nodes or routers in the routes to prevent unauthorized access.

3. Control the information flows on trusted routes and control the status of trusted routes and trusted routers. And this is detailed in the standard ISO 7498 "Open Systems Interconnection" [7] in terms of the requirements for the control and routing. In TR, routing device (RD) is considered trusted RD only when there is a chance to directly or indirectly control it, and thus have means of changing its configuration. Such RD must be identifiable within the list of all the routes that exist between the sender and the recipient. Thereafter, on each of the identified trusted RD is then be stored unique label. Figure 2 shows a TR in an IP Network.

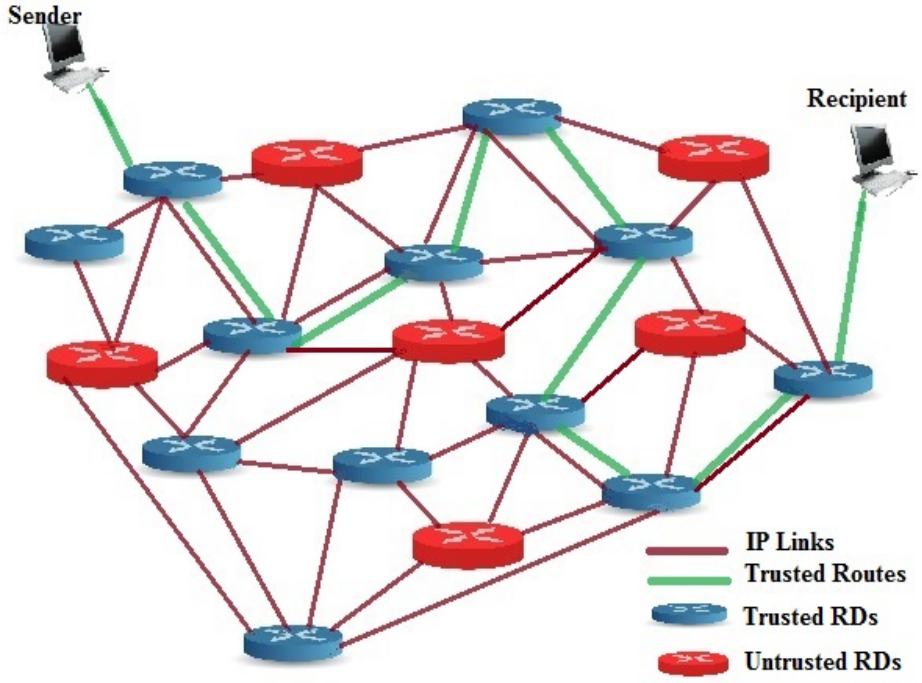

Figure 2. A Trusted Routing in an IP Network

2. Implications of Techniques and Realization Method of TR and MPLS

2.1. Implications of MPLS Techniques and Realization Method

Already, MPLS technology as a service building block and foundation for enterprise virtualization implementation offers the following benefits, according to Luc De Ghein [3], for service providers and enterprises:

I. The use of one unified network infrastructure makes administration and maintenance easier for service provider while the provider's unified network infrastructure is carrying all kinds of customer traffic.

II. Better IP over ATM integration and facilitation of the evolution of legacy services via Any Transport over Multiprotocol Label Switching (AToM) and even the introduction of Layer 2 VPNs as the cost of retaining Frame Relay and ATM infrastructures becomes excessively high. ATM did have plenty of success but its usage was limited to its use as a WAN protocol in the core of service provider networks. So, to better integrate IP over ATM, the networking community came up with a few solutions such as to implement IP over ATM according to the RFC 2684 [4] which specifies how to encapsulate multiple routed and bridged protocols over ATM adaptation Layer (AAL) 5. In this solution, all ATM circuits had to be established manually, and all mappings between IP next hops and ATM endpoints had to be manually configured on every ATM-attached router in the network. Another method was to implement LAN Emulation (LANE). LANE basically allows Ethernet networks with several Ethernet segments to be

bridged together as if the ATM WAN network in the middle were an Ethernet switch. In essence, it makes the networks look like an emulated Ethernet network. And finally, MPOA (Multiprotocol over ATM) that gives the tightest integration of IP over ATM but also the most complex solution. A better solution for integrating IP over ATM is allowed by MPLS. However, this demands that ATM switches must be more intelligent by running an IP routing protocol and implement a label distribution protocol (LDP).

III. BGP-free core – MPLS enables the forwarding of packets based on a label lookup rather than a lookup of the IP addresses, and enables a label to be associated with an egress E-LSR or destination LSR, if the destination is within MPLS domain, rather than with the destination IP address of the packet. The MPLS' core routers or LSRs then no longer need to have the information to forward the packets based on the destination IP address. Thus, the LSRs in the MPLS network no longer need to run BGP. Contrariwise, the E-LSRs of the MPLS network still need to look at the destination IP address of the packet and hence still need to run BGP. Each BGP prefix on the ingress E-LSR has a BGP next-hop IP address associated with it, which also is an IP address of an egress E-LSR. The label that is associated with an IP packet is the label that is associated with this BGP next-hop IP address. Since every LSR forwards a packet based on the attached MPLS label that is associated with the BGP next-hop IP address, then each BGP next-hop IP address of an egress E-LSR must be known to all LSRs. This task is usually accomplished by any of interior gateway routing protocol like OSPF or ISIS.

IV. Optimal traffic flow aided by flexible classification of packets and the optimization of network resources. For instance, when using MPLS VPN, the traffic flows directly, thus optimally, among all sites since there is no need to interconnect routers manually to create virtual circuits between them, to enable them send traffic directly to any other router. Moreover, creating virtual circuits manually is tedious and costly since the requirement most at time is any-to-any connection between sites, which necessitate having a full mesh of virtual circuits between the sites.

V. Traffic engineering – as the idea behind traffic engineering is to optimally use the network infrastructure, especially the links that are underutilized. In MPLS, the traffic that is destined for a particular prefix or with a particular quality of service flow from point A to point B along a path that is different from the least-cost path can be spread more evenly over the available links in the network and make more or better use of underutilized links in the MPLS network.

VI. Label distribution using various protocols such as LDP, BGP and RSVP.

VII. The coexistence of different distribution protocols in the same LSR [5].

VIII. The introduction of value-added services and applications such as QoS, traffic engineering, multicast, and VPN [5].

2.2. Implications of TR Techniques and Realization Method

The set of techniques and realization method of TR allow:

i. for safe transfer of information from the sender to the recipient. The necessity to have control over the information being sent over the Internet has become a growing concern. Sender wants to be sure that only the recipient, that the data is meant for, receives the data being sent. In TR, trusted RD are being used to transfer data via trusted paths using labels.

ii. to dynamically generate routes between sender and recipient. Manually adding routes to the routing tables (RT) of RD within a network may be very tedious, especially when the network is large. Moreover static routing type of configuration is not fault tolerant. Whenever there is a change in the network or a failure occurs between two statically defined nodes (routers), traffic will not be rerouted. Thus all requests to failed routers will ultimately be failing until these routers are repaired or the static routes are updated by the administrator. TR avoids this by using RT dynamically generated by IP routing protocols to map out or re-map out trusted routes through which to securely transmit data.

iii. use secure autonomous systems (ASs) for data transmission.

iv. zombifying RD in the absence of a route that passes only through trusted ASs. This implies the injection into OS of RD personal software modules that must interact with software module of the RD.

v. for control of the traffic flows. After forming a trusted routes and transfer session initiation, there must be control over traffic flow and the state of the trusted RD. Control traffic flow is in two parts: the control over the delivery of packets from the sender to the recipient and the control of the packets to pass only through trusted RD. The first part means that all packets must be delivered to the recipient. Protocol stack TCP/IP provides guaranteed delivery of packets between the sender and recipient, so in order to reduce network load and increase the data transfer rate, additional control measures are not necessary.

vi. for control of the operation of RD. Monitoring the status of the trusted RD to ensure that:

– RD regularly sends data according to their RT;

– RT of RD excludes changes to the RT during the data transfer session;

– Labels that are in the MIB of RD do not change during the data transfer session.

3. Comparision between TR and MPLS

Though the concepts TR and MPLs have a lot in common as does MPLS with ATM, there are distinct differences between the two mechanisms.

3.1. Network Type

MPLS is a network of a group of connected devices, LSRs, and it performs label switching under a single administrative control and functioning in accordance with the unique routing policy. Thus, MPLS is more or less a private network. It connects to other IP networks or Internet via E-LSR at its boundary. TR only creates virtual links within Internet or any other IP networks intermittently. These virtual links changes every now and then as Internet or IP networks RT change or whenever status of RD already participating in TR changes.

3.2. Labeling Method

Method of Label Creation in MPLS could be topology-based (using normal processing of routing protocols e.g. OSPF and BGP), traffic-based (using the reception of a packet to trigger the assignment and distribution of a label) or request-based (using processing of request-based control traffic e. g. RSVP) [5]. In TR mechanism, labeling is via processes that incorporate using routing protocols e. g. BGP and OSPF that allow constructing a map of the routes from the sender to the recipient, to get a list of all the transit RD between the sender and the recipient. Then map out trusted routes based on the degree of trust of the RD through which information flows. On each of such trusted RD identified within the list of all the routes that exist between the sender and the recipient is then stored unique label by injecting into operating system (OS) of RD personal control code. This is possible since every network device has its own administrative database, Management Information Base (MIB), which in the Internet are controlled by the protocol Simple Network Management Protocol, (SNMP). SNMP is being used to remotely manage and control network devices and applications through the exchange of management information between agents i. e. SNMP is software, which runs on managed network devices, and managers (administrative computers) located at the control station [8].

3.3. Data routing

Routing in TR is explicit but dynamic as trusted routes are recalculated each time there is a change in RT of any of RD. Routes are pre-calculated based on trustworthiness of the RD. Within MPLS domain, routing is hop-by-hop (LSR independently selects the next hop for a given Forwarding Equiva-

lence Class – FEC) or explicitly static (the ingress LSR specifies the list of nodes through which the packet traverses) [5].

Differences mentioned above are insignificant considering often-preferred requirements and the fact that for every networks, QoS, network scalability, data integrity and confidentiality are always of higher priority. Table below shows comparison between TR and MPLS based on the aforementioned preferred requirements for a network.

	Security	QoS	Scalability	Data Integrity	Data Confidentiality
MPLS	+ (but within its domain)	++	+	+–	+–
TR	+ (from sender to the recipi-ent)	+–	+	+++	+++

+ => exist, +– => to some extent, ++ => of high degree, +++ => of high degree and priority

4. Discursion

Though the three basic properties of security are availability, integrity and confidentiality. Thus when defining security, one can be more precise by defining which of the security properties is required and to which extent. For instance, in a military environment the most important security property is probably confidentiality. So also is in a bank, though more important to a bank is the integrity of the data. However, availability of the web page is an important factor for online shopping sites. Since primarily the sender, using TR mechanism is having all the three properties of security over the data transfer in an IP network or Internet. And as virtual links being created is in-termittent, the attacker or intruder finds it difficult to get access to the data. In MPLS both sender and recipient or either of the two may not be located within the MPLS domain, even if MPLS domain is properly configured and its internal structure of the core network is not visible to outside networks (and thus secured from various attacks e.g. denial-of-service). Supposing the data is secured while traversing within the MPLS domain, it is subjected to various attacks outside MPLS domain.

TR ensures that data being tranferred is not being tampered with in any form as it is not available to third-party router (i.e. router that is not participating in the process of TR) either in encrypted form or otherwise. And data traverses only through trusted routes, calculated via TR mechanism, from sender to the recipient.

In essence, MPLS provides ability to control where and how traffic is routed on privately owned network, to manage capacity, prioritize different

services, and prevent congestion; while TR provides the ability to control where and how traffic is routed within IP networks globally or over Internet to ensure that data is delivered securely.

In the face of faults and challenges to normal operation, security mechanism in TR encourages re-calculation of trusted routes. The case with MPLS is almost the same where network resiliency is improved via MPLS Fast Reroute.

MPLS, unlike in a traditional IP network, does label switching. In MPLS, the first E-LSR device (ingress router) does a routing lookup but instead of finding a next-hop, it finds the final destination. Based on the destination, it finds a pre-determined path, LSP, from itself to the final destination. It then applies a label based on this information, and the future routers use the label to route the traffic within the MPLS domain without needing to perform any additional IP lookups. The label is removed at the egress router, and then the packet is delivered via normal IP routing to its destination router. TR also uses label-switching but the trusted path are calculated using RT of traditional IP network to map out trusted routes from the sender to the recipient. Then data is delivered solely from sending router to the receiving router via the trusted routes using labels.

When transferring data over Internet or IP networks, TR provides complete traceroute for the sender via IP RT. MPLS, very unlike TR, when using traceroute, label stack information associated with subsequent routing along the LSPs to the receiving router and back to the sending router is not displayed. This undermines security issue in MPLS considering the possibility of availability of mischievous Facility Backup within the domain. Furthermore, this necessitates running services like IPsec over MPLS, especially when either or both the sender and the recipient is located outside MPLS domain. Obvious that attackers could routinely use traceroutes to map how information moved within a given network, so MPLS not allowing to perform a traceroute to eliminate this security threat is an advantage though. Even with traceroute, attackers find it difficult to have access to the data being transferred via TR as routes are not static. They are calculated and recalculated every now and then especially whenever there are changes in IP RT. Moreover, RD can only participate in TR only after it is determined or made a trusted RD.

TR is a mechanism that allow to find one or more trusted routes over IP networks from a sender to a recipient at a gven point in time, and MPLS is becoming a very popular technology for providing virtual private network (VPN) services as many enterprises are thinking of replacing traditional Layer 2 VPNs such as ATM or Frame Relay (FR) with MPLS-based services. Yet the purpose and usability of the two mechanisms are different. TR can run over MPLS if the MPLS domain's E-LSRs (ingress and egress routers) can be

controlled directly or indirectly, and provided the MPLS domain can be trusted.

5. Conclusion

Is Internet going to be within MPLS? Or are we going to restrict all telecommunications to nodes within MPLS domain? Impossible! Furthermore, running IPsec over MPLS is not a guarantee that data, though encrypted, would not get to the hand of third-party or attackers. Also there is no 100% assusrance that encrypted data can not be decrypted by this third-party as far as it has access to the data. Thus TR is mechanism that is prefarable to avoid third-party unauthorized access. TR does not allow third-party having access to copy, modify, delete or redirect the data. Thus TR mechanism, to higher degree than MPLS mechanism, provides for data security, in addition to usage flexilibility, data integrity and data confidentiality. Moreover these two mechanisms are meant for different purposes, one for secure data transfer via trusted routing over IP networks while other is being used mostly as backbone and or private networks. TR is not replacing MPLS, though both offers privacy during data transfer. TR is just another routing mechanism for a different purpose compared to that of MPLS.

References

1. **UTStarcom**, Inc. USA, 2009. MPLS-TP based Packet Transport Networks. [pdf] Available at: <http://www.utstar.com/files/CP-WP-MPLS-TP_PTN-1109.pdf> [Accessed 17 August 2013]

2. **Metaswitch** Networks, 2013. What Is MPLS and GMPLS? [Online] Available at: <http://network-technologies.metaswitch.com/mpls/what-is-mpls-and-gmpls.aspx> [Accessed 17 August 2013].

3. **Luc De Ghein**, 2007. MPLS Fundamentals. Indianapolis: cisco press.

4. **Grossman, D.** and Heinanen, J., 1999. RFC 2684: Multiprotocol Encapsulation over ATM Adaptation Layer 5. Fremont: IETF.

5. **Morrow, M. J.**, and Sayeed, A., 2008. MPLS and next-generation networks: Foundations for NGN and enterprise virtualization. India: Pearson Education.

6. **Tiamiyu, A. Osuolale**, 2013. On the Simulation of Trusted Routing Mechanism. In: SPbSUT, 2nd International Conference on Scientifc and Technical Methods: Topical Issues on Information and Telecommunications in Science and Education. St. Petersburg, Russia 26-27 February 2013. SPbSUT Russia.

7. **International** Standard Organization, 1996. ISO 7498 Information technology: Open Systems Interconnection: Basic Reference Model Part 1 & 2 // Information security architecture. Geneva: ISO.

8. **Larry Walsh,** 2008. SNMP MIB Handbook. Parsippany: Wyndham Press.

Annotation

MPLS technology implements packet switching in multiprotocol networks, using labels. Although, MPLS incorporates concept of data flow control (improves performance, integrates IP and ATM networks, supports creation of virtual channels), there exists problem associated with the integrity and confidentiality of traffic passing through the MPLS network. Trusted routing mechanism ensures the integrity and confidentiality of network traffic. This paper analyzes and compares these two mechanisms in relation to data security, QoS and network scalability.

Тиамийу А. Осуолале

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

ДОВЕРЕННАЯ МАРШРУТИЗАЦИЯ VS MPLS: БЕЗОПАСНОСТЬ ДАННЫХ, КАЧЕСТВО ОБСЛУЖИВАНИЯ И МАСШТАБИРУЕМОСТЬ

Аннотация

Технология MPLS реализует коммутацию пакетов в многопротокольных сетях, используя механизм меток. Несмотря на то, MPLS дает преимущества управления потоками данных (повышает производительность, интегрирует IP и ATM-сети, позволяет создавать виртуальные каналы), существует проблема потери целостности и конфиденциальности трафика, проходящего через сеть MPLS. Механизм доверенной маршрутизации гарантирует целостность и конфиденциальность сетевого трафика. В статье анализируются и сравниваются эта два механизма по отношению к безопасности данных, QoS и масштабируемости сети.

Ключевые слова: MPLS, доверенная маршрутизация, безопасность данных, QoS, целостность данных, масштабируемость сети.

Тиамийу А. Осуолале – аспирант кафедры «Защищенные системы связи» Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», ozutiams@yahoo.com

СИСТЕМНЫЙ АНАЛИЗ И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 621.39

Е. Г. Богданова, С. Ф. Глаголев
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ФИЛЬТРОВ НИЖНИХ ЧАСТОТ ДЛЯ ОПТИЧЕСКИХ РЕФЛЕКТОМЕТРОВ

оптический рефлектометр, сигнал обратного рассеяния, зондирующий оптический импульс, рефлектограмма, фильтр нижних частот, динамический диапазон, мертвая зона.

В данной статье, являющейся продолжением работы[1], приведены результаты математического моделирования сигнала обратного рассеяния (СОР) от волоконно-оптической линии связи (ВОЛС) небольшой протяженности. Фактически такой сигнал может быть получен с помощью оптического рефлектометра во временной области (ОР), являющегося наиболее универсальным прибором для анализа ВОЛС. Моделируемая линия (рис. 1) содержит отражающую неоднородность, образованную, например, некачественным сварным или разъемным соединением двух оптических волокон (ОВ). При расчете были учтены отражения посылаемого ОР оптического импульса от начала и конца ОВ, а также многократные переотражения от указанных неоднородностей, что имеет место в практической рефлектометрии.

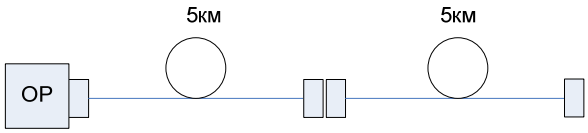

Рис. 1. Моделируемая волоконно-оптическая линия связи

Изображение на дисплее ОР представляет собой СОР, прошедший первичную цифровую обработку, включая фильтрацию, накопление и логарифмирование. Особое внимание в работе уделено моделирова-

[1] Былина М. С., Богданова Е. Г., Глаголев С. Ф. Применение метода обратного рассеяния для измерения параметров волоконно-оптических линий связи // Материалы II Международной научно-технической и научно-методической конференции «Актуальные проблемы инфотелекоммуникаций в науке и образовании». – СПб. : СПбГУТ, 2013.

нию фильтра нижних частот (ФНЧ), который подавляет шумы фотоприемного устройства (ФПУ). В работе проведено сравнение двух фильтров нижних частот: ФНЧ второго порядка, образованного RC-цепью и реализованного программно фильтра, основанного на методе скользящего усреднения. Фильтры сопоставляются по параметрам «мертвая зона» по событию и «мертвая зона» по затуханию.

Рефлектограмма представляет собой зависимость мощности СОР от расстояния или времени задержки. ОР являются цифровыми приборами, поэтому мощность СОР (P_j) определяется для дискретных точек j. При моделировании рефлектограмма разбивается на ряд участков. Для расчета каждого из участков линии используется своя формула.

Ввиду громоздкости выражений для СОР на отдельных участках рефлектограммы с учетом многократных отражений в статье приведены лишь некоторые формулы.

Выражение для расчета мощности СОР от 1 участка рефлектограммы, отражение от входного торца ОВ:

$$P_j = \left[\left(P_0 \cdot 10^{\frac{Yso1}{5}} \cdot A1 \cdot \left(1 - 10^{-a1 \cdot \frac{l_j}{5}} \right) \right) + P_0 \cdot ra \;\; if \;\; l_j \le dl \right].$$

Выражение для первого однородного пятикилометрового участка:

$$P_j = \left(P_0 \cdot 10^{\frac{Yso1}{5}} \cdot A2 \cdot 10^{-a1 \cdot \frac{l_j}{5}} \cdot (1 - ra) \right) \;\; if \;\; l_j \ge dl \cup l_j \le l_0.$$

Выражение для расчета отражения от соединения двух ОВ:

$$P_j = \left(P_0 \cdot (1 - ra) \cdot 10^{\frac{-a1 \cdot l_0}{5}} \cdot \left[10^{\frac{Yso1}{5}} \cdot A2 \cdot 10^{-a1 \cdot \frac{[\![(l]\!]_j - l_0}{5}} \cdot \left(1 - \frac{l_j - l_0}{dl} \right) + \right. \right.$$

$$\left. \left. + \left[10^{\frac{Yso2}{5}} \cdot A2 \cdot 10^{\frac{[\![-ac - a1 \cdot (l]\!]_j - l_0)}{5}} \cdot \left(\frac{l_j - l_0}{dl} \right) \right] + r1 \right] \right)$$

$$if \;\; l_j > l_0 \cup l_j \le l_0 + dl.$$

В приведенных выражениях: P_0 – мощность зондирующего импульса, Yso – относительный уровень СОР для ближней зоны волоконно-оптического тракта для одномодового ОВ на длине волны излучения 1550 нм, $A1$ и $A2$ – величины, введенные для удобства записи, зависящие от длительности импульса, групповой скорости распространения оптического излучения и коэффициентов рассеяния для волокон ($a1$ и $a2$); l_j – расстояние до рассматриваемой точки; l_0 – длина первого однородного участка; ra, $r1$ – коэффициенты отражения от неоднородностей.

В результате моделирования была получена зависимость уровня СОР от расстояния, представленная на рисунке 2. На рефлектограмме видны сигналы обусловленные отражениями от начала и конца ВОЛС.

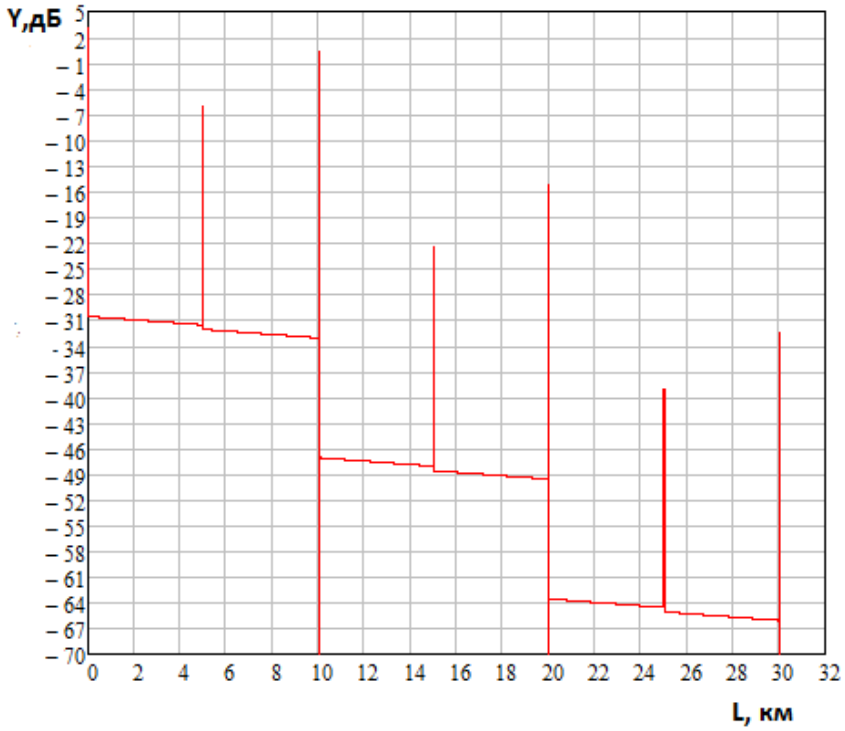

Рис. 2. Зависимость СОР от расстояния

Моделирование сигнала с аддитивным шумом ФПУ

На рисунке 2 показана зависимость оптической мощности от расстояния. При преобразовании оптической мощности в электрический сигнал в ФПУ к нему добавляется аддитивный шум, распределенный по нормальному закону. Обычно отношение сигнала к шуму значительно меньше 1, особенно при больших протяженностях ВОЛС. Для увеличения отношения сигнала к шуму, используется фильтрация, а также накопление результатов многократных измерений (усреднение).

При моделировании шума ФПУ можно использовать пороговую мощность приемника *Pmino*, которая определяется в частотном диапазоне достаточном для приема импульсов длительностью *tu*0 = 1 нс. Фактически это среднеквадратическое отклонение (СКО) случайной величины.

Пороговая мощность ФПУ с учетом реальной длительности импульса *tu* в нс и числа накоплений *N*:

$$Pminn = \frac{Pmino}{\sqrt{N \cdot tu}}.$$

Для СОР от j – точки при i-м измерении с учетом шумов ФПУ и накопления справедливо:

$$P1_{j,i} = Pnn_i + P_j,$$

где Pnn_i – мгновенное значение шума ФПУ.

Алгоритмы фильтрации. Моделирование аппаратного ФНЧ

В ходе работы были рассчитаны параметры ФНЧ, представляющего собой RC-цепь второго порядка с использованием алгоритма моделирования КИХ-фильтра из теории цифровой обработки сигналов. Сигнал, прошедший такой фильтр будет определяться выражением:

$$[\![P2]\!]_{\downarrow} j = \beta 0 \cdot [\![P1]\!]_{\downarrow} j + \beta 1 \cdot [\![P1]\!]_{\downarrow} (j-1) + \beta 2 \cdot [\![P1]\!]_{\downarrow} (j-2) - \\ - \alpha 1 \cdot [\![P2]\!]_{\downarrow} (j-1) - \alpha 2 \cdot [\![P2]\!]_{\downarrow} (j-2),$$

где $\beta 0 = \dfrac{T^2}{(2 \cdot \tau + T)^2}$;

$\beta 1 = 2 \cdot \beta 0$;

$\beta 2 = \beta 0$;

$\alpha 1 = \dfrac{2 \cdot (T - 2 \cdot \tau)}{(T + 2 \cdot \tau)}$;

$\alpha 2 = \dfrac{(2 \cdot \tau - T)^2}{(2 \cdot \tau + T)^2}$;

$T = \dfrac{tu}{kiz}$.

Из приведенных выше формул видно, что коэффициенты перед отсчетами сигнала в текущий и предшествующие моменты времени зависят только от длительности импульса и постоянной времени цепи τ. Параметр kiz – характеристика избыточности измерений, она показывает, на сколько полное число точек на рефлектограмме превышает минимально необходимое, при котором временной интервал между отсчетами равен длительности импульса.

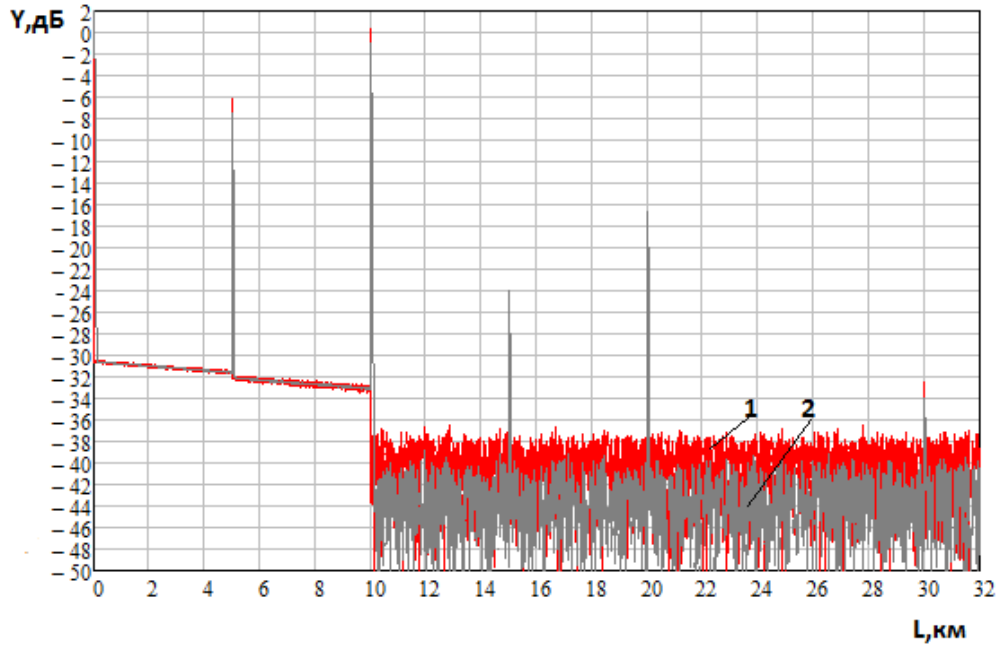

Рис. 3. СОР без фильтрации (1) и с фильтрацией (2)

На рисунке 3 представлены уровни СОР без учета и с учетом фильтрации. Видно, что после фильтрации отношение сигнала к шуму увеличилось, а, следовательно, увеличился и динамический диапазон – один из важнейших параметров ОР.

Моделирование программного ФНЧ

При невозможности аппаратной реализации перестраиваемого ФНЧ или в целях дальнейшего увеличения отношения сигнала к шуму при уже установленном ФНЧ может использоваться дополнительная цифровая обработка реализующая, например, метод скользящего усреднения.

Суть предлагаемого метода состоит в анализе значений СОР в ближайших точках. Так значением сигнала в текущей точке является среднее арифметическое значений СОР в нескольких (в нашем случае в 8) предыдущих моментах времени. При этом необходимо постоянно сравнивать текущее значение со значениями мощности сигнала в предыдущих точках. Если это отклонение достаточно велико, то есть имеет место френелевское отражение или скачок затухания, следует запретить фильтрацию, то есть в качестве значения в текущей точке записывать не среднее арифметическое предыдущих, а непосредственно значение сигнала, пришедшего на вход устройства обработки.

$$w_j = \begin{cases} s \ if \ \dfrac{P1_j - \dfrac{\sum_{k=j-10}^{j-1} P1_j}{10}}{P1_j} < 30; \\ 1 \ otherwise. \end{cases}$$

Приведенная формула определяет число слагаемых w_j для вычисления среднего арифметического предыдущих точек. Если сигнал $P1$ в точке j превосходит среднее арифметическое 10 предыдущих точек более, чем в 30 раз, тогда число слагаемых для суммирования равно 1, что равноценно отсутствию фильтрации. Иначе это число равно s (для данного расчета $s = 8$).

$$m_j = \begin{cases} \dfrac{\sum_{k=0}^{j} P1_k}{j+1} \ if \ \ j < w_j; \\ \dfrac{\sum_{k=j-w_j+1}^{j} P1_k}{w_j} \ if \ \ j \geq w_j. \end{cases}$$

Последняя формула определяет сигнал, прошедший ФНЧ (рис. 4). Она описывает непосредственно вычисление среднего арифметического. Дополнительное условие $j < w_j$ наложено для точек, порядковый номер которых меньше числа усредняемых точек, то есть для ближней зоны рефлектограммы.

Сравнение двух методов фильтрации

Основной параметр, по которому могут сравниваться приведенные в работе алгоритмы фильтрации – это протяженность мертвых зон (МЗ), то есть минимальных расстояний после френелевского отражения, на котором ОР способен обнаружить другое событие. Мертвые зоны можно разделить на МЗ по событию и по затуханию. Для сравнения рассмотрим отраженный от соединения двух ОВ импульс, расположенный на расстоянии 5 км от начала линии (рис. 5).

МЗ по затуханию определяется на уровне 0,5 дБ выше участка, следующего за френелевским отражением. Необходимо оценить ширину импульса на этом уровне. МЗ для аппаратного фильтра составляет приблизительно 8,5 м, для программного – 2,5 м. Не смотря на то, что оба показателя укладываются в норму, принятую для рефлектометров (она составляет 10 м), очевидно превосходство программного фильтра, так как он практически исключает наличие расширения заднего фронта импульса.

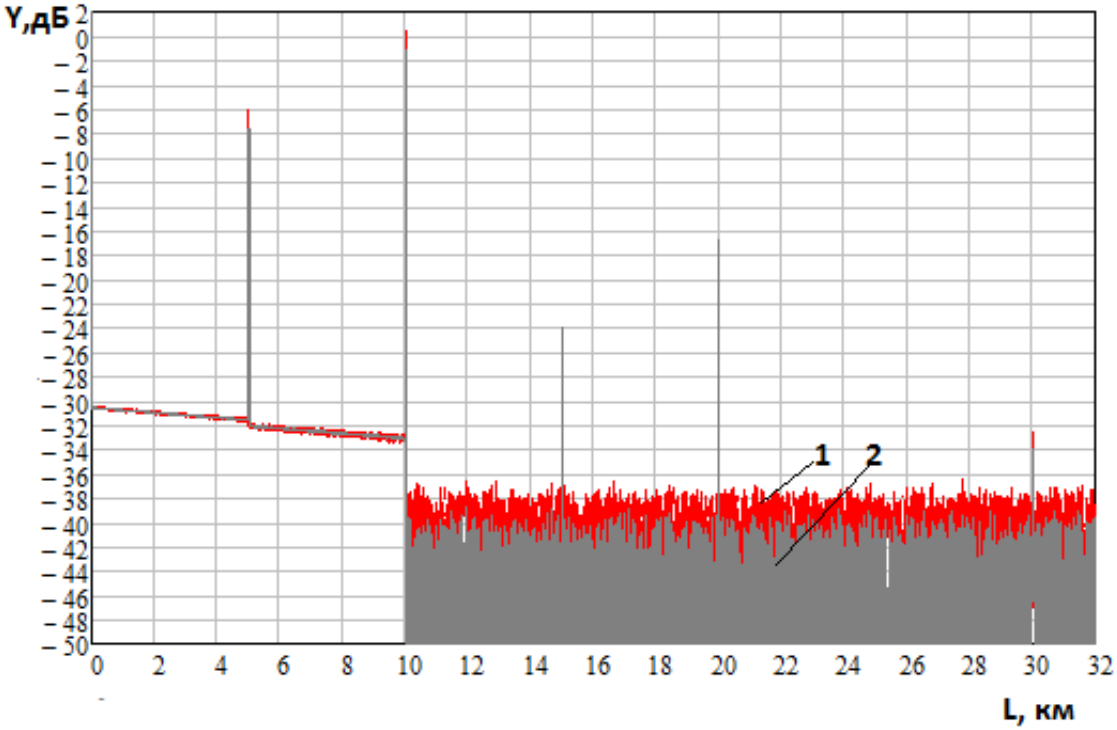

Рис. 4. СОР без фильтрации (1) и с фильтрацией (2)

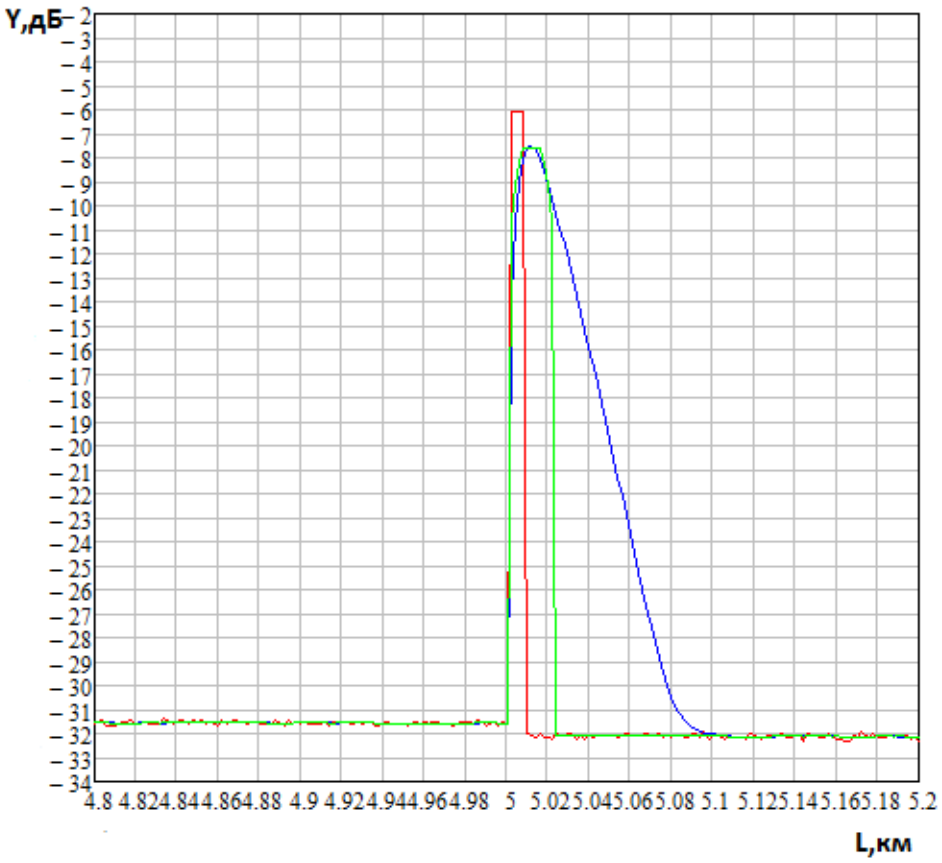

Рис. 5. Френелевское отражение от неоднородности:
1 - без фильтрации, 2 - программный ФНЧ, 3 – аппаратный ФНЧ

МЗ по событию определяется на уровне 1,5 дБ ниже максимального уровня отраженного импульса. Расчет показал, что для программного и аппаратного фильтра этот параметр идентичен и составляет 1,5 м, что также соответствует норме (МЗ по событию должно быть меньше 3 м).

По полученным результатам можно сделать вывод о преимуществах программного фильтра над аппаратным, так как качество фильтрации шумов практически одинаково, а МЗ по событию при использовании предложенного метода меньше.

Аннотация

В работе рассмотрены принципы оптической рефлектометрии, проведено моделирование сигнала обратного рассеяния для линии небольшой протяженности с учетом шумов фотоприемного устройства, накопления и фильтрации. Были сравнены программный и аппаратный фильтры по протяженности мертвых зон.

E. G. Bogdanova, S. F. Glagolev

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

MATHEMATIC MODELING OF LOW-PASS FILTERS FOR OPTIC REFLECTOMETERS

Annotation

The principles of optic reflectometry were described in the article, the backscattered signal for a line of small extent was simulated with the photodetector noise, accumulation and filtering. The software and hardware filters were compared on the length of the dead zones.

Keywords: OTDR, backscatter signal, probe optical pulse, low-pass filter, dynamic range, dead zone.

Богданова Евгения Геннадьевна - студентка факультета Инфокоммуникационных сетей и систем Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», Evgenia15V@gmail.com.

Глаголев Сергей Федорович - заведующий кафедрой Фотоники и линий связи факультета Инфокоммуникационных сетей и систем Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», glagolevsf@yandex.ru.

УДК 004.652.42

С. Д. Бороненко, О. Ю. Ильяшенко, С. В. Хорошенко

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

ПОДХОД К ПРОЕКТИРОВАНИЮ РЕКУРСИВНЫХ ОТНОШЕНИЙ

отношения, рекурсивные отношения, реляционная модель данных, проектирование рекурсивных отношений

В предлагаемом исследовании показано, что использование рекурсивных отношений позволяет решать следующие задачи:

1. существенно упростить концептуальную модель данных;
2. реализовать концептуальную модель, инвариантную к реляционной модели данных.

Решение первой из перечисленных задач существенно упрощает концептуальную модель данных, тем самым позволяя снизить затраты как на этапе проектирования, так и затраты на сопровождение базы данных.

Вторая задача весьма неординарна. Решение данной задачи позволяет реализовать возможность создания «настраиваемой» концептуальной модели данных под спроектированную реляционную модель данных. Иначе говоря, решение указанной задачи позволяет создавать гибкие СУБД, с точки зрения адаптации их концептуальной модели под разные варианты реляционной модели в рамках, по крайней мере, одной предметной области.

Известно, что важной составляющей моделирования базы данных является отбор и анализ данных [1].

Приведем пример анализа данных для типичной задачи «Классификатор товаров». Классификатор товаров имеет иерархическую структуру данных (табл. 1).

ТАБЛИЦА 1. Характерный фрагмент двухуровневого классификатора товаров

Код группы	Наименование товарной группы (ТГ)	Код подгруппы	Наименование подгруппы товаров (ПГТ)
1	продукты питания	1	напитки
		2	бакалея
2	промтовары	1	обувь
…	…	…	…

Реляционная модель двухуровневого классификатора товаров после нормализации будет иметь следующий вид (рис. 1).

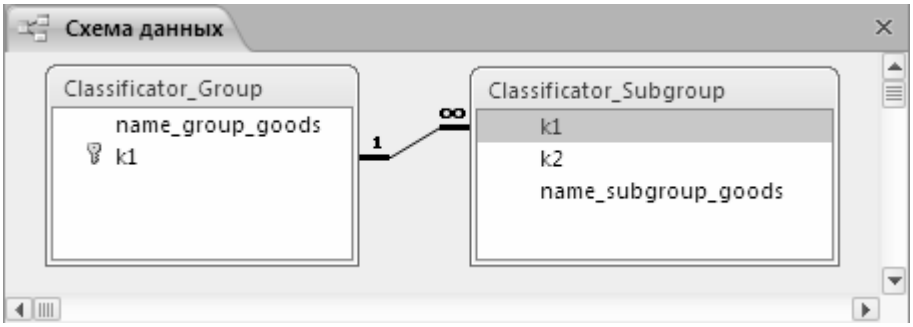

Рис. 1. Нормализованная реляционная модель двухуровневого классификатора товаров

Одним из методов проверки корректности нормализованных отношений (данных) является процедура восстановления исходных данных на основе SQL-запроса.

Ниже приведен пример SQL-запроса, полностью восстанавливающий исходные данные:

```
SELECT Classificator_Group.k1, Classifica-
tor_Group.name_group_goods, Classifica-
tor_Subgroup.k2, Classifica-
tor_Subgroup.name_subgroup_goods
FROM Classificator_Group INNER JOIN Classifica-
tor_Subgroup ON Classificator_Group.k1 = Classi-
ficator_Subgroup.k1;
```

Представление данных на основе запроса к модели двухуровневого классификатора товаров имеет следующий вид (рис. 2):

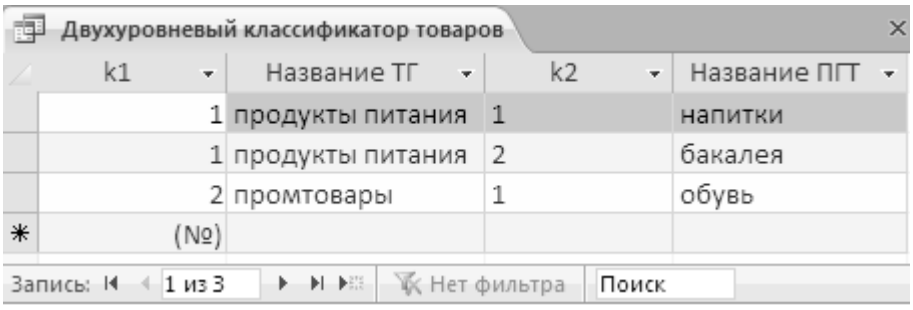

Рис. 2. Результат выполнения запроса двухуровневый классификатор товаров

Как видно из представления данных (рис. 2), двухуровневый классификатор товаров восстановлен полностью, что свидетельствует о проведенной декомпозиции без потерь на основе нормализации. Повторение данных (2 строка 1/продукты питания) при отображении легко исключается при использовании запроса при документировании представления данных. Например, в MS Access при документировании запроса посредством объектов – Отчетов - эта ситуация корректируются подключением опции <Исключить повторы>.

В современных базах данных моделирование возможно на основе использования рекурсивных отношений (рис. 3).

ks	kx	k	m
11		1	продукты питания
12		2	промтовары
13	11	1	напитки
14	11	2	бакалея
15	12	1	обувь

Рис. 3. Рекурсивное отношение, моделирующее двухуровневый классификатор

Поля в представленном реляционном отношении имеют следующие значения:

- ks - суррогатный ключ;
- kx - внешний ключ;
- k - универсальный код групп/подгрупп;
- m - универсальное наименование групп/подгрупп.

Многие СУБД (Oracle, MS Accessи др.) имеют прямые или косвенные средства реализации рекурсивного отношения.

Например, реализация рекурсивного отношения, представленного на рисунке 3, средствами MS Access выглядит следующим образом (рис. 4).

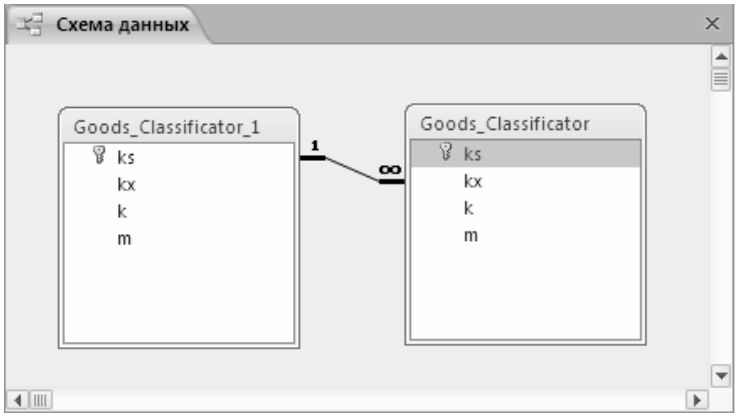

Рис. 4. Реализация рекурсивного отношения средствами MS Access

Запрос, восстанавливающий данные по товарным группам и подгруппам без потерь, выглядит следующим образом:

```
SELECT Goods_Classificator_1.ks,
First(Goods_Classificator_1.kx) AS kx,
First(IIf([Goods_Classificator_1].[kx] Is
Null,[Goods_Classificator_1].[k],[Goods_Classifi
cator_1].[k])) AS k,
First(Goods_Classificator_1.m) AS m
FROM Goods_Classificator AS
Goods_Classificator_1 LEFT JOIN
Goods_Classificator
ON Goods_Classificator_1.ks =
Goods_Classificator.kx
GROUP BY Goods_Classificator_1.ks
ORDER BY Goods_Classificator_1.ks;
```

Результат представления данных по товарным группам и подгруппам приведен на рисунке 5.

Товарные группы

ks	kx	k	m
11		1	продукты питания
12		2	промтовары
13	11	1	напитки
14	11	2	бакалея
15	12	1	обувь

Запись: 5 из 5 Нет фильтра Поиск

Рис. 5. Результат представления данных
по товарным группам и подгруппам

Как видно, результат представления данных идентичен результату рекурсивного отношения (рис. 3).

Определение внешнего ключа отношения в реляционной модели данных [1] содержит необходимые предпосылки существования рекурсивного отношения. Определение рекурсивного отношения через определение внешнего ключа можно сформулировать следующим образом: рекурсивными отношениями R являются отношения, имеющие внешний ключ IDx, множество значений которого определяется множеством значений первичного ключа ID отношения R.

Классическое проектирование реляционной модели данных на основе нормализации не предусматривает формальных процедур, приводящих к реализации рекурсивных отношений. Можно привести ряд

примеров, таких как проектирование модели классификатора строго неограниченного числа уровней, которые практически разрешимы только на основе рекурсивных отношений.

В моделях типа «сущность - связь», например, IDEF1X, предусмотрены сущности с использованием рекурсивных связей [2]. При проектировании концептуальной модели данных на основе IDEF1X возможно моделирование сущностей одного и того же класса на основе сущности с рекурсивными связями. Однако, использование сущности с рекурсивными связями в модели осуществляется на основе семантики предметной области, которой владеет эксперт, что нельзя отнести к формальным методам проектирования.

Рассмотрим пример обобщённого классификатора товаров для «неограниченного» числа уровней (n уровней).

ТАБЛИЦА 2. Обобщенный классификатор товаров

Код группы	Наименование товарной группы	Код подгруппы	Наименование подгруппы товаров	…	Код n-ой подгруппы	Наименование подгруппы товаров *n*-го уровня
1	продукты питания	1	напитки	…	01	
		2	бакалея	…		
2	промтовары	1	обувь	…	01	туфли
…	…	…	…	…		

Обобщенную реляционную модель для формально неограниченного числа уровней можно представить следующим образом.

Рассмотрим 1NF нормализованное отношение R^n (рис. 6) с вложенными функциональными зависимостями (в отношении используются данные на примере классификатора товаров (табл. 2). Пусть отношение R^n определено на двух множествах атрибутов: $K = \{k_1, k_2, \ldots, k_n\}$ и $M = \{m_1, m_2, \ldots, m_n\}$, где K является множеством атрибутов потенциального ключа отношения, а M - множеством неключевых атрибутов. Будем называть рекурсивными потенциальными ключами такие атрибуты отношения, для которых цепочно выполняется множество функциональных зависимостей вида:

$$\begin{aligned} &\{k_1\} \rightarrow \{m_1\} ; \\ &\{k_1, k_2\} \rightarrow \{m_2\} ; \\ &\ldots \\ &\{k_1, k_2, \ldots, k_n\} \rightarrow \{m_n\} . \end{aligned} \qquad (1)$$

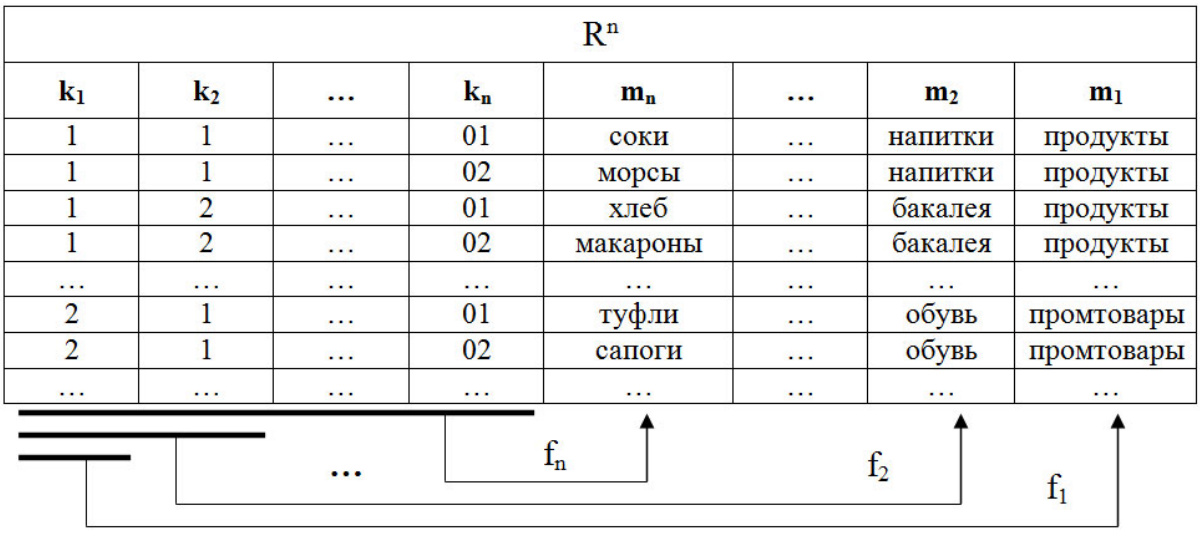

Рис. 6. 1NF отношение R^n с рекурсивными потенциальными ключами и цепочно-вложенными функциональными зависимостями

Обобщенный вид отношения R^n по параметру n не позволяет осуществить конкретное проектирование схемы данных в рамках 2NF нормализации, т. к. не конкретизировано значение n. Приведем пример для n = 3 2NF нормализации отношения R^n, в этом случае нормализация отношения R^n будет иметь вид (рис. 7):

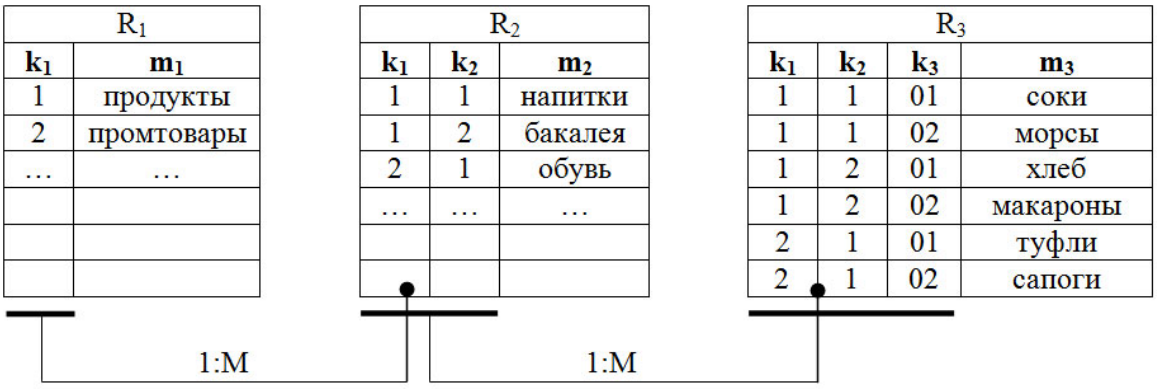

Рис. 7. 2NF нормализация отношения $R^n|_{n=3}$

Таким образом, классический подход к проектированию на основе нормализации имеет следующие недостатки. Во-первых, без конкретизации параметра n невозможно однозначно осуществить проектирование схемы данных, т. к. число отношений не конкретизировано. Во-вторых, с повышением уровня вложенности отношений (начиная с n = 3), внешний ключ отношений становится композитным, что усложняет реализацию модели.

Устранение указанных недостатков классического подхода проектирования реляционной модели предлагается осуществлять посредством

эквивалентной замены отношения вида R^n рекурсивным отношением RR{ks, kx, E}, где ks - суррогатный ключ, kx - внешний ключ, E = {k, m} множество универсальных уровневых атрибутов, в котором, k - атрибут уровневого индекса, m - уровневый неключевой атрибут.

Поясним эквивалентную замену на примере проектирования рекурсивного отношения RR для отношения $R^n|_{n=3}$.

Рис. 8. Эквивалентная замена отношения $R^n|_{n=3}$
на рекурсивное отношение RR

В рассматриваемом отношении RR введён суррогатный ключ ks, который детерминирует все остальные атрибуты отношения RR. Внешним ключом, ссылающимся на первичный ключ ks, является атрибут kx. Заметим, что атрибут kx должен удовлетворять всем требованиям внешнего ключа. В отношении RR можно выделить три уровня рекурсии, соответствующие трём отношениям, получаемым в результате применения классического метода нормализации. Преимуществом использования рекурсивного отношения RR является то, что число моделируемых уровней n может быть теоретически неограниченным. При этом структура самого отношения RR остаётся инвариантной числу уровней цепочно-вложенных функциональных зависимостей (1).

На рисунке 9 представлен примерный набор данных рекурсивного отношения RR для n = 4 уровней классификатора товаров. Как видно из иллюстрации структурное увеличение числа уровней моделируемого классификатора не требует структурных изменений модели — рекурсивного отношения RR.

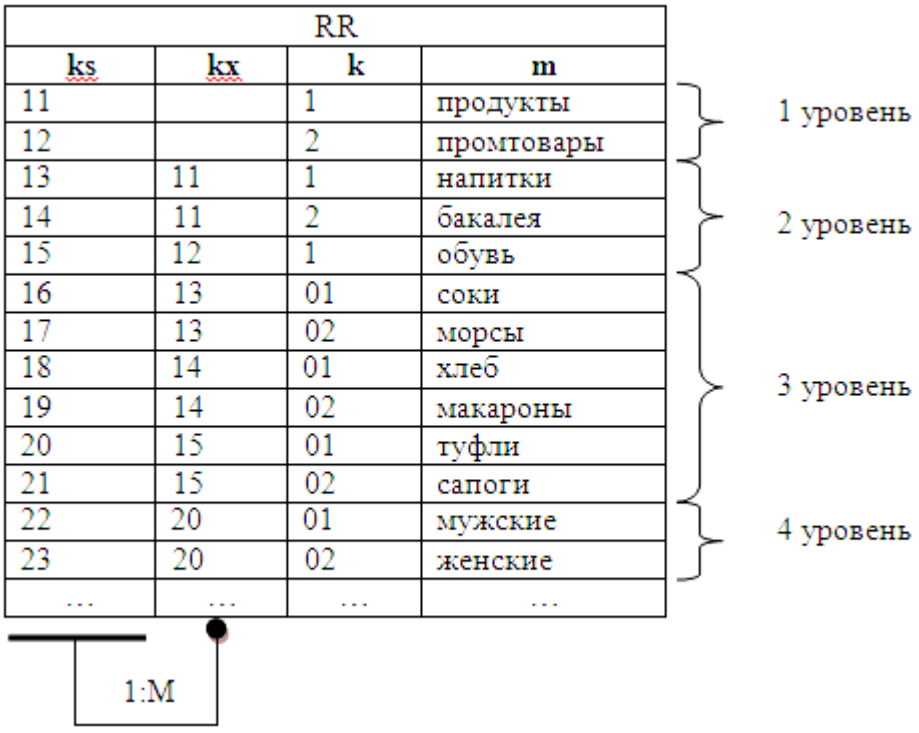

Рис. 9. Примерный набор данных рекурсивного отношения RR, моделирующего отношение $R^n|_{n=4}$

Интересным является то, что структура запросов для представления данных рекурсивного отношения, моделирующего обобщенное отношение R^n, имеет регулярную цепочную структуру.

Таким образом, проектирование реляционной модели данных предлагаемым методом на основе рекурсивных отношений позволяет реализовать обобщенную реляционную модель данных на единственном рекурсивном отношении. Реализуется уникальное свойство «настраиваемости» СУБД в случае обобщенной реляционной модели.

Библиографический список

1. **Дейт, К. Дж.** Введение в системы баз данных : пер. с англ. - 8-е изд. - М. : Издательский дом «Вильямс», 2005. - 1328 с.

2. **Крёнке, Д.** Теория и практика построения баз данных. - 9-е изд. - СПб. : Питер, 2005. - 859 с.

Аннотация

Рекурсивная модель данных предусматривает отношения с рекурсивными связями. Классические методы проектирования реляционной модели не отражают формальных процедур проектирования, позволяющих реализовать рекурсивные отношения. Предлагается подход к проектированию курсивных отношений.

S. D. Boronenko, O. Y. Iliashenko, S. V. Khoroshenko

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

DESIGN APPROACH RECURSIVE RELATIONS

Annotation

Recursive data model provides for relations with recursive relations. Classical methods of designing relational model does not reflect the formal procedures of design, enable to implement the recursive relationships. An approach to the design of italic relations.

Keywords: relations, recursive relations, relational data model, the design of recursive relations.

References

1. **Dejt, K. Dzh.** Vvedenie v sistemy baz dannyh : per. s angl. – 8-e izd. – M. : Izdatel'skij dom «Vil'jams», 2005. – 1328 s.

2. **Krjonke, D.** Teorija i praktika postroenija baz dannyh. – 9-e izd. – SPb. : Piter, 2005. – 859 s.

Бороненко Сергей Дмитриевич – кандидат технических наук, доцент, профессор кафедры Безопасности информационных систем Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», bsd1@yandex.ru

Ильяшенко Оксана Юрьевна - кандидат технических наук, доцент кафедры Безопасности информационных систем Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», ioy12@yandex.ru

Хорошенко Сергей Викторович - кандидат технических наук, доцент, заведующий кафедрой Безопасности информационных систем Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», khoroshenko@mail.ru

УДК 681.883

В. А. Сапрыкин

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

М. Б. Гладких

ЗАО «ТЕЛРОС»

НОВАЯ ТЕХНОЛОГИЯ ОБРАБОТКИ ГИДРОЛОКАЦИОННЫХ СИГНАЛОВ, ОТРАЖЕННЫХ ОТ БЫСТРОДВИЖУЩИХСЯ ОБЪЕКТОВ БИОЛОГИЧЕСКОЙ ПРИРОДЫ

ГЧМ сигнал, корреляционный отклик, шум

Современная техническая гидроакустика развивается в направлении формирования процедур обработки сигналов эффективно функционирующих в условиях обнаружения высокоскоростных и быстро маневрирующих подводных живых организмов. Однако теоретические исследования в этом направлении столкнулись с определенными трудностями фундаментального плана, затрагивающих базовые основы методов обработки.

Всякое физическое явление G, описывающее передачу, рассеивание от сцены и прием акустической энергии имеет симметричную природу (обладает групповой структурой G). В этом случае изучение физического явления связывают с изучением представлений группы G. Это означает, что группа представляется в пространстве отображений (операторных функций на группе) T согласно правилу:

$$T(g_1 g_2) = T(g_1)T(g_2) \text{ для всех } g_1, g_2 \in G \text{ и } T(e) = I_0,$$

где e – нейтральный элемент группы; I_0 – тождественное отображение (действие).

Как правило, в гидролокационных системах при корреляционной обработке учитываются дальность и радиальная скорость цели, которые описываются с помощью преобразований времени t: $t \to \alpha t + \tau$, где α – доплеровский параметр, τ – параметр задержки. Вместо приближенного доплеровского эффекта, описываемого сдвигом частот спектральной функции сигнала используется точное описание эффекта Доплера. Необходимость такого описания преобразования сигнала объясняется большим числом Маха $2v/c$, где v – относительная скорость сближения

цели, c – скорость звука в воде. При таком подходе вместо известной функции неопределенности (ФН) Вудворда [1] применяют широкополосную функцию неопределенности [1–5], которую определяют соотношением:

$$\Psi(t,\alpha)=\sqrt{\alpha}\cdot\int_0^{\infty}\psi(\alpha\cdot f)\cdot\overline{S(f)}\exp(i2\pi ft)\frac{df}{f},$$

где $Y(f)$ – спектральная функция Фурье принимаемой реализации, $S(f)$ – спектральная функция Фурье сигнала.

Решение поставленной задачи формирования процедуры согласованной фильтрации возможно при условии введения дополнительного числового параметра β, отвечающего за линейные изменения доплеровского эффекта, что достигается использованием специальной группы, которую называют инверсной группой преобразований времени.

Элемент группы инверсных преобразований времени IB определяется правилом:

$$g(\alpha,\beta)t\to\frac{\alpha t}{-\beta t+1}$$

с обратным элементом $g(\alpha,\beta)^{-1}t=\dfrac{t/\alpha}{\beta t/\alpha+1}$ и представляется в виде произведения матриц

$$\begin{pmatrix}\alpha & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0\\ -\beta & 1\end{pmatrix}=\begin{pmatrix}\alpha & 0\\ -\beta & 1\end{pmatrix}\text{ и }\begin{pmatrix}\alpha & 0\\ -\beta & 1\end{pmatrix}\begin{pmatrix}1/\alpha & 0\\ \beta/\alpha & 1\end{pmatrix}=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}.$$

Важно заметить, что группа матриц $\begin{pmatrix}1 & 0\\ -\beta & 1\end{pmatrix}$ образует коммутативную инверсную подгруппу I, а группа матриц $\begin{pmatrix}\alpha & 0\\ 0 & 1\end{pmatrix}$ образует коммутативную подгруппу доплеровских преобразований. Напротив группа IB является некоммутативной подгруппой.

Подгруппа I является нормальным делителем группы IB, поэтому множество IB/I – факторгруппа.

Заметим, что факторгруппа IB/I изоморфна мультипликативной подгруппе B. Отсюда следует, что задачу исследования IB/I можно свести к изучению подгруппы B. Смежные классы определяются элементами $g(\alpha,0)$ – доплеровскими преобразованиями. Каждое такое преобразование дает свой смежный класс $B=\{g:g(\alpha,\beta),\forall\beta\in R^1\}$.

Группа IB не инвариантна относительно сдвига. Это означает, что при обработке сигналов с IB симметрией необходимо иметь информа-

цию о параметре задержки τ или сформировать перебор по данному параметру.

Таким образом введение инверсной группы преобразования времени позволило записать переменный доплеровский эффект при условии определенности по параметру задержки (приемнику известна информация о задержке сигнала) в виде:

$$s\left(g^{-1}t\right) = s\left(\frac{t/\alpha}{\frac{t\beta}{\alpha}+1}\right).$$

Аналогом преобразования Фурье для группы I является инверсное преобразование Фурье или просто инверсное преобразование:

$$\tilde{S}(s) = I\{s(t)\} = \int_{-\infty}^{\infty} s(t)\exp(i2\pi s/t)\frac{dt}{t^2}, \qquad (1)$$

где $I\{...\}$ – инверсное преобразование.

Запишем обратное инверсное преобразование $I^{-1}\{...\}$

$$I^{-1}\left\{\tilde{S}(s)\right\} = \int_{-\infty}^{\infty} \tilde{S}(s)\exp(-i2\pi s/t)ds = s(t), \qquad (2)$$

где s – инверсная частота.

Описание инверсного преобразования Фурье позволяет определить аналог широкополосной функции неопределенности в области инверсных частот, в соответствии с выражением [7, 8]:

$$\tilde{\Psi}(\alpha,\beta) = \left|\frac{1}{\sqrt{\alpha}}\int_0^{\infty}\tilde{S}\left(\frac{s}{\alpha}\right)\overline{\tilde{S}(s)}\exp\left(-i2\pi s\frac{1}{\beta}\right)ds\right|^2. \qquad (3)$$

Соотношение (3) является базовым оператором решения задачи формирования согласованного фильтра в условиях обнаружения и определения КПДЦ высокоскоростных живых организмов.

В силу неинвариантности инверсной группы относительно сдвига, для определения задержки формируется специальная операция перебора процедуры (3) по всем моментам времени с шагом, определяемым интервалом корреляции входного процесса. Таким образом, при обработке сигналов в области времени производиться его секционирование и сдвиг блока для осуществления перевода в инверсную область и компенсации задержки. Причем компенсация задержки τ заключается в выборе истинного начала блока. Далее сигналы секций переводятся в инверсную область времени. Там сигнал подвергается корреляционной обработки, основанной на согласованной фильтрации (3) принятого процесса и эта-

лонного сигнала, заключающейся в вычислении инверсного преобразования Фурье (ИПФ) принятого процесса, в вычислении результата перемножения этого преобразования с комплексно сопряженным ИПФ эталонного сигнала, вычислении обратного инверсного преобразования Фурье от результата перемножения, вычислении квадрата модуля, сравнении полученного отклика квадрата модуля взаимно корреляционной обработки с пороговым напряжением и принятием решения об обнаружении цели и измерении параметров движения цели. При этом согласованная фильтрация осуществляется по всем возможным значениям постоянного и переменного доплеровских параметров, находящихся в банке данных.

Для сокращения вычислительных затрат и упрощения облика приемной системы целесообразно сократить количество параметров банка данных. Из трех оцениваемых параметров наиболее важными являются параметры задержки и доплеровский параметр, поэтому может быть поставлена задача исключения параметра переменного доплеровского эффекта путем использования, в качестве зондирующего посылки сигнала инвариантного к переменному доплеровскому эффекту. Этим сигналом является отрезок тонального импульса $s(t) = \exp(2 \cdot i \cdot \pi \cdot f_0 \cdot t)$, подвергнутый инверсному временному преобразованию $s(t) \to s\left(-\frac{1}{t}\right)$. Таким образом, в среду излучается сигнал с инверсной частотной модуляцией от верхних частот к нижним вида, рассматриваемый относительно его инверсной меры (рис. 1):

$$s(t) = \exp\left(-i \cdot 2 \cdot \pi \cdot s_0 \cdot \frac{1}{t}\right) \cdot rect\left(-\frac{1}{t \cdot T}\right), \qquad (4)$$

где f_0 – начальная частота; T – длительность зондирующего сигнала в естественном масштабе времени

$$rect(t) = \begin{cases} 1, & |t| \le \frac{1}{2}; \\ 0, & |t| > \frac{1}{2}. \end{cases}$$

Покажем, что сигнал (4) инвариантен относительно переменного доплеровского действия β. Действительно, сигнал (4) перепишется:

$$s(t) \to s\left(\frac{t}{\beta t + 1}\right) = \exp\left(i \cdot 2 \cdot \pi \cdot s_0 \cdot \left(-\beta - \frac{1}{t}\right)\right) \cdot rect\left(\frac{1}{T}\left(-\beta - \frac{1}{t}\right)\right).$$

Вычисляя инверсную взаимную корреляционную функцию при малом параметре β, получим:

$$\tilde{r}_{s,s}(t) = \int_{-\infty}^{\infty} s(\tau) \cdot \overline{s\left(\frac{\tau - t}{\tau t}\right)} \frac{d\tau}{\tau^2} = \exp(-i \cdot 2 \cdot \pi \cdot s_0 \cdot \Delta t) \int_{-\infty}^{\infty} rect\left(\frac{1}{T}x\right) \cdot rect\left(\frac{1}{T}(x - \Delta t)\right) dx.$$

Видно, что результатом корреляционного отклика является треугольная функция и при малом параметре β смещением максимума функции можно пренебречь.

Для задания инверсного эталонного сигнала необходимо определить его время начала и окончания. Сначала определим мгновенную частоту инверсного сигнала [9]:

$$f(t) = s_0 / t^2$$

Зададим верхнюю f_w и нижнюю f_n частоты Фурье.

Если задана инверсная частота s_0, то начало t_n сигнала равно $t_n = \sqrt{s_0 / f_w}$, а конец сигнала равен $t_k = \sqrt{s_0 / f_n}$.

Длительность сигнала равна:

$$T_i = \frac{1}{t_n} - \frac{1}{t_k} = \frac{t_k - t_n}{t_k t_n} = \frac{\sqrt{\frac{s_0}{f_n}} - \sqrt{\frac{s_0}{f_w}}}{\sqrt{\frac{s_o^2}{f_n f_w}}} = \frac{\sqrt{f_w} - \sqrt{f_n}}{\sqrt{s_0}}$$

Количество волн инверсного сигнала равно:

$$KW = s_0 T_i = s_0 \left(\frac{t_k - t_n}{t_n t_k}\right)$$

Заметим, что инверсная частота имеет размерность секунды, а инверсное время имеет размерность $1/c$ или [Гц].

Использование сигнала (4) обусловлено тем, что в своем масштабе он является отрезком экспоненциального тонального сигнала. На рисунке 1 приведена временная диаграмма инверсного сигнала.

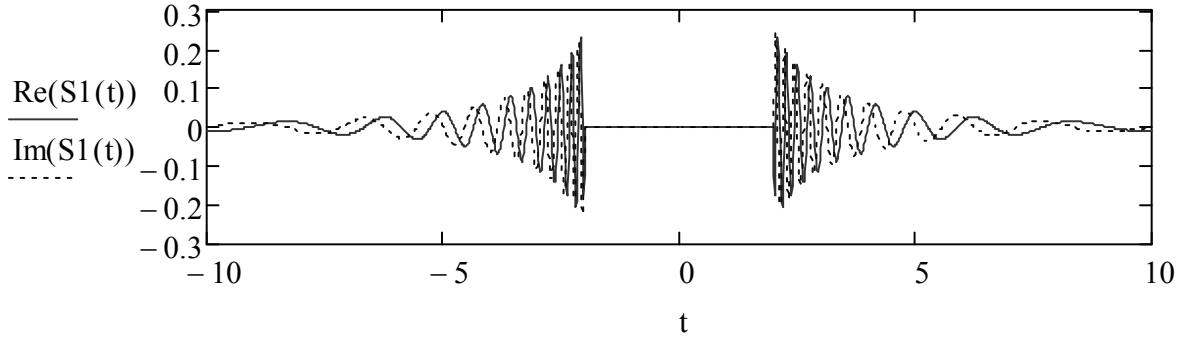

Рис. 1. Временная диаграмма сигнала, промодулированного по инверсному закону

Для проверки достоверности предложенного способа обработки проведено моделирование широкополосной функции неопределенности (ШФН) (3) на ПК в среде MathCad [10]. Максимальное значение ШФН достигается при согласовании параметров эталонного и принимаемого сигнала. Вид ШФН представлен на рисунке 2.

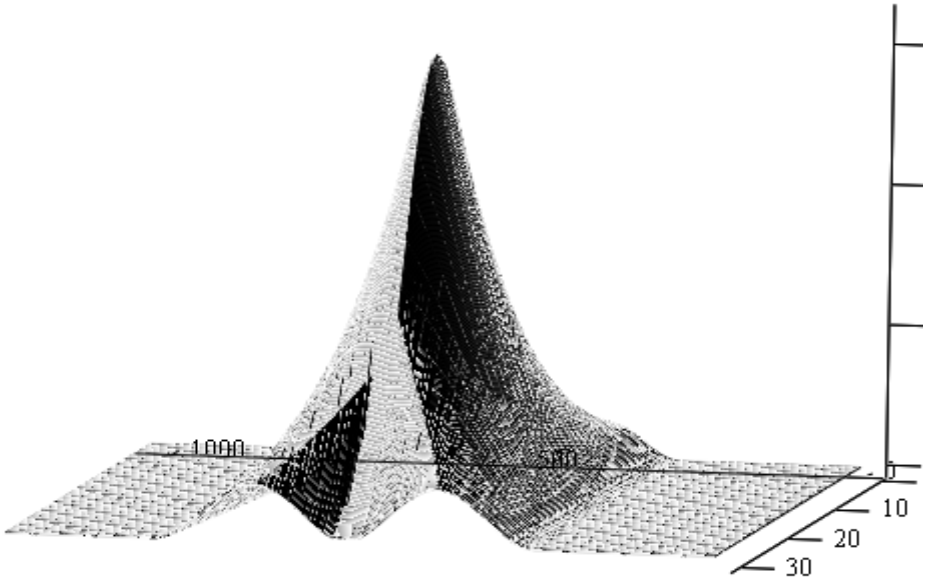

Рис. 2. Вид широкополосной функции неопределенности на ПК в среде MathCad

Из анализа рисунка 2 следует, что инверсный сигнал имеет при комплекснозначном анализе хорошее разрешение по параметру β и инвариантный разрешение при по параметру β при суммировании квадратов (практически инвариантного относительно переменного доплеровского эффекта).

Резюмируя вышеизложенное, можно сделать выводы:

– применение для представления гидроакустических сигналов группы инверсных преобразований времени в условиях локации быстродвижущихся морских животных позволило преодолеть фундаментальные трудности построения процедур согласованной фильтрации. Найдено техническое решение, позволяющее реализовать трехпараметрическую процедуру согласованной фильтрации;

– применение инверсного представления зондирующего сигнала, позволило сократить размерность процедуры согласованной фильтрации с 3-х параметрической до 2-х параметрической. Другими словами, при использовании данного типа сигнала в системах гидролокации можно ограничиться перебором по параметру задержки τ и доплеровскому параметру α.

Библиографический список

1. **Бурдик, В. С.** Анализ гидроакустических систем. – Л. : Судостроение, 1988. – 392 с.

2. **Сапрыкин, В. А., Пронин, Л. Н.** Природа широкополосной функции отклика (ШПФО) // Научно-техническая конференция «Развитие и совершенствование методов освещения и эксплуатации комплексов радиоэлектронных средств». – Петродворец, 1987. – С. 13–15.

3. **Сапрыкин, В. А., Пронин, Л. Н.** Применение гармонического анализа для описания доплеровских и сдвиговых преобразований сигнала. Модели, алгоритмы, принятие решений. – М., 1985. – С. 67–71.

4. **Сапрыкин, А. В.** Корреляционный анализ групповых сигналов // XIV Межвузовская научно-техническая конференция «Военная радиоэлектроника: Опыт использования и проблемы, подготовка специалистов». – Петродворец : ВМИРЭ, 2004. – С. 10–16.

5. **Пат. 2040010** Российская Федерация, МПК^{6} G01S 15/00. Способ определения скорости движущегося судна относительно дна / Сапрыкин В. А., Одинцов Е. Н., Павликов С. Н. ; заявители Сапрыкин В. А., Одинцов Е. Н., Павликов С. Н. и патентообладатель Павликов С. Н. – № 5035574/09 ; заявл. 02.04.1992; опубл. 20.07.1995.

6. **Пат. 2042152** Российская Федерация, МПК^{6} G01S 15/00. Способ определения скорости движущегося судна относительно дна / Сапрыкин В. А., Павликов С. Н., Убанкин Е. И., Артамонов О. А. ; заявители Сапрыкин В. А., Павликов С. Н., Убанкин Е. И., Артамонов О. А. и патентообладатель Павликов С. Н. – № 92012391/22 ; заявл. 16.12.1992; опубл. 20.08.1995.

7. **Пат. 2293997** Российская Федерация, МПК G01S 13/06. Способ корреляционной обработки сигналов, отраженных от быстродвигающихся целей / Сапрыкин В. А., Яковлев А. И., Сапрыкин А. В., Бескин Д. А. ; патентообладатель Военно-морской институт радиоэлектроники им. А. С. Попова. – № 2005128998/09 ; заявл. 13.09.2005; опубл. 20.02.2007, Бюл. № 5. – 17 с.

8. **Пат. 2487367** Российская Федерация, МПК G01S 15/00; G01S 13/00. Способ и устройства быстрого вычисления функции неопределенности сигнала с учетом реверберационной помехи / Сапрыкин А. В., Ковалевский Н. Г., Бескин Д. А. , Блынский А. А. ; патентообладатель Федеральное государственное казенное военное образовательное учреждение высшего профессионального образования «Военный учебно-научный центр Военно-Морского Флота «Военно-морская академия имени Адмирала флота Советского Союза Н.Г. Кузнецова». – № 2009122422/28 ; заявл. 15.06.2009; опубл. 10.07.2013, Бюл. № 19. – 11 с.

9. **Пат. 2467350** Российская Федерация, МПК G01S 15/00; G01S 17/00. Способ и устройство обнаружения сигналов при наличии пере-

менного доплеровского эффекта / Сапрыкин А. В., Быков С. Ф., Блынский А. А. ; патентообладатель Федеральное государственное казенное военное образовательное учреждение высшего профессионального образования «Военный учебно-научный центр Военно-Морского Флота «Военно-морская академия имени Адмирала флота Советского Союза Н.Г. Кузнецова». – № 2009122420/28 ; заявл. 15.06.2009; опубл. 20.11.2012, Бюл. № 32. – 12 с.

10. **Пат. 2467383** Российская Федерация, МПК G06F 17/10; G06N 7/00. Способ и устройство прогнозирования нестационарного временного ряда / Доценко С. М., Сапрыкин А. В., Магон А. Я., Яковлев А. И. ; патентообладатель Министерство обороны Российской Федерации Федеральное государственное казенное военное образовательное учреждение высшего профессионального образования «Военный учебно-научный Центр Военно-Морского Флота «Военно-Морская академия имени Адмирала Флота Советского Союза Н. Г. Кузнецова». – № 2009122419/08 ; заявл. 15.06.2009; опубл. 20.11.2012, Бюл. № 32. – 12 с.

Аннотация

В условиях сложной взаимной кинематики носителя ГАС и объектов биологической природы на основе теоретико-группового подхода разработана новая технология обработки гидролокационных сигналов. Использование технологии позволяет решить актуальную для технической гидроакустики задачу обнаружения и определения, кинематических характеристик биологических объектов. В основу технологии положены свойства групп линейных и инверсных преобразований времени, которые определяют их теоретическую базу. Сформированы процедуры согласованной трехпараметрической фильтрации, позволившие приблизить эффективность приемной системы к своему потенциальному значению по помехоустойчивости.

V. A. Saprykin

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

M. B. Gladkikh

CJSC "TELROS"

NEW TECHNOLOGY OF PROCESSING OF THE SONAR SIGNALS REFLECTED FROM FAST-MOVING OBJECTS OF THE BIOLOGICAL NATURE

Annotation

In the conditions of the composite relative kinematics of sonar and objects of the biological nature on the basis of group-theoretic approach the new technology of processing of sonar signals is developed. Use of technology allows to solve a problem of detection actual for a technical acoustics defining motion characteristics of biological objects. Properties of groups of the linear and inverse transformations of time which define their theoretical base are the basis for technology. In paper are created procedures of the coordinated three-parametrical filtration, allowed to approach effectiveness of receiving system to the potential value on noise stability.

Key words: hyperbolic FM signal, correlation response, noise.

References

1. **Burdik, V. S.** Analiz gidroakusticheskih sistem. – L. : Sudostroenie, 1988. – 392 s.

2. **Saprykin, V. A., Pronin, L. N.** Priroda shirokopolosnoj funkcii otklika (ShPFO) // Nauchno-tehnicheskaja konferencija «Razvitie i sovershenstvovanie metodov osveshhenija i jekspluatacii kompleksov radiojelektronnyh sredstv». – Petrodvorec, 1987. – S. 13–15.

3. **Saprykin, V. A., Pronin, L. N.** Primenenie garmonicheskogo analiza dlja opisanija doplerovskih i sdvigovyh preobrazovanij signala. Modeli, algoritmy, prinjatie reshenij. – M., 1985. – S. 67–71.

4. **Saprykin, A. V.** Korreljacionnyj analiz gruppovyh signalov // HIV Mezhvuzovskaja nauchno-tehnicheskaja konferencija «Voennaja radiojelektronika: Opyt ispol'zovanija i problemy, podgotovka specialistov». – Petrodvorec : VMIRJe, 2004. – S. 10–16.

5. **Pat. 2040010** Rossijskaja Federacija, MPK^6 G01S 15/00. Sposob opredelenija skorosti dvizhushhegosja sudna otnositel'no dna / Saprykin V. A., Odincov E. N., Pavlikov S. N. ; zajaviteli Saprykin V. A., Odincov E. N., Pavlikov S. N. i patentoobladatel' Pavlikov S. N. – № 5035574/09 ; zajavl. 02.04.1992; opubl. 20.07.1995.

6. **Pat. 2042152** Rossijskaja Federacija, MPK^6 G01S 15/00. Sposob opredelenija skorosti dvizhushhegosja sudna otnositel'no dna / Saprykin V. A., Pavlikov S. N., Ubankin E. I., Artamonov O. A. ; zajaviteli Saprykin V. A., Pavlikov S. N., Ubankin E. I., Artamonov O. A. i patentoobladatel' Pavlikov S. N. – № 92012391/22 ; zajavl. 16.12.1992; opubl. 20.08.1995.

7. **Pat. 2293997** Rossijskaja Federacija, MPK G01S 13/06. Sposob korreljacionnoj obrabotki signalov, otrazhennyh ot bystrodvigaju-shhihsja celej / Saprykin V. A., Ja-

kovlev A. I., Saprykin A. V., Beskin D. A. ; patentoobladatel' Voenno-morskoj institut radiojelektro-niki im. A. S. Popova. – № 2005128998/09 ; zajavl. 13.09.2005; opubl. 20.02.2007, Bjul. № 5. – 17 s.

8. **Pat. 2487367** Rossijskaja Federacija, MPK G01S 15/00; G01S 13/00. Sposob i ustrojstva bystrogo vychislenija funkcii neoprede-lennosti signala s uchetom reverberacionnoj pomehi / Saprykin A. V., Kovalevskij N. G., Beskin D. A. , Blynskij A. A. ; patentoobladatel' Federal'noe gosudarstvennoe kazennoe voennoe obrazovatel'noe uchrezhdenie vysshego professional'nogo obrazovanija «Voennyj uchebno-nauchnyj centr Voenno-Morskogo Flota «Voenno-morskaja akademija imeni Admirala flota Sovetskogo Sojuza N. G. Kuznecova». – № 2009122422/28 ; zajavl. 15.06.2009; opubl. 10.07.2013, Bjul. № 19. – 11 s.

9. **Pat. 2467350** Rossijskaja Federacija, MPK G01S 15/00; G01S 17/00. Sposob i ustrojstvo obnaruzhenija signalov pri nalichii pere-mennogo doplerovskogo jeffekta / Saprykin A. V., Bykov S. F., Blynskij A. A. ; patentoobladatel' Federal'noe gosudarstvennoe kazennoe voennoe obrazovatel'noe uchrezhdenie vysshego professional'nogo obrazovanija «Voennyj uchebno-nauchnyj centr Voenno-Morskogo Flota «Voenno-morskaja akademija imeni Admirala flota Sovetskogo Sojuza N.G. Kuznecova». – № 2009122420/28 ; zajavl. 15.06.2009; opubl. 20.11.2012, Bjul. № 32. – 12 s.

10. **Pat. 2467383** Rossijskaja Federacija, MPK G06F 17/10; G06N 7/00. Sposob i ustrojstvo prognozirovanija nestacionarnogo vremennogo rjada / Docenko S. M., Saprykin A. V., Magon A. Ja., Jakovlev A. I. ; patentoobladatel' Ministerstvo oborony Rossijskoj Federacii Federal'noe gosudarstvennoe kazennoe voennoe obrazovatel'noe uchrezhde-nie vysshego professional'nogo obrazovanija «Voennyj uchebno-nauchnyj Centr Voenno-Morskogo Flota «Voenno-Morskaja akademija imeni Admirala Flota Sovetskogo Sojuza N. G. Kuznecova». – № 2009122419/08 ; zajavl. 15.06.2009; opubl. 20.11.2012, Bjul. № 32. – 12 s.

Сапрыкин Вячеслав Алексеевич – доктор технических наук, профессор, заведующий лабораторией автоматизации информационных свойств акустики Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

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

УДК 004.43; 004.4'6

Ю. В. Скворцов

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

ГИБРИДНАЯ МОДЕЛЬ УПРАВЛЕНИЯ ПАМЯТЬЮ ПРОГРАММ

память, управление, модели, сборка мусора, программы

Терминология

Выделение памяти – операция по получению динамической памяти (в языке *C – malloc)*.

Освобождение памяти/объекта – операция по возвращению выделенной памяти; память помечается как свободная и становится доступной для дальнейших операций выделения (*C free*).

Разрушение объекта – операция по освобождению ресурсов, занимаемых объектом (*C++* деструкторы).

Удаление объекта – операция по разрушению объекта и освобождению памяти (*C++ delete*).

«Ручные» модели

Простейшей моделью работы с памятью программы является, так называемая, «ручная» модель управления. Она состоит в вызове функции выделения памяти, работе с полученной памятью и затем освобождения (например, функцией *free*). Менеджмент этих функций заключается в том, что функция получения просматривает «вверенный» ей объём памяти с поиском непрерывного места, большего или равного запрошенному размеру, после чего найденный участок помечается как занятый; при освобождении – как свободный, и может быть использован снова. Ошибки при этом возникают в случае, если выделенный блок памяти в действительности больше не нужен, но не был освобождён; тогда при выполнении программы доступная оперативная память начнёт исчерпываться, и вскоре её станет слишком мало для дальнейшего продолжения работы, либо этот процесс будет «отъедать» память у других процессов. Такая ошибка называется утечкой памяти. Обратная ситуация возникает, если освободить память слишком рано, пока некоторые объекты продолжают ссылаться на неё; тогда возможна ситуация, когда программа будет читать/писать_в данные либо другого объекта, который получил этот адрес в памяти, либо случайные данные, либо служеб-

ные данные управления памятью, либо произойдёт ошибка чтения, приводящая к срабатыванию механизма защиты памяти и аварийному завершению программы. Ссылки на память, которая была уже удалена, называются «висячими ссылками». Также неправильно работающие приложения могут записывать больше данных, чем запрошенная область. Перезапись приводит к повреждению данных, расположенных после выделенной области, и часто является причиной аварийного завершения программы.

Существует множество схем управления памятью с «ручной» моделью, каждая из которых оптимизирована под различные условия применения. Рассмотрим основные из них.

Оптимизированные под многопоточность. При выделении и освобождении памяти необходимо изменить состояние менеджера памяти; если это будут делать сразу несколько потоков, то возникнет состояние называемое «гонкой условий» (от англ. *race condition*), которое может привести к установке неправильного состояния и последующему сбою. Тривиальное решение, такое как установка мьютекса (от англ. *MUTal EXclusion – mutex*) для обеспечения доступа только одному потоку может значительно снизить быстродействие программы, если сразу несколько потоков выделяют множество памяти. Можно значительно уменьшить эту проблему, если зарезервировать несколько участков памяти за конкретным потоком, который сможет раздавать их без глобальной блокировки, что значительно увеличит скорость работы программы. Недостатком является то, что если подобные меры принимаются тогда, когда они не нужны, то скорость, наоборот, снизится.

Оптимизированные под уменьшение фрагментации. При выделении и освобождении памяти может возникнуть ситуация, при которой занятые блоки чередуются со свободными – это явление называется фрагментацией. Фрагментация памяти снижает быстродействие программы. Так при попытке выделить большой блок памяти тривиальная реализация находит первый свободный блок, больший или равный требуемому размеру. Если при этом будет множество меньших блоков, то программа потратит много времени, выбирая подходящий блок. Проблема заключается в работе кэш-памяти процессора – если данные окажутся «разбросанными» по памяти, то процессор будет простаивать, ожидая выборки из памяти; если же данные идут кучно, то скорость обработки значительно повышается. Схема делит запрашиваемые размеры на несколько блоков, в каждом из которых лежат только фиксированные размеры: маленькие (до 48 Кб) и средние (до 1 Мб). Алгоритм сосредотачивает однотипные элементы рядом, что помогает процессору эффективнее использовать кэш, уменьшая вероятность того, что из-за одного маленького блока памяти не удастся выделить непрерывный большой.

Оптимизированные под поиск ошибок. Обычно содержат одну или несколько возможностей обнаружить ошибку работы с динамической памятью. Возможность указать место вызова позволяет при завершении программы вывести на экран (или в файл) места, где память была выделена, но не была освобождена. Возможность создания пролога/эпилога позволяет обнаружить проблемы связанные с перезаписью выделенного блока. Более «агрессивное» возвращение памяти операционной системе и освобождение страниц виртуальной памяти позволяет быстро обнаружить код, обращающийся к удалённой памяти.

Оптимизированные под использование внутри функций (пулы). Обычно представляют собой большой блок памяти, из которого затем выделяется множество маленьких кусков. Затем, после того как данные обработаны, большой блок освобождается; соответственно, все внутренние блоки становятся «висячими». Часто внутренние блоки нельзя освободить по отдельности – только все вместе. Преимуществом этой схемы является её масштабируемость (один поток работает с пулом, что не требует механизмов ограничения доступа) и скорость (не требуется находить блок запрошенного размера – достаточно просто сдвинуть границу на запрошенное число). Недостатками является принудительное освобождение памяти, возможно, содержащей необходимые данные, что существенно ограничивает применимость этой схемы.

Следует отметить, что типизированные схемы не являются взаимоисключающими и могут комбинироваться.

Модели-«счётчики»

Полностью императивное ручное управление памятью может быть весьма утомительно: необходимо следить, чтобы после выхода из функции вся ставшая ненужной память была удалена, а также инициализировать её в сложных объектах. Часто эту работу могут выполнить «умные указатели» (от англ. s*mart pointers*). Они автоматически инициализируются пустым значением (если не передать им указатель при инициализации), освобождают память после покидания область видимости (*scope*), а если понадобится освободить память раньше, то они примут пустое значение, избегая появления «висячих указателей».

Но одних «умных указателей» и схем управления памятью может быть недостаточно для правильного определения необходимости удалить объект и освободить память, так как один объект может использоваться множеством других объектов. Самым простым решением является введения счётчика использований объекта (или счётчика ссылок): каждый используемый контекст должен увеличить его, а при его покидании области видимости – уменьшить. Когда счётчик ссылок достигнет нуля, то объект удаляется. Эту модель называют управлением памятью со счётчиком ссылок (от англ. r*eference counting*). Язык программирова-

ния *Objective-C* умеет отслеживать передачу различных объектов и добавлять/уменьшать счётчик использования объектов автоматически, используя механизм автоматического подсчёта ссылок (от англ. *automatic reference counting – ARC).*

Однако подсчёт ссылок работает лишь, если объекты не имеют циклической зависимости. Приведём пример – есть структура данных, содержащая множество подобных структур, а также ссылку на содержащую её структуру (родителя):

```
object Foo {
 Foo[] Children;
 Foo Parent;
}
```

При использовании счётчика ссылок окажется, что если имеется хотя бы один дочерний объект, то вся структура данных никогда не будет освобождена, так как дочерний объект увеличивает счётчик использований родителя, а родитель – дочерних объектов. Для решения подобной зацикленности часто применяют механизм использования «слабых ссылок», не увеличивающих счётчик при копировании; а если объект, на который ссылается «слабая ссылка», был разрушен, то «слабая ссылка» станет пустым указателем (*null*). *Cocoa Framework* [1] рекомендует использовать «слабые ссылки» для обращения к родителю, и «сильные» для обращения к дочерним объектам. Но если запомнить определённый объект, и через некоторое время потребуется получить доступ к родительскому элементу, который уже удалён, то слабый указатель будет пустым, что может привести (при дальнейшей обработке данных) к неверному результату.

В языке *Objective-C* с *ARC* для разрешения ситуации раннего удаления применяется объект *Autorelease Pool,* в который можно добавить объекты путём посылки им сообщения *autorelease*, а затем, когда данные будут посчитаны, можно уменьшить число использований всех попавших в пул объектов. Этот механизм позволяет упростить ведение ссылок при решении некоторых задач, но в приведённом выше примере он эквивалентен простому хранению ссылки на корень, а в более сложном случае приводит к большому накоплению памяти для последующего удаления и фрагментации.

В комплексных задачах, например, таких как высокоуровневое представление реляционных таблиц в памяти (*Object Relational Mapping*), где внешний ключ заменяется ссылкой на объект /1:1/ или на массив /1:M/, управление памятью объектов будет очень сложным [2].

Сборщики мусора

Если бы память компьютера была бесконечной, можно было бы просто оставлять в ней ненужные объекты. Сборка мусора – это эмуляция такого бесконечного компьютера на конечной памяти. Впервые модель автоматического управления памятью, называемая «сборкой мусора» (от англ. *garbage collector* – *GC*), была применена в языке Лисп. Многие ограничения сборщиков мусора (нет гарантии, что объект будет разрушен; управляет только памятью, но не другими ресурсами) вытекают из этой метафоры. Принцип работы заключается в том, что все глобальные переменные и переменные на стеке являются сильными, из них высчитываются достижимые объекты, достижимые у достижимых, и т. д. Объекты, которые так и не были достигнуты, удаляются. В приведённом примере с классом *Foo* не имеет значения, что объекты имеют циклические ссылки друг на друга – если ни один из них не имеет ссылки с достигнутого объекта, то все они будут удалены. Иногда этот алгоритм называют «консервативным», поскольку он делает предположение о том, что определённый объект можно удалить, если им невозможно воспользоваться [3]. «Агрессивный» *GC* (в противовес консервативному) может удалить даже достижимый объект, если эвристически посчитает, что им не воспользуются или при необходимости пересоздадут.

Сборка мусора решает проблему утечек памяти и висячих указателей, упрощая при этом процесс программирования. Тем не менее, она позволяет полностью игнорировать вопрос выделения и освобождения ресурсов: некоторые объекты, такие как файлы (если не закрыты специальным образом) будут недоступны для повторного открытия до тех пор, пока не будет запущен сборщик мусора, и он решит разрушить объект. Блоки памяти, даже если на них больше нет ссылок, удаляются не сразу, а «когда-нибудь», поэтому вместо повторного использования выделяется новый блок, что может быть неэффективно при использовании в цикле. Другим недостатком *GC* является необходимость заблокировать работу программы на момент сборки мусора. Если свободной памяти остаётся мало или в программе имеется множество объектов, то программа «замирает» на достаточное длительное время, что не позволяет выполнять алгоритмы, которые требуют работы в реальном времени, такие как показ графики и видео, вывод звука, и другие [4]. Некоторые алгоритмы, напротив, начинают работать быстрее при отложенном удалении объектов: главный поток перестаёт тратить такты на удаление и сосредотачивается на вычислениях. Ещё больше увеличить быстродействие помогает тривиальное присвоение объектов (в «считающей» модели приходится при каждом использовании объекта «крутить» счётчик).

Существует две схемы работы *GC*: с перемещаемой и неперемещаемой памятью. В неперемещающем *GC* адрес объекта остаётся в памяти постоянным. При перемещающем *GC* достижимые объекты могут изменять свой адрес в памяти так, чтобы избегать фрагментации. Хотя перемещающий алгоритм может казаться длительным, но в целом он позволяет увеличить быстродействие программы за счёт оптимизации работы кэша процессора и простого выделения памяти (для которого, подобно пулу, надо просто сдвинуть границу свободной и занятой областей). Обычно языки, в которых используется перемещающий алгоритм, не позволяют производить арифметику над адресами, а сами объекты, передаваемые в другое окружение, обычно «закрепляются» (от англ. *pin*), сигнализируя сборщику о невозможности переместить объект. Перемещающая реализация практически нереализуема в тех языках, которые не имеют встроенного механизма сборки мусора.

Как видно из вышеизложенного, каждый подход к управлению памятью программ имеет свои преимущества и недостатки. Языки *D* и *Objective-C* позволяют выбрать модель управления памятью, при этом все библиотеки могут её использовать. Хотя гипотетически это позволяет выбрать наиболее оптимальный способ решения задачи, оказывается, что некоторые библиотеки поддерживают только одну из моделей.

Попробуем синтезировать такую модель, которая объединила бы достоинства рассмотренных моделей, и назовём её гибридной.

Гибридная модель

Сначала определим требования к гибридной модели. Во-первых, потребуется детерминированное разрушение – это означает, что если на объект точно никто не ссылается, то его следует разрушить. Во-вторых, понадобится отслеживание циклических ссылок. Объекты могут иметь циклические связи, при этом, если ни один из достижимых объектов на них не ссылается, то они будут удалены «когда-нибудь». В-третьих, желательно отсутствие глобальной остановки на момент сборки мусора. В-четвёртых, должна существовать возможность использования модели из «неприспособленных» языков, таких как *C++*.

Чтобы точно знать время, когда объект станет гарантировано не нужен, проще всего воспользоваться счетчиком ссылок аналогично тому, как он применяется в соответствующей модели. Различие между моделями управления памяти заключается в способе подсчёта ссылок: если в первой счётчик всего один, то в гибридной модели их два. Один считает количество использований объекта на стеке и как глобальной переменной – считает «сильные ссылки»; другой считает число использований объекта внутри других объектов – назовём такие ссылки «внут-

ренними» (*member reference*). Если оба счётчика равны нулю, то объект гарантировано не используется и его можно удалить.

Введём правила увеличения/уменьшения счётчиков.

1) Если объект должен использоваться внутри функции, то следует увеличить число сильных ссылок на этот объект, чтобы гарантировать, что во время использования он не будет разрушен. При создании объекта число сильных ссылок равняется единице. При выходе из области видимости счётчик сильных ссылок уменьшается на единицу.

2) Если на объект ссылается другой объект, то следует увеличить число внутренних ссылок на этот объект. При присваивании другого объекта или разрушении хранящего объекта внутренний счётчик объекта члена класса следует уменьшить на единицу.

Сборка мусора осуществляется отдельным потоком немного модифицированным алгоритмом «отметок и зачистки» (от англ. *mark and sweep – MaS*). В начале процесса сборки мусора, начинающегося обычно после создания определённого количества объектов или занятия определенного объёма оперативной памяти, объекты, имеющие «сильные ссылки», помечаются как достижимые. Затем достижимыми помечаются все вложенные объекты, и так до тех пор, пока не будут обработаны все объекты. После этого все недостигнутые объекты удаляются. При отсутствии блокировки возможна следующая ситуация.

Пусть `object A { A Child; }`, тогда:

Сборщик мусора обрабатывает объект *A*, у которого ссылка на *Child* пока является пустым указателем. Пользовательский поток в это время присваивает объекту *A* дочерний объект, создав его. Согласно правилам подсчёта ссылок, после создания у дочернего объекта будет одна «сильная ссылка», после присвоения – одна «сильная» и одна «внутренняя»; затем, когда поток выйдет из области видимости создания объекта, у него будет одна «внутренняя ссылка».

Наивная реализация посчитает, что раз за время обхода он не был отмечен как достижимый, то его «держит» другой недостижимый объект, после чего удалит его. Исправить ситуацию можно, дав знать сборщику мусора, что объект присваивался с начала сборки мусора, то есть при каждом присваивании (и в сильном и во внутреннем) объекта флаг использования объекта выставляется. При старте сборщик должен сбросить флаги для всех объектов.

Наличие этого флага добавляет ещё один шаг к алгоритму *MaS*: после проверки достижимости всех объектов выделяются потенциально недостижимые. У каждого недостижимого объекта проверяется флаг присваивания после начала сборки мусора. Если флаг установлен, то объект считается достижимым и все его внутренние ссылки также становятся достижимыми, после чего они исключаются из списка по-

тенциально недостижимых. Если произошёл хотя бы один возврат, то цикл повторяется снова. В результате останутся только гарантированно недостижимые объекты.

Возможна ситуация, когда при обходе всех объектов сборщиком мусора просматриваемый объект удалит другой поток (*GC* не увеличивает счётчик использований), поскольку он стал гарантировано не использоваться – тогда сборщик мусора обратится к данным только что удалённого объекта. Разрешение этой проблемы постановкой эксклюзивного доступа приведёт к тому, что другие потоки не смогут ни выделить, ни освободить блоки памяти во время сборки мусора, что приведёт к их простою. Отсутствие простоя и хорошее быстродействие можно обеспечить, не удаляя объект, а лишь уничтожая (память останется выделенной и доступной для чтения и записи, а все ресурсы, например, соединение с базой данных, будут освобождены), а память будет освобождена сборщиком.

Программная реализация

Чтобы неподготовленные языки могли воспользоваться преимуществами автоматической сборки мусора необходимо, чтобы каждый объект имел способ указать сборщику, какие объекты он может достигнуть, а желательно имел эффективные средства для организации подсчёта ссылок (возможность написания «умных» указателей). В таких языках гибридная модель будет реализована поверх выбранной схемы памяти. Учитывая особенности работы *GC,* рекомендуется выбирать схему, оптимизированную под многопоточность и уменьшение фрагментации.

В языке *C++* базовый класс с поддержкой сборки мусора принимает следующий вид:

```
class Collectable {
 public:
      virtual void PingReachable(class GC*) {};
      virtual ~Collectable() {}
     Collectable();
 private:
     void *operator new(size_t);
     void *operator new(size_t, std::nothrow_t) = delete;
     void *operator new[](size_t) = delete;
     void operator delete(void *);
     void operator delete[](void *) = delete;

     std::atomic<int> _strongCount;
     std::atomic<int> _membersCount;
     enum : short {
          NotMarked,
          Marked,
```

```
        Checked
    } _reachFlag;
    bool _wasAssigned;
    bool _suppressDtor;

    friend class GC;
};
```

Сборщик мусора должен иметь возможность обойти все объекты – некоторые схемы управления памяти позволяют обойти все выделенные объекты (например, *Heap HeapWalk*), тогда следует воспользоваться этой возможностью. Если такой возможности нет (как, например, у *C malloc*), то следует связать объекты двунаправленным списком.

Этим классом (и его наследниками) помогают управлять умные указатели *Strong* и *Member* (считающие сильные и внутренние ссылки соответственно). Рассмотрим следующий пример:

```
1: { Strong<ORM_Table> q = new ORM_Table(); }
2: { Strong<ORM_Table> t = this; }
```

В первой строке создаётся новый объект, который сразу будет разрушен (так как сразу покинет область видимости). Во второй строке этот объект инициализируется локальной переменной, которая никак не может быть защищена умным указателем от неправильного применения. Чтобы избежать подобных ошибок, оператор выделения памяти (*new)* для объектов *Collectable* (и его наследников) доступен только объекту *GC*. Для выделения памяти и конструирования объекта следует использовать функцию *GC::New<Tun>(параметры конструктора)*:

```
class GC {
    template <typename T, typename... Args>
    static Strong<T> &&New(Args&&... args) {
        Strong<T> result(new T(args...));
        return std::move(result);
    }
    ...
};
```

Такая схема работы аналогична *C++* функции *make_shared*. Это позволяет убрать двусмысленность при инициализации «умных указателей»: теперь при получении обыкновенного указателя *Strong* и *Member* увеличивают количество ссылок, а не просто инициализируются ими.

Минусом является сложность передачи объекта по ссылке (*ref object*), так как принимающая функция не сможет определить тип переданного аргумента (сильная или внутренняя ссылка), а передача дополнительного параметра и проверка его перед присваиванием очень не-

удобное решение. С другой стороны, в *Java* нельзя передавать объекты по ссылке, но это не сильно снижает удобство от её использования.

Результаты сравнения

Опираясь на вышеизложенное, проведём сравнение функционала моделей и результаты представим в табличном виде, используя следующие обозначения: «+» – функционал присутствует, «–» – функционал отсутствует, «§» – имеющийся функционал неудобен для использования.

ТАБЛИЦА. Сравнение функционала моделей

Функционал	**Модель управления памятью**			
	«Ручная»	**Счётчик**	***GC***	**Гибридная**
Детерминированное удаление	+	+	–	+
Использование во множестве объектов	§	+	+	+
Удаление циклически связанных объектов	§	§	+	+
Управление памятью без глобальной блокировки	+	+	–	+
Использование из любых языков	+	+	–	+

Как видно из таблицы, предложенная гибридная модель обладает наибольшим функционалом, который позволяет использовать её даже в сложных системах реального времени, чего не дают делать традиционные модели. А использование существующих схем управления памятью программ даёт возможность оптимальной реализации модели согласно предъявляемым требованиям.

Библиографический список

1. **Scott Anguish**, Erik M. Buck, Donald A. Yacktman. Cocoa Programming. Sams Publishing, 1st Edition 2002, Paperback. - ISBN 0-672-32230-7.

2. **E. D. Berger**, B. G. Zorn, K. S. McKinley (2001). Composing high-performance memory allocators. ACM SIGPLAN Notices 36 (5): 114–124. doi:10.1145/381694.

3. **Richard Jones**, Antony Hosking, Eliot Moss, (19 August 2011). The Garbage Collection Handbook: The Art of Automatic Memory Management. CRC Applied Algorithms and Data Structures Series. Chapman and Hall/CRC. - ISBN 1-4200-8279-5.

4. **Richard Jones**, Rafael D. Lins (1996). Garbage Collection: Algorithms for Automatic Dynamic Memory Management. Wiley. - ISBN 0-471-94148-4.

Аннотация

В статье рассматриваются традиционные модели управления динамической памятью программ. Приводятся схемы их использования, преимущества и недостатки. Предлагается синтезировать на их основе некую гибридную модель, которая сможет детерминировано разрушать неиспользуемые объекты в памяти и которую можно реализовать в языках программирования без поддержки механизма сборки мусора, например C++.

Y. V. Skvortsov

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

HYBRID MANAGEMENT MODEL OF PROGRAM BY MEMORY

Annotation

The article is about the traditional dynamic memory management model of programs. Their schemes of use, advantages and disadvantages are being looked into. It is proposed to synthesize, on their basis, a kind of hybrid model that can deterministically destroy the unused objects in memory, and which can be implemented in a programming language without the support mechanism for garbage collection, such as C+ +.

Keywords: memory, management, model, garbage collector, program.

Скворцов Юрий Владимирович – аспирант кафедры Защищенных сетей связи Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», yuriy709@gmail.com

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.056

Д. Н. Шакин

Управление Федеральной службы по техническому и экспортному контролю (ФСТЭК России) по Северо-Западному федеральному округу

ЭСКИЗ СИСТЕМНОГО ПОДХОДА К ОПРЕДЕЛЕНИЮ СУЩНОСТИ И СОДЕРЖАНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

информационная безопасность, системный подход, сущность, содержание, защищенность информационных систем, информационно-техническая безопасность, психофизическая безопасность

Правилом современного научного исследования стало рассмотрение объектов, предметов, процессов и явлений с точки зрения системного подхода, т. е. комплексно, многоаспектно, во всей совокупности составляющих их элементов, связей и отношений. Системный подход дает возможность рассматривать любую организацию как единое целое, с целью достижения наибольшей эффективности функционирования всей системы, несмотря на наличие у ее элементов противоречивых стремлений [1].

По мнению автора, это утверждение в полной мере относится и к такой специфической предметной области теории и практики, как информационная безопасность.

С развитием глобального информационного общества увеличилось число комплексных проблем информационной безопасности, требующих участия в их решении специалистов различных областей знаний. Научные направления, занимающиеся исследованием проблем информационной безопасности, развиваясь на различной прикладной и теоретической основе, используют сходные понятия с неоднозначной трактовкой. Объективно возникла необходимость системного представления современной терминологии информационной безопасности, с учетом исторического опыты, современных реалий и результатов научного прогнозирования развития данной области знаний.

Как считает автор, начать следует с уточнения сущности и содержания информационной безопасности.

Определение сущности информационной безопасности

Под сущностью, как философской категорией, автор понимает атрибут, внутреннее содержание, постоянное основное свойство предмета, явления, процесса, без которого она немыслима.

Необходимо отметить, что, несмотря на множество разнообразных определений сущности, все они содержательно сводятся к тому, что под сущностью понимается нечто, определяющее данный вещь предмет или явление, процесс, то, что «делает эту вещь именно данной вещью» [2]. Сущность, прежде всего, заключается в целевом предназначении и в этом отношении автор считает показательным известное определение понятия «система»: *«Система – это целевая сущность»* [3].

Необходимо отметить, что категории «безопасность» и «опасность» являются диалектически взаимосвязанными характеристиками объективной реальности, отражающими ее противоречивые стороны.

История становления и развития информационной безопасности неразрывно связано с единством и борьбой противоположностей: новых технологий, новых общественных отношений и новых информационных опасностей.

Известны *три* основных аспекта этого единства и борьбы противоположностей: информационный, информационно-психологический и информационно-технический [4].

Информационный аспект

Информационный аспект предусматривает систему технических мер и организационных мероприятий, направленных на противодействие совершению неправомерных действий в отношении непосредственно самой информации.

Информационно-технический аспект

Информационно-технический аспект предусматривает устранение или ослабление деструктивного информационного воздействия на средства добывания, переработки и передачи информации, а также комплекс мер и мероприятий по защите от средств добывания информации противоборствующей стороны.

Информационно-психологический аспект предусматривает сбор и обобщение данных об источниках деструктивного информационного воздействия, предупреждение подобного воздействия, его срыв, нейтрализацию и ликвидацию последствий деструктивного воздействия.

В рамках рассмотренных аспектов, по мнению автора, и следуют трактовать *сущность информационной безопасности*, состоящую в превентивной защите от комплекса угроз деструктивного информационного воздействия (информационных опасностей).

При определении сущности и содержания информационной безопасности автор рассматривает эту категорию в широком и узком смысле ее понимания.

В широком смысле информационная безопасность представляет собой важнейшую составную часть национальной безопасности и включает в себя борьбу в информационной сфере.

В узком смысле информационная безопасность рассматриваться как составная часть борьбы в информационной сфере, выражающая функции защиты.

Определение содержания информационной безопасности

Понятие категории «сущность» родственно понятию категории «содержание». Содержание характеризует различные стороны сущности. Для определения содержания необходимо представить явление, процесс или предмет с одной стороны как единое целое, а с другой стороны, как совокупность отдельных элементов – составных частей [2].

По мнению автора, содержание информационной безопасности можно представить *двумя* составными частями: информационно-технической безопасностью и психофизической безопасностью.

Информационно-техническая безопасность

Информационно-техническая безопасность включает в себя защищенность информационных систем добывания, переработки и передачи информации от деструктивного информационного воздействия и обеспечивается защищенностью элементной базы радиоэлектронных средств и систем различного назначения, а также алгоритмов их функционирования. Следовательно, необходимо говорить о защищенности систем добывания, переработки и передачи информации от радиоэлектронного и программно-технического деструктивного воздействия.

Для систем добывания информации основными угрозами являются: несанкционированный доступ к конфиденциальной информации, утечка информации и деструктивное воздействие путем создания радиоэлектронных помех или физического уничтожения (функционального поражения) поражения элементов радиоэлектронных средств и систем.

Кроме этого воздействие на системы добывания информации может осуществляться снижением радиоэлектронной заметности и оптико-электронной контрастности объектов за счет изменения их отражающих (рассеивающих) и поглощающих свойств, а также проведением специальных организационных мероприятий. Снижение заметности достигается применением специальных покрытий, выбором мало отражающих форм объектов, использованием радиопрозрачных материалов и проведением ряда организационных мероприятий, не требующих использова-

ния специальных средств: например, выбором высоты полета летательных аппаратов или глубины маневрирования подводных объектов. Еще одним направлением воздействия на системы добывания информации является изменение условий распространения электромагнитной энергии путем ионизации слоев ионосферы или созданием искусственных образований с целью изменения свойств среды.

Это направление деструктивного воздействия представляет угрозу и для систем передачи информации.

Основными угрозами для систем переработки информации кроме несанкционированного доступа к источникам информации, разглашения и утечки информации по техническим каналам является дезорганизация функционирования технических средств переработки информации путем их силового (энергетического) или программно-технического подавления.

Сам факт существования информационно-технической безопасности в качестве элемента информационной безопасности не вызывает сомнений, а ее содержание довольно подробно описано в специальной литературе.

Вместе с этим жизнедеятельность социума реализуется не только в физической среде, но и в виртуальном мире искусственно созданным человеком. Следовательно, всю информационную сферу условно можно разделить на две основные составные части: информационно – техническую сферу (искусственно созданный человеком мир техники, технологий и т. п.) и информационно-психологическую сферу (естественный мир живой природы, включающий и самого человека).

По компонентам информационной сферы классифицируются и угрозы информационной безопасности, как угрозы не только в информационно-технической, но и в информационно-психологической сфере. Исходя из этой гипотезы, целесообразно включить в содержание информационной безопасности элемент психофизической безопасности.

Информационно-психологическая сфера является частью информационной сферы. Она образуется совокупностью людей, информации, которой они обмениваются и которую воспринимают, а также общественными отношениями, возникающими в связи с информационным обменом (информационными коммуникациями).

Психофизическая безопасность

По мнению автора, психофизическая безопасность включает информационно-психологическую защищенность сознания и защищенность органов и систем человека и (или) групп людей от энергоинформационного воздействия.

Энергоинформационные воздействия – это преднамеренное или непреднамеренное воздействие физических и (или) психических факторов

информационной и энергетической природы на психические и физиологические процессы в организме человека [4].

По источнику происхождения энергоинформационные воздействия, делятся на естественные, осуществляющие непреднамеренные воздействия и искусственные, могущие воздействовать как преднамеренно, так и непреднамеренно.

К естественным энергоинформационным воздействиям относят:

– космические воздействия (гравитационное излучение, энергоинформационное поле и др.);

– взаимодействия в биосфере (микроорганизмы – растения – животные – человек – человек);

– воздействие земли (геопатогенные зоны, магнитное поле земли, атмосферные явления, сейсмические явления).

К искусственным энергоинформационным воздействиям относят:

– техногенные воздействия (акустические, электромагнитные и ионизирующие излучения и др.);

– антропогенные воздействия (этнические, военные и другие социальные конфликты, социально-психологическая напряженность, психоэмоциональные всплески, психофизическое воздействие и др.).

К числу источников энергоинформационного воздействия можно отнести: специальные технические средства, средства массовой информации, аномальные энергетические зоны и др.

По мощности энергоинформационные воздействия можно подразделить на сверхмощные и слабые энергетические воздействия, модулированные смысловыми сигналами, т. е. информацией.

Слабые энергетические воздействия, модулированные смысловыми сигналами – информацией, представляют угрозу информационно-психологической сфере, а механизм их воздействия на сознание достаточно часто является предметом научных исследований.

Однако, перефразируя известное утверждение Н. Винера [5] можно утверждать, что важно не только то, чтобы органы чувств человека, находились под надлежащим контролем сознания, о защищенности которого мы говорили выше, но и важно обеспечить защиту самих органов чувств – эффекторов – от вредного воздействия.

В течение последних десятилетий опубликован ряд работ, посвященных механизму воздействия сверхмощного (более 1 ГВт) электромагнитного излучения крайне высоких частот (от единиц до сотен ГГц) на биологические объекты различного уровня организации, от отдельных клеточных компонентов, изолированных клеток и микроорганизмов до организмов животного и человека [6].

Электромагнитное излучение оказывает биологическое воздействие, связанное с нагревом организма, вызывающим хромосомные и генетические изменения, активацию вирусов, изменение иммунологи-

ческих и поведенческих реакций, что может привести к неадекватным действиям и даже к смертельному исходу. Другими последствиями облучения могут быть временное нарушение активности нервных клеток, возникновение слуховых галлюцинаций, а при более высокой плотности энергии – повреждение тканей слуховых органов [6].

Наиболее чувствительны к действию электромагнитного излучения нервная и эндокринная системы организма человека.

В качестве примера тесной взаимосвязи информационно-технической и информационно-психологической сферы следует привести оценку воздействия оружия функционального поражения.

Под оружием функционального поражения автор понимает технические средства деструктивного силового воздействия на радиоэлектронные системы с целью вывода из строя («выжигания») их элементной базы. Применяемые для этих целей источники сверхмощного и сверхвысокочастотного электромагнитного излучения, оказывают побочное воздействие и на организм человека. В классификации видов вооружения такое оружие относят к классу оружия нелетального действия, т. е. условно говоря, не приводящим к людским потерям, однако могущего создавать факторы и условия, приводящие к гибели биологических организмов. Мощное энергоинформационное воздействия на органы и системы, обеспечивающие жизнедеятельность человека, как отдельного индивида, так и групп людей, представляет угрозу их стабильному существованию и развитию. Природа деструктивного энергоинформационного воздействия может быть основана и на явлениях биологического резонанса в диапазоне частот, близких к биологическим ритмам физиологических процессов, а также резонанса с физико-химическими явлениями в организме человека.

В качестве средств деструктивного воздействия на человека могут использоваться средства, генерирующие модулированные сверхвысокочастотные, лазерные, ультразвуковые и другие колебания. При определенной плотности мощности и частоте излучения у человека могут возникать звуковые ощущения. Если такое излучение модулировать голосом, то облучаемый человек будет «слышать» навязываемую ему, таким образом, информацию.

Ряд исследователей не без основания считают, что существует реальная возможность создать принципиально новый вид оружия, основанный на синтезе слабого и мощного воздействия информационной и энергетической природы [4, 6].

Из сказанного следует, что было бы ошибочно ограничивать определение содержания информационной безопасности только рамками информационно-технической защищенности информационных систем. Исходя из приведенных рассуждений, автор считает, что содержание информационной безопасности должно быть расширено за счет включе-

ния в него, кроме безопасности технических информационных систем – безопасности биологических информационных систем, а именно психофизической безопасности человека.

Таким образом, системное представление содержания информационной безопасности, приведенное на рисунке, видится состоящей из двух частей: информационно-технической безопасности систем добывания, переработки и передачи информации и психофизической безопасности индивидуального, группового, общественного сознания, а также органов и систем человека.

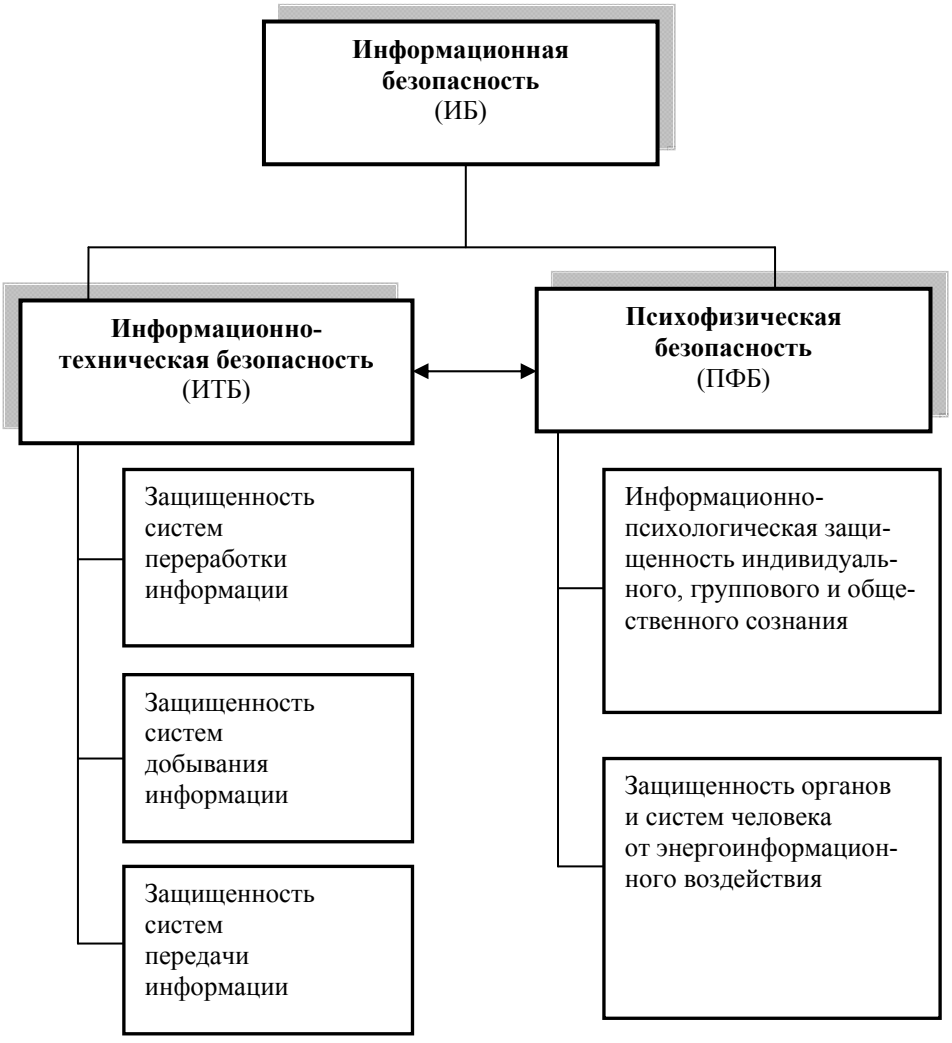

Рисунок. Системное представление содержания информационной безопасности

Современное общество вступило в постиндустриальный период своего развития, который по сути своей назван информационным. На новом этапе истории мира, когда возможности экстенсивного роста за счет механического присоединения новых ресурсов путем вооруженного захвата территории других стран и всех имеющихся на этой территории богатств оказались неэффективными, встал вопрос о формах

и способах геополитической конкуренции в информационной сфере. Информация стала стратегическим национальным ресурсом любой страны и эффективным оружием в геополитической конкуренции. Разворачивающееся вокруг информационного ресурса соперничество, борьба за достижение и удержание информационного превосходства занимают все более значимое место в общей геополитической конкуренции развитых стран мира.

В связи с этим обеспечение информационной безопасности требует серьезного осмысления и системного взгляда на существующую проблему.

Информационная безопасность перестала быть технической дисциплиной, частью информатики. Вследствие этого, уточнение сущности и содержания информационной безопасности и формирование современной международной системы терминов является актуальной задачей, требующей для ее решения использования достижений многих разноплановых наук.

Библиографический список

1. **Системный** анализ и принятие решений. Словарь-справочник под общей редакцией В. Н. Волковой. – М. : Высшая школа, 2004. – 286 с.

2. **Большой** энциклопедический словарь. – URL: http://slovari.299.ru/enc.php (дата обращения 04.09.2013).

3. **Бутырский, Е. Ю.** Математическое моделирование систем. Монография. – Петродворец, 2006. – 344 с.

4. **Шакин, Д. Н.** и др. Информационная безопасность / Д. Н. Шакин (руководитель), Е. Г. Бунев, С. М. Доценко, А. П. Ильин, П. С. Марголин, В. С. Пирумов, С. И. Тынянкин. – М. : Оружие и технологии, 2009. – 264 с.

5. **Винер, Н.** Кибернетика, или управление и связь в животном и машине. – 2-е изд. – М. : Наука; Главная редакция изданий для зарубежных стран, 1983. – 344 с.

6. **Пирумов, В. С.** Информационное противоборство. Четвертое измерение противостояния. – М. : Оружие и технологии, 2010. – 252 с.

Аннотация

Развитие глобального информационного общества требует переосмысления сущности и содержания информационной безопасности и формирования международной системы терминов с использованием результатов современных исследований в технической и гуманитарной сфере.

В статье предпринята попытка анализа угроз существующих в информационно-технической и информационно-психологической сфере. С позиции системного

подхода автор предлагает расширить содержание информационной безопасности, дополнив его, кроме информационно-технической безопасности – психофизической безопасностью информационных систем.

D. N. Shakin

Department of Federal service for technical and export control (FSTEC Russia) in the North-West Federal district

SKETCH OF A SYSTEMATIC APPROACH TO DEFINING THE ESSENCE AND CONTENT OF INFORMATION SECURITY

Annotation

Development of the global information society requires a rethinking of the nature and content of information security and the formation of the international system of terms using the results of modern researches in technical and humanitarian sphere.

The paper attempts to analyze the threats existing in the information technology an information-psychological sphere. From the position of a systemic approach, the author proposes to expand the content of information security, supplementing it, in addition to information and technical security - physical security of information systems.

Keywords: information security, system approach, the essence, the content, the security of information systems, information technology security, security psychophysical.

References

1. **Sistemnyj** analiz i prinjatie reshenij. Slovar'-spravochnik pod obshhej redakciej V. N. Volkovoj. – M. : Vysshaja shkola, 2004. – 286 s.

2. **Bol'shoj** jenciklopedicheskij slovar'. – URL: http://slovari.299.ru/ enc.php (data obrashhenija 04.09.2013).

3. **Butyrskij, E. Ju.** Matematicheskoe modelirovanie sistem. Monografija. – Petrodvorec, 2006. – 344 s.

4. **Shakin, D. N. i dr.** Informacionnaja bezopasnost' / D. N. Shakin (rukovoditel'), E. G. Bunev, S. M. Docenko, A. P. Il'in, P. S. Margolin, V. S. Pirumov, S. I. Tynjankin. – M. : Oruzhie i tehnologii, 2009. – 264 s.

5. **Viner, N.** Kibernetika, ili upravlenie i svjaz' v zhivotnom i mashine. – 2-e izd. – M. : Nauka; Glavnaja redakcija izdanij dlja zarubezhnyh stran, 1983. – 344 s.

6. **Pirumov, V. S.** Informacionnoe protivoborstvo. Chetvertoe izmerenie protivostojanija. – M. : Oruzhie i tehnologii, 2010. – 252 s.

Шакин Дмитрий Николаевич – кандидат военных наук, доцент, заместитель руководителя Управления Федеральной службы по техническому и экспортному контролю (ФСТЭК России) по Северо-Западному федеральному округу, szfo@fstec.ru

УДК 004.491.22

С. И. Штеренберг, А. В. Красов

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

ВАРИАНТЫ ПРИМЕНЕНИЯ ЯЗЫКА АССЕМБЛЕРА ДЛЯ ЗАРАЖЕНИЯ ВИРУСОМ ИСПОЛНИМОГО ФАЙЛА ФОРМАТА ELF

компьютерный вирус, язык Ассемблера, стеганография, контейнер, исполнимый файл, формат elf

Стеганография... Этот сравнительно недавно вошедший в компьютерный обиход термин обычно переводится как тайнопись. Совместно с криптографией стеганография (steganography) используется для защиты информации, фундаментального свойства окружающего мира, в таковой вовсе не нуждающейся, как, впрочем, и две другие его категории - материя и мера. Требует защиты только изображение информации, представленное на материальном носителе, причем под ней понимается создание условий, исключающих либо затрудняющих доступ к носителю, внесение изменений или уничтожение носителя, а также восприятие представленных на нем данных, производимое с помощью методов криптографии и стеганографии. И если, образно говоря, криптография делает понятное непонятным, то стеганография делает видимое невидимым (иногда и в прямом смысле слова). Достигается это «растворением» скрываемой информации среди других данных значительно большего объема. Видимо, значение стеганографии в деле сокрытия информации сегодня помимо спецслужб в полной мере оценили лишь стоящие вне закона взломщики программ и авторы компьютерных вирусов [1, 3].

Сравнивать язык команд с алгоритмическими языками имеет смысл только в отношении сложности и трудоемкости процесса программирования. Безусловно, при таком сравнении Ассемблер уступает алгоритмическим языкам, но это не означает, что его не следует знать и уметь применять в случае необходимости. Неоспоримым достоинством Ассемблера является возможность составления программ, рационально использующих все особенности системы команд конкретной ЭВМ. Он предоставляет неограниченные возможности для различного рода трюков (в хорошем смысле этого слова), тут все зависит от профессиональных навыков программиста и его изобретательности. Другим положительным свойством является универсальность языка, – он позволяет составить программу для любой задачи, которая имеет решение и может

быть решена на машинах данного семейства. Это утверждение основано на том очевидном факте, что любая программа, составленная на языке высокого уровня, при компиляции преобразуется в последовательность машинных команд.

К коду, генерируемому компилятором, предъявляются следующие требования: он должен быть полностью перемещаемым (т. е. независимым от базового адреса загрузки), не модифицировать никакие ячейки памяти, за исключением стекового пространства, и не использовать стандартные механизмы импорта функций, либо подключая все необходимые библиотеки самостоятельно, либо обращаясь в native-API. Этим требованиям удовлетворяет подавляющее большинство компиляторов.

Дизассемблер позволяет конвертировать машинный код в код на языке Ассемблера. Код на языке Ассемблера является читабельной формой машинного кода (по крайней мере, более читабельной, чем строка битов). С помощью дизассемблера можно узнать, какие машинные инструкции используются в машинном коде. Машинный код является специфическим для конкретной аппаратной архитектуры (например, для чипа PowerPC или Intel Pentium). Поэтому и дизассемблеры пишутся специально для конкретной аппаратной архитектуры.

Декомпилятор – это средство, которое позволяет преобразовать код на языке ассемблера или машинный код в исходный код на высокоуровневом языке, например на *C*. Также существуют декомпиляторы для преобразования кода на промежуточных языках наподобие байт_кода Java и кода на языке MSIL (Microsoft Intermediate Language) в исходный код наподобие Java. Эти средства оказывают огромную помощь в определении структуры кода на высоком уровне, например циклов, операторов switch и конструкций if-then. Хорошая пара дизассемблер/декомпилятор может использоваться для компиляции своего собственного результата восстановления кода обратно в двоичный код.

Декомпиляция – это процесс преобразования двоичных исполняемых файлов (скомпилированной программы) в символический код на языке более высокого уровня, который лучше воспринимается человеком. Как правило, это означает превращение выполняемой программы в исходный код на языке программирования, подобном языку C. Большинство систем для декомпиляции не способны на полное преобразование программ в исходный код. Вместо этого предоставляется нечто среднее. Многие из декомпиляторов одновременно являются и дизассемблерами, которые предоставляют дамп машинного кода, который и заставляет программу работать.

Фактически под UNIX существует всего один более или менее самостоятельный отладчик прикладного уровня – gdb (GNU Debugger), являющийся фундаментом для большинства остальных (рис. 1). Простейшие «антиотладочные» приемы разрушают gdb или позволяют вирусу

вырваться из-под его контроля, поэтому отлаживать вирусный код на рабочей машине категорически недопустимо и лучше использовать для этой цели эмулятор – такой, например, как BOCHS. Особенно предпочтительны эмуляторы, содержащие интегрированный отладчик, обойти который вирусу будет очень тяжело, а в идеальном варианте вообще невозможно (BOCHS такой отладчик содержит).

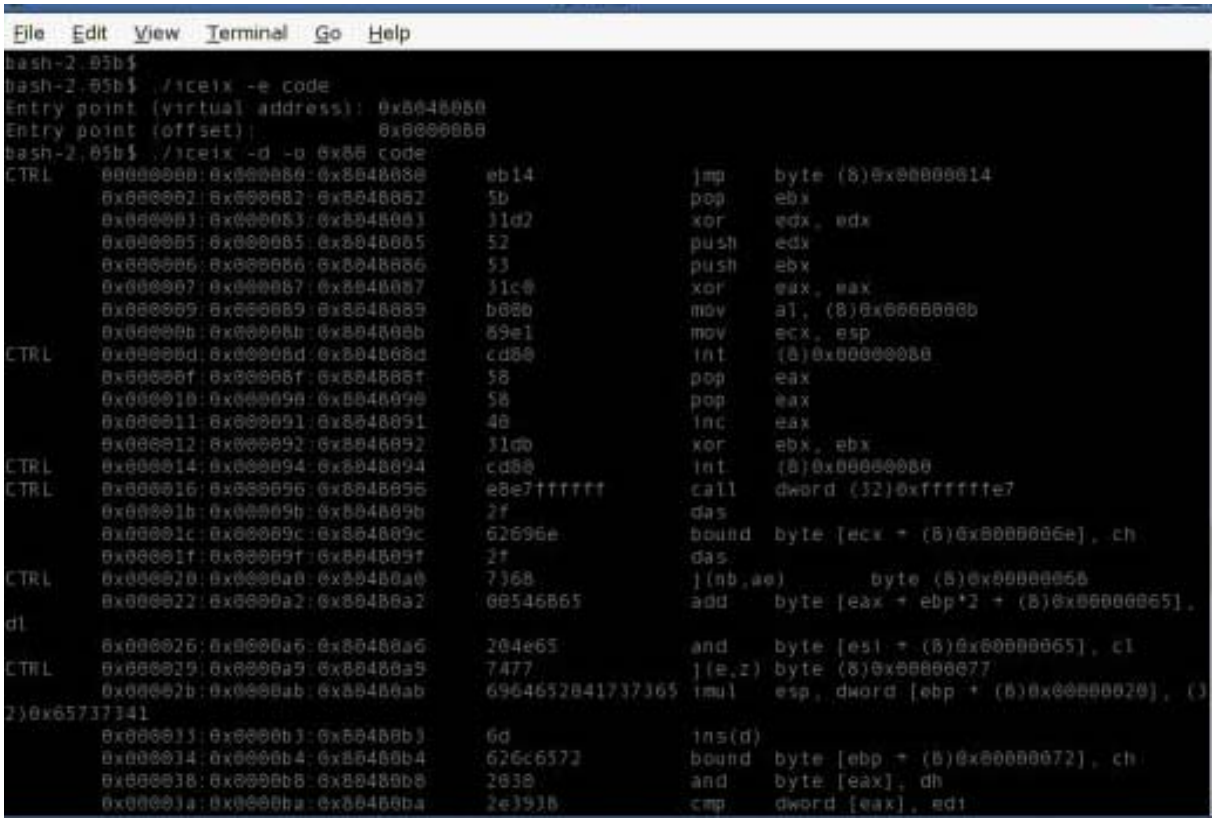

Рис. 1. Исследование вирусов под UNIX с помощью дизассемблера iceix

Антивирусные программы, в том виде, в котором они есть сейчас, категорически не справляются со своей задачей, да и не могут с ней справиться в принципе. Это не означает, что они полностью бесполезны, но надеяться на их помощь было бы, по меньшей мере, неразумно. Как уже отмечалось выше, в настоящий момент жизнеспособных UNIX-вирусов практически нет. И, стало быть, антивирусным сканерам сканировать особо и нечего. Эвристические анализаторы так и не вышли из ясельной группы детского сада и к реальной эксплуатации в промышленных масштабах явно не готовы. Ситуация усугубляется тем, что в скриптовых вирусах крайне трудно выделить устойчивую сигнатуру – такую, чтобы не встречалась в «честных» программах и выдерживала хотя бы простейшие мутации, отнюдь не претендующие на полиморфизм. Например, Антивирус Касперского ловит многие из существующих скриптовых вирусов, но процесс сканирования недостаточно эф-

фективный. Во-первых, вирусы обнаруживаются не во всех файлах, а во-вторых, простейшее переформатирование зараженного файла приводит к тому, что вирус остается незамеченным [2].

Откомпилировав полученный файл, вы получите «объективник» и ругательство компилятора по поводу отсутствия main. Остается только слинковать его в двоичный 32/64-разрядный файл. Естественно, внедрять его в исполнимый файл придется вручную, т. к. системный загрузчик откажется обрабатывать такой файл.

Чрезвычайно важно, чтобы отладчик умел дизассемблировать команды. При подходе к точке останова или пошагового события каждый поток исследуемого процесса по-прежнему указывает на определенную команду. Используя функции структуры CONTEXT, можно определить адрес памяти, где хранится команда, но это не позволяет узнать, какая именно команда была использована.

Формат ELF (Executable and Linkable Format) очень похож на COFF и фактически является его разновидностью, спроектированной для обеспечения совместимости с 32- и 64-разрядными архитектурами. В настоящее время – это основной формат исполняемых файлов в системах семейства UNIX. Не то чтобы он всех сильно устраивал (та же FreeBSD сопротивлялась нашествию Эльфов, как могла, но в версии 3.0 была вынуждена объявить ELF-формат как формат, используемый по умолчанию, поскольку последние версии популярного компилятора GNU C древних форматов уже не поддерживают), но ELF – это общепризнанный стандарт, с которым приходится считаться, хотим ли мы того или нет. Поэтому в настоящей статье речь главным образом пойдет о нем. Для эффективной борьбы с вирусами вы должны изучить ELF-формат во всех подробностях.

Вирусы этого типа пишутся преимущественно начинающими программистами, еще не успевшими освоить азы архитектуры операционной системы. Алгоритм заражения в общем виде выглядит так: вирус находит жертву, убеждается, что она еще не заражена и что все необходимые права на модификацию этого файла у него присутствуют. Затем он считывает жертву в память (временный файл) и записывает себя поверх заражаемого файла. Оригинальный файл дописывается в хвост вируса как оверлей, либо же помещается в сегмент данных (рис. 2, 3).

Получив управление, вирус извлекает из своего тела содержимое оригинального файла, записывает его во временный файл, присваивает ему атрибут исполняемого и запускает «излеченный» файл на выполнение, после чего удаляет с диска вновь. Поскольку подобные манипуляции редко остаются незамеченными, некоторые вирусы «отваживаются на ручную» загрузку жертвы с диска. Впрочем, процедуру для корректной загрузки ELF-файла написать нелегко и еще сложнее ее отладить,

поэтому появление таких вирусов представляется достаточно маловероятным (ELF – это не a.out!).

Характерной чертой подобных вирусов является крошечный сегмент кода, за которым следует огромный сегмент данных (оверлей), представляющий собой самостоятельный исполняемый файл. Попробуйте контекстным поиском найти ELF/COFF/a.out заголовок – в зараженном файле их будет два! Только не пытайтесь дизассемблировать оверлей/сегмент данных, – осмысленного кода все равно не получится, т. к., во-первых, для этого требуется знать точное расположение точки входа, а во-вторых, расположить хвост дизассемблируемого файла по его законным адресам. К тому же оригинальное содержимое файла может быть умышленно зашифровано вирусом, и тогда дизассемблер вернет бессодержательный мусор, в котором будет непросто разобраться. Впрочем, это не сильно затрудняет анализ. Код вируса вряд ли будет очень большим, и на восстановление алгоритма шифрования (если тот действительно имеет место) не уйдет много времени.

Хуже, если вирус переносит часть оригинального файла в сегмент данных, а часть – в сегмент кода. Такой файл выглядит как обыкновенная программа за тем единственным исключением, что большая часть кодового сегмента представляет собой «мертвый код», никогда не получающий управления. Сегмент данных на первый взгляд выглядит как будто бы нормально, однако при внимательном рассмотрении обнаруживается, что все перекрестные ссылки (например, ссылки на текстовые строки) смещены относительно их «родных» адресов. Как нетрудно догадаться – величина смещения и представляет собой длину вируса.

Дизассемблирование выявляет характерные для вирусов этого типа функции exec и fork, использующиеся для запуска «вылеченного» файла, функцию chmod для присвоения файлу атрибута исполняемого и т. д.

Рис. 2. Типовая схема заражения исполняемого файла путем его поглощения

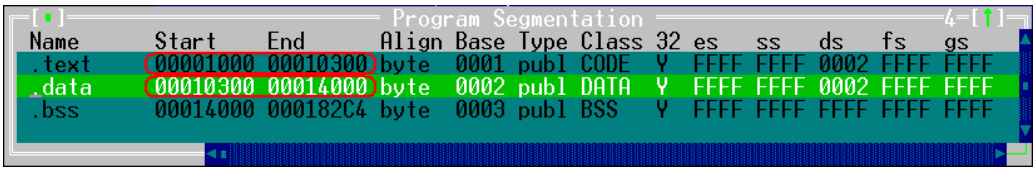

Рис. 3. Пример файла, поглощенного вирусом UNIX.a.out

Простейший способ неразрушающего заражения файла состоит в расширении последней секции/сегмента жертвы и дозаписи своего тела в ее конец (далее по тексту просто «секции», хотя применительно к ELF-файлам это будет несколько некорректно, т. к. системный загрузчик исполняемых ELF-файлов работает исключительно с сегментами, а секции игнорирует). Строго говоря, это утверждение не совсем верно. Последней секцией файла, как правило, является секция .bss, предназначенная для хранения неинициализированных данных. Внедряться сюда можно, но бессмысленно, поскольку загрузчик не настолько простой, чтобы тратить драгоценное процессорное время на загрузку неинициализированных данных с медленного диска. Правильнее было бы сказать «последней значимой секции».

Перед секций .bss обычно располагается секция .data, содержащая инициализированные данные. Вот она-то и становится основным объектом вирусной атаки! Натравив дизассемблер на исследуемый файл, посмотрите, в какой секции расположена точка входа. И если этой секцией окажется секция данных (как, например, в случае, изображенном в листинге 1), исследуемый файл с высокой степенью вероятности заражен вирусом.

При внедрении в a.out-файл вирус в общем случае должен проделать следующие действия:

- считав заголовок файла, убедиться, что это действительно a.out-файл;
- увеличить поле a_data на величину, равную размеру своего тела;
- скопировать себя в конец файла;
- скорректировать содержимое поля a_entry для перехвата управления (если вирус действительно перехватывает управление таким образом).

Внедрение в ELF-файлы происходит несколько более сложным образом:

- вирус открывает файл и, считывая его заголовок, убеждается, что это действительно ELF;
- просматривая Program Header Table, вирус отыскивает сегмент, наиболее подходящий для заражения (для заражения подходит любой сегмент с атрибутом PL_LOAD; собственно говоря, остальные сегменты

более или менее подходят тоже, но вирусный код в них будет смотреться несколько странно);

- найденный сегмент «распахивается» до конца файла и увеличивается на величину, равную размеру тела вируса, что осуществляется путем синхронной коррекции полей p_filez и p_memz;
- вирус дописывает себя в конец заражаемого файла;
- для перехвата управления вирус корректирует точку входа в файл (e_entry), либо же внедряет в истинную точку входа jmp на свое тело (впрочем, методика перехвата управления тема отдельного большого разговора).

Маленькое техническое замечание. Секция данных, как правило, имеет всего лишь два атрибута: атрибут чтения (Read) и атрибут записи (Write). Атрибут исполнения (Execute) у нее по умолчанию отсутствует. Означает ли это, что выполнение вирусного кода в ней окажется невозможным? Вопрос не имеет однозначного ответа. Все зависит от особенностей реализации конкретного процессора и конкретной операционной системы. Некоторые из них игнорируют отсутствие атрибута исполнения, полагая, что право исполнения кода напрямую вытекает из права чтения. Другие же возбуждают исключение, аварийно завершая выполнение зараженной программы. Для обхода этой ситуации вирусы могут присваивать секции данных атрибут Execute, выдавая тем самым себя с головой, впрочем, такие экземпляры встречаются крайне редко, и подавляющее большинство вирусописателей оставляет секцию данных с атрибутами по умолчанию.

Другой немаловажный и не очевидный на первый взгляд момент. Задумайтесь, как изменится поведение зараженного файла при внедрении вируса в не-последнюю секцию .data, следом за которой расположена .bss? Он никак не изменится! Несмотря на то, что последняя секция будет спроецирована совсем не по тем адресам, программный код об этом «не узнает» и продолжит обращаться к неинициализированным переменным по их прежним адресам, теперь занятых кодом вируса, который к этому моменту уже отработал и возвратил оригинальному файлу все бразды правления. При условии, что программный код спроектирован корректно и не закладывается на начальное значение неинициализированных переменных, присутствие вируса не нарушит работоспособности программы.

Однако в суровых условиях реальной жизни этот элегантный прием заражения перестает работать, поскольку среднестатистическое UNIX-приложение содержит порядка десяти различных секций всех назначений и мастей.

Взгляните, например, на строение утилиты ls, позаимствованной из следующего дистрибутива UNIX: Red Hat 5.0:

Листинг 1. Так выглядит типичная карта памяти нормального файла

```
Name      Start     End     Align Base Type Class 32  es   ss   ds   fs
gs

 .init    08000A10 08000A18 para  0001 publ CODE  Y  FFFF FFFF 0006 FFFF
FFFF

 .plt     08000A18 08000CE8 dword 0002 publ CODE  Y  FFFF FFFF 0006 FFFF
FFFF

 .text    08000CF0 08004180 para  0003 publ CODE  Y  FFFF FFFF 0006 FFFF
FFFF

 .fini    08004180 08004188 para  0004 publ CODE  Y  FFFF FFFF 0006 FFFF
FFFF

 .rodata  08004188 08005250 dword 0005 publ CONST Y  FFFF FFFF 0006 FFFF
FFFF

 .data    08006250 08006264 dword 0006 publ DATA  Y  FFFF FFFF 0006 FFFF
FFFF

 .ctors   08006264 0800626C dword 0007 publ DATA  Y  FFFF FFFF 0006 FFFF
FFFF

 .dtor    0800626C 08006274 dword 0008 publ DATA  Y  FFFF FFFF 0006 FFFF
FFFF

 .got     08006274 08006330 dword 0009 publ DATA  Y  FFFF FFFF 0006 FFFF
FFFF

 .bss     080063B8 08006574 qword 000A publ BSS   Y  FFFF FFFF 0006 FFFF
FFFF

 extern   08006574 08006624 byte  000B publ       N  FFFF FFFF FFFF FFFF
FFFF

 abs      0800666C 08006684 byte  000C publ       N  FFFF FFFF FFFF FFFF
FFFF
```

Секция .data расположена в самой середине файла, и чтобы до нее добраться, вирусу придется позаботиться о модификации семи остальных секций, скорректировав их поля p_offset (смещение секции от начала файла) надлежащим образом. Некоторые вирусы этого не делают, в результате чего зараженные файлы не запускаются.

С другой стороны, секция .data рассматриваемого файла насчитывает всего 10h байт, поскольку большая часть данных программы размещена в секции .rodata (секции данных, доступной только на чтение). Это типичная практика современных линкеров, и большинство исполняемых файлов организованы именно так. Вирус не может разместить свой код в секции .data, поскольку это делает его слишком заметным, не мо-

жет он внедриться и в .rodata, т. к. в этом случае он не сможет себя расшифровать (выделить память на стеке и скопировать туда свое тело – не предлагать: для современных вирусописателей это слишком сложно). Да и смысла в этом будет немного. Коль скоро вирусу приходится внедряться не в конец, а в середину файла, уж лучше ему внедриться не в секцию данных, а в секцию .text, содержащую машинный код. Там вирус будет не так заметен (рис. 4, 5).

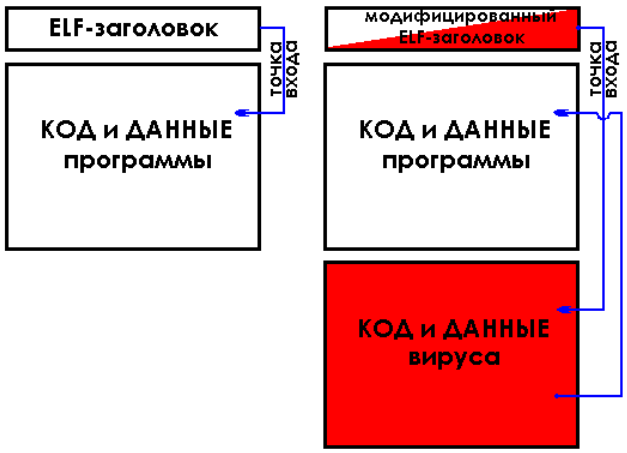

Рис. 4. Типовая схема заражения исполняемого файла путем расширения его последней секции

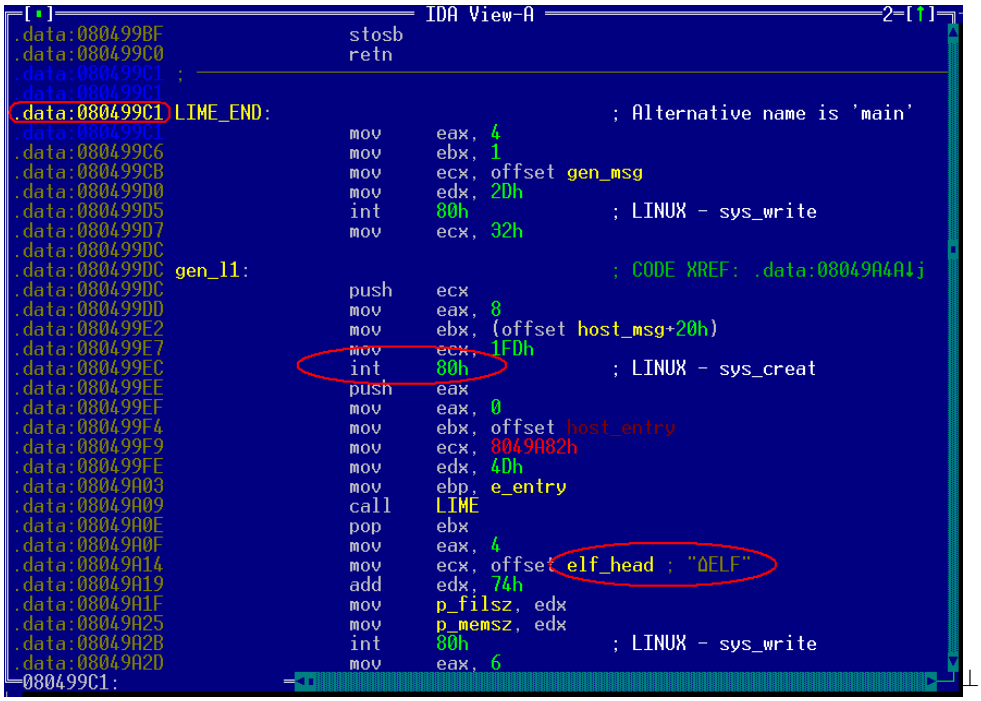

Рис. 5. Внешний вид файла, зараженного вирусом PolyEngine.Linux.LIME.poly; вирус внедряет свое тело в конец секции данных и устанавливает на него точку входа. Наличие исполняемого кода в секции данных делает присутствие вируса чрезвычайно заметным

В данной статье мы исследовали актуальность стеганографии в проблемах защиты информации. В настоящее время, в связи с широким распространением цифровой фототехники и мониторов высокого разрешения в глобальной сети Интернет появляется все больше высококачественных графических файлов. Такие файлы очень хорошо подходят на роль стегоконтейнеров для скрытой передачи личных сообщений. В качестве контейнера были выбраны исполнимые файлы формата elf. Были изучены разные способы заражения этих форматов. Искажение структуры исполняемых файлов – характерный, но недостаточный признак вирусного заражения. Быть может, это защита хитрая такая или завуалированный способ самовыражения разработчика. К тому же некоторые вирусы ухитряются внедриться в файл практически без искажений его структуры. Однозначный ответ дает лишь полное дизассемблирование исследуемого файла, однако это слишком трудоемкий способ, требующий усидчивости, глубоких знаний операционной системы и неограниченного количества свободного времени. Поэтому на практике обычно прибегают к компромиссному варианту, сводящемуся к беглому просмотру дизассемблерного листинга на предмет поиска основных признаков вирусного заражения.

Библиографический список

1. **Беляев, А.** Стеганограмма: скрытие информации // Программист. – 2002. - № 1 (электронная версия).
2. http://www.insidepro.com/kk/336/336r.shtml
3. **Левитин, Ананий В.** Алгоритмы: введение в разработку и анализ = Introduction to The Design and Analysis of Algorithms. - М. : Вильямс, 2006. - С. 392-398. - ISBN 0-201-74395-7.

Аннотация

Целью статьи является исследование вариантов применения языка Ассемблера для скрытого вложения информации в исполнимые файлы формата elf. Актуальность исследования объясняется широким распространение операционной системы Linux и необходимостью решения задачи определения вирусов среди исполнимых файлов, необходимостью определения частей программного обеспечения созданного различными пользователями, отсутствием в настоящее время устоявшихся эффективных методик решения подобных задач скрытого вложения информации в исполнимые файлы.

Тип реализующего ЭВМ (персональные компьютеры и рабочие станции), язык программирования (язык Ассемблера), вид и версия операционной системы (Linux, Mac OS, Windows), исполнимые файлы формата elf.

S. I. Shterenberg, A. V. Krasov

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

METHODS OF USING ASSEMBLY LANGUAGE FOR INFECTION THE VIRUS EXECUTABLE FILE FORMAT .ELF

Annotation

The purpose of this paper is the study of applications of the Assembly Language for concealed attachment information in the executable file format elf. Relevance of the study due to a wide spread of the Linux operating system and the need to address the problem of determining the virus executable file, you must identify the portions of the software created by different users, the current lack of established effective methods of solving such problems hidden embedding information into executables.

Type implements computers (PCs and workstations), a programming language (assembly language), the type and version of the operating system (Linux, Mac OS, Windows), executable file format elf.

Keywords: computer virus, assembly language, steganography, container, executable file format elf.

References

1. **Beljaev, A.** Steganogramma: skrytie informacii // Programmist. – 2002. – № 1 (jelektronnaja versija).
2. http://www.insidepro.com/kk/336/336r.shtml
3. **Levitin, Ananij V.** Algoritmy: vvedenie v razrabotku i analiz = Introduction to The Design and Analysis of Algorithms. – M. : Vil'jams, 2006. – S. 392–398. – ISBN 0-201-74395-7.

Штеренберг Станислав Игоревич – аспирант кафедры Защищенных сетей связи Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», shterenberg.stanislaw@yandex.ru

Красов Андрей Владимирович – кандидат технических наук, доцент, профессор кафедры Защищенных сетей связи Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», krasov@pisem.net

УДК 338.28

В. В. Макаров, В. И. Гусев

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

С. А. Синица

ООО «СулуС»

МЕТОДИЧЕСКИЙ ПОДХОД К ОЦЕНКЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ

информация, интеллектуальный капитал, информационная экономика

Роль информации в условиях информатизации общества постоянно возрастает, и как экономическое благо информация материализуется в экономике как товар или услуга (в виде информационных продуктов), а также как ресурс, используемый в процессе хозяйственной деятельности. К информационным товарам и услугам относятся программное обеспечение, базы данных, образовательные услуги, инженерно-технические услуги, консультирование, результаты НИОКР, разработанные технологии и ноу-хау, прочие результаты интеллектуальной деятельности.

Информационные продукты отличаются рядом особенностей, как на стадиях разработки и производства, так и на этапе обращения. Наиболее ценными информационными ресурсами следует считать продукты научных исследований, технические и технологические услуги – патенты, лицензии, различного рода прикладную текущую научно-техническую информацию (ноу-хау, авторские свидетельства и т. д.), – содержащие описания новых технических достижений и технологических решений.

Учитывая прикладное применение различной информации, функционирующей в телекоммуникационных сетях, можно обоснованно утверждать, что она становится нематериальным ресурсным активом, используемым в самой разнообразной деятельности различных государственных, общественных, коммерческих и иных структур. Таким образом, появляется необходимость оценки этого ресурса, т. е. определение

его потенциала и фактической стоимости в общем объёме активов, используемых в экономической и (или) коммерческой деятельности.

Информация как товар может реализовываться в экономических взаимоотношениях в двух основных формах: а) продажа права собственности на защищённую авторским, смежным или патентным правом информацию; б) лицензионная продажа права пользования на защищённую законодательно информацию.

По расчётам профессоров Стокгольмской школы экономики Кьелла А. Нордстрема и Йонаса Риддерстрале приблизительно 70 % стоимости нового автомобиля приходится на его нематериальную интеллектуальную часть [1], что коррелируется с мнением академика РАН С. Ю. Глазьева, который утверждает, что до 70 % цены на современные товары составляет интеллектуальная рента, а объём мировой интеллектуальной ренты в целом составляет $ 2,2–3,1 трлн. В том числе объём интеллектуальной ренты США составляет $ 365,9–512,3 млрд, Японии – $ 297,2–416,1 млрд, Германии – $ 152,4–212,8 млрд, … России – $ 25,3–35,4 млрд [2], что для России составляет ~ 0,9 %, а для США ~ 16,5 % от общего объёма мировой интеллектуальной ренты.

Экономические интересы при формировании и использовании информационных ресурсов и наиболее полного удовлетворения информационных потребностей при информатизации телекоммуникационного сетевого пространства в мировом масштабе защищены существующими международными и внутренними правовыми актами, законодательно обеспечивающими защиту интеллектуальной собственности правообладателей и условия использования информационных ресурсов потребителями.

В соответствии с Законом РФ «Об информации, информатизации и защите информации» субъектами правоотношений являются:

– «пользователь (потребитель) информации, обращающийся к информационной системе или посреднику за получением необходимой ему информации и пользующийся ею» (ст. 2);

– «собственник информационных ресурсов, наполняющий информацией базы данных, входящих в сеть Интернет информационных систем, технологий и средств их обеспечения – субъект, в полном объёме реализующий полномочия владения, пользования, распоряжения указанными объектами».

Но, необходимо отметить, что субъектами правоотношений являются также юридические и физические лица, на определённых условиях предоставляющие информацию из этих ресурсов потребителям и/или обеспечивающие им возможность подключиться к Интернет и пользоваться его возможностями самостоятельно, обеспечивающие функционирование инфокоммуникационных сетей, распространяющие информацию в Интернет.

Таким образом, весь информационный рынок можно разделить на 4 субъектные группы, состоящие из участников экономических взаимоотношений:

1. Правообладатели (авторы, производители, изготовители, наследники и т. д.) информации;

2. Потребители информации;

3. Сетевые посредники (организаторы, программисты, провайдеры и иные субъекты, участвующие в обеспечении информационного обмена).

4. Авторы и разработчики технических и технологических средств.

Количество производителей интеллектуального продукта в информационном обществе (учёных, инженеров, авторов, разработчиков компьютерных и сетевых программ и т. д.) постоянно растёт, что приводит к увеличению доли результатов деятельности отдельной интеллектуальной личности в конечном рыночном продукте как фактора материального производства.

Соответственно, при определении стоимости востребованного объёма информации часто возникают противоречия между собственниками интеллектуального продукта, производителями и потребителями информации, использующими данную информацию для материализации конечного продукта (в том числе, например, определение издержек на содержание каналов связи, стоимости создания, переработки, транспортировки информации и т. д.), хотя все субъекты правоотношений понимают, что любой товар, реализуемый на рынке, включает в себя информационную основу, которая имеет бóльшую или мéньшую экономическую оценку в общей стоимости товара.

Стоимость затрат на производство всех видов продукции и услуг таким образом, должна включать в себя затраты на используемую информацию (авторские отчисления, лицензирование технологий, программного обеспечения и т. п.) и затраты на материализацию информации (услуги посредников, используемые материальные и людские ресурсы, энергоресурсы), которые, как правило, включаются в себестоимость конечной продукции.

Информация как экономический ресурс в общественном потреблении и на производстве используется различными способами, результатом чего является многообразие форм её использования и, соответственно, расчётов стоимости в конечном продукте или услуге. Среди основных видов использования информации в экономике следует выделить следующие:

а) капитализация информации в товарах, услугах, в новых технологиях производства, управления и т. д.;

б) рыночные потребности и ожидания экономических субъектов путём создания информационного уровня продукта, имиджа компании;

в) интеллектуальная рента в сфере непосредственного производства и информационного обмена в виде того вида информации, которая выступает как фактор производства по цене намного ниже затрат на производство данной информации;

г) фактор процессного подхода в сфере маркетинга и управления для ускорения сроков реализации производимой продукции и (или) услуг;

д) способ унификации (стандартизации) информационного обслуживания и документооборота в управленческой деятельности.

При создании информационных продуктов основным средством производства таких нематериальных ресурсов выступает человеческий интеллект, представляющий собой уникальную способность человека, переосмысливая существующие знания, создавать новые знания, выстраивать логические умозаключения в виде новых идей, открытий, изобретений, технологий и т. п., что позволяет производителю таких интеллектуальных продуктов становиться участником экономических отношений, на равных с обладателями материальных ресурсов [3].

Таким образом, характерной особенностью процесса информационного производства является его субъективность и отсутствие прямой зависимости между материальными затратами и оценкой конечной стоимости результатов производства новой информации и нематериальных ресурсов.

Уникальность подобного вида ресурсов заключается в том, что в результате интеллектуальной деятельности создается некоторый продукт, изначально обладающий неизвестной потенциальной ценностью, но который способен приносить неограниченный доход его создателю в процессе применения и в общественном потреблении (при использовании, тиражировании, продаже) или овеществлении в товарах, средствах производства, технологиях.

Следует также обратить внимание на достаточно чёткое подразделение использования информационного контента, как содержательного наполнения, на коммерческие и некоммерческие цели:

1. Коммерческий контент используется участниками рыночной деятельности (производители продукта, потребители, посредники, авторы и т. д.) и предназначен для получения дохода в том или ином виде за счёт коммерциализации тех или иных результатов интеллектуальной деятельности с возможностью выхода на рынок потребителей предлагаемого ресурса. В этом случае все коммуникативные связи ориентированы на экономически активных участников, заинтересованных в развитии определённого направления коммерческой деятельности с получением той или иной выгоды за счёт заинтересованной целевой аудитории.

2. Некоммерческий контент содержат информацию для посетителей об общественных и деловых событиях, формирует благоприятную об-

щественную и деловую среду, способствующую социальной активности населения. В нём размещается различная бизнес-информация, стимулирующая коммуникативное взаимодействие с органами власти и с представителями бизнеса, что тем или иным образом ориентирует посетителей к участию в коммерческих коммуникационных сетях.

Применение информационных ресурсов в производстве первоначально отражается лишь на себестоимости продукции и других финансовых показателях отдельных применяющих данную информацию предприятий. Но по мере распространения нововведений, овеществляющих новую научно-техническую информацию, её использование снижает общественную стоимость производимых продуктов, а это означает экономию общественного (а не только частного) труда и характеризует переход информации из состояния информационного ресурса в состояние общественно значимого нововведения (нововведения-продукты или нововведения-процессы) [4]. Потребительская стоимость отражает эффект, полученный от использования этой информации.

Имеет значение то, произведена данная информация с целью реализации в общественном производстве, или данная информация не является предметом реализации на производительном рынке. Особое значение имеет информация, имеющая непреходящее культурное или историческое значение в виде общедоступных фактов и документов или произведений искусства (например, произведения фантастов, опередивших в своём воображении время, в котором они жили), или необъяснимая наукой информация о неидентифицируемых наукой явлениях природы, космоса и т. п.

Таким образом, изложенный в данной статье подход к оценке информационных ресурсов, как к одной из составляющих интеллектуального капитала, представляет собой сложную и многоаспектную систему технических, технологических, экономических и правовых взаимозависимостей, служащих для оценки этих ресурсов, являющихся существенной частью интеллектуальной ренты в производительных силах современного общества.

Библиографический список

1. **Кьелл, А. Нордстрем**, Риддерстрале, Йонас Бизнес в стиле фанк. Капитал пляшет под дудку таланта. – СПб. : Стокгольмская школа экономики, 2005.

2. **Глазьев, С. Ю.** Экономическая безопасность в условиях кризиса мировой финансовой системы // Доклады научно-практической конференции «Российские элиты на рубеже двух веков» (социальные технологии нового элитаризма). – СПб., 2000. – С. 19.

3. **Макаров, В. В.** Интеллектуальный капитал. Материализация интеллектуальных ресурсов в глобальной экономике. Монография /

В. В. Макаров, М. В. Семёнова, А. С. Ястребов; под ред. В. В. Макарова. – СПб. : Политехника, 2012. – 688 с.: ил.

4. **Нижегородцев, Р. М.** Информационная экономика. Книга 2. Управление беспорядком. – Москва ; Кострома: Московский государственный университет им. М. В. Ломоносова. Центр общественных наук; Костромской государственный университет им. Н. А. Некрасова, 2002. – 173 с.

Аннотация

В данной статье изложен методический подход к оценке информационных ресурсов, как к одной из составляющих интеллектуального капитала. Раскрыта значимость данного подхода в производительных силах современного общества.

V. V. Makarov, V. I. Gusev

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

S. A. Sinitsa

LLC “SuluS”

METHODICAL APPROACH TO THE EVALUATION OF INFORMATION RESOURCES

Annotation

In this article the methodical approach to assessment of information resources, as one of the components of the intellectual capital. Revealed the importance of this approach in productive forces of modern society.

Key words: information, intellectual capital, information economy.

References

1. **K'ell, A. Nordstrem**, Ridderstrale, Jonas Biznes v stile fank. Kapital pljashet pod dudku talanta. – SPb. : Stokgol'mskaja shkola jeko-nomiki, 2005.

2. **Glaz'ev, S. Ju.** Jekonomicheskaja bezopasnost' v uslovijah krizisa mirovoj finansovoj sistemy // Doklady nauchno-prakticheskoj konferencii «Rossijskie jelity na rubezhe dvuh vekov» (social'nye tehnologii novogo jelitarizma). – SPb., 2000. – S. 19.

3. **Makarov, V. V.** Intellektual'nyj kapital. Materializacija intellektual'nyh resursov v global'noj jekonomike. Monografija / V. V. Makarov, M. V. Semjonova, A. S. Jastrebov; pod red. V. V. Makarova. – SPb. : Politehnika, 2012. – 688 s.: il.

4. **Nizhegorodcev, R. M.** Informacionnaja jekonomika. Kniga 2. Upravlenie besporjadkom. – Moskva ; Kostroma: Moskovskij gosudarstvennyj universitet im. M. V. Lomonosova. Centr obshhestvennyh nauk; Kostromskoj gosudarstvennyj universitet im. N. A. Nekrasova, 2002. – 173 s.

Макаров Владимир Васильевич – доктор экономических наук, профессор, заслуженный деятель науки РФ, заведующий кафедрой Экономики и управления в связи Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

Гусев Василий Игоревич – кандидат экономических наук, доцент кафедры Экономики и управления в связи Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

Синица Сергей Александрович – кандидат экономических наук, Генеральный директор ООО «СулуС»

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 3-2013

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 30.12.2013
Вышло в свет 30.12.2013
Уст. объем 4,394 печ. л. Заказ № 004-ИТТ-2013.
191186, СПб, наб. р. Мойки, 61
E-mail: telecomsut@gmail.com
www.itt.sut.ru