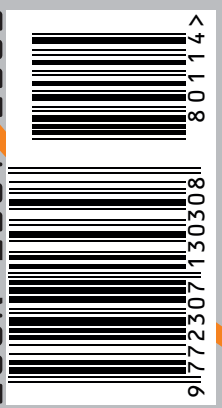
электронный научный журнал

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ

ISSN 2307-1303



выпуск 1, 2014

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ

электронный научный журнал, выпуск 1 за 2014 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: 193232, Россия , С.-Петербург, пр. Большевиков д.22, корп.1, e-mail: telecomsut@gmail.com

Address of the Settler and Publisher, Editorial Office: 193232, Russia, St-Petersburg, Prospekt Bolshevikov 22 - 1, 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

ЛИНИИ СВЯЗИ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ОБРАБОТКИ ДАННЫХ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

электронный научный журнал ISSN 2307-1303
© информационные технологии и телекоммуникации, выпуск 1 за 2014 год

ЛИНИИ СВЯЗИ

УДК 621.396.43. (075.8)

О. С. Данилович

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

ВЛИЯНИЕ ШЕРОХОВАТОСТИ ЗЕМНОЙ ПОВЕРХНОСТИ НА ИНТЕРФЕРЕНЦИОННУЮ НЕУСТОЙЧИВОСТЬ ИНТЕРВАЛОВ РАДИОРЕЛЕЙНЫХ ЛИНИЙ СВЯЗИ

радиорелейные линии, многолучевые замирания, интерференционная неустойчивость, рельеф местности.

Показатели качества передачи на интервалах цифровых радиорелейных линий связи (РРЛ) в значительной степени определяются влиянием замираний сигналов. В случае правильного выбора высот подвеса антенн на интервалах РРЛ сантиметрового диапазона главную роль играют многолучевые замирания, обусловленные отражением радиоволн от слоистых неоднородностей тропосферы и земной поверхности. На пересеченных интервалах отсутствует влияние отражения от земной поверхности, однако при этом имеет место зависимость интенсивности образования слоистых неоднородностей тропосферы от степени шероховатости земной поверхности.

Влияние многолучевости характеризуется интерференционной составляющей неустойчивости. Вопросам оценки интерференционной неустойчивости посвящено большое количество версий Рекомендации Р.530 Международного союза электросвязи (МСЭ). При этом, начиная с версии Р.530-9 [1], все последующие версии содержат выражения для оценки интерференционной неустойчивости как с учетом, так и без учета влияния неровностей земной поверхности.

В последней версии Р.530-14 [2] с учетом неровностей земной поверхности интерференционная неустойчивость определяется выражением, %:

$$T_{\text{инт}} = K_c \cdot R^{3,4}(1+|\varepsilon_p|)^{-1,03} \cdot f^{0,8} \cdot 10^{-0,00076h_L - 0,1M}, \qquad (1)$$

где K_c – геоклиматический параметр,

$$K_c = 10^{-4,4-0,0027dN_1}(10+S_a)^{-0.46}. \quad (2)$$

В (1) и (2) dN_1 – значение градиента рефракции в нижнем 65 м – слое тропосферы, не превышаемое в течение 1 % времени среднего года, 1/км; S_a – стандартное отклонение высотных отметок рельефа местности, м; R –длина интервала, км; $|\varepsilon_p|$ – модуль величины наклона трассы распространения радиоволн, мрад; f – частота, ГГц; h_L – меньшая из высот антенн над уровнем моря, м; M – величина запаса на замирания, дБ.

При этом для приближенной оценки интерференционной неустойчивости рекомендуется использовать выражение, %:

$$T^*_{\text{инт}} = K^*_c \cdot R^{3,1}(1+|\varepsilon_p|)^{-1,29} \cdot f^{0,8} \cdot 10^{-0,00089h_L-0,1M}. \quad (3)$$

В данном случае геоклиматический параметр K^*_c не учитывает рельеф местности и определяется выражением:

$$K^*_c = 10^{-4,6-0,0027dN_1}. \quad (4)$$

Целью данной работы является исследование степени зависимости результатов оценки интерференционной неустойчивости от учета влияния неровностей земной поверхности на пересеченных интервалах разной протяженности в различных геоклиматических условиях.

Для решения указанной задачи были выполнены расчеты интерференционной неустойчивости на пересеченных интервалах в соответствии с выражениями (1), (2), с одной стороны, и выражениями (3), (4), с другой стороны, и проведен сравнительный анализ полученных результатов. При этом в первом случае для оценки S_a использовались высотные отметки профилей интервалов.

Расчет выполнен с использованием массива исходных данных, содержащего топографические данные 15 реальных интервалов длиной от 13 до 54 км с разной величиной наклона трассы распространения радиоволн от 0 до 2,4 мрад, расположенных в районах с разными статистическими характеристиками рефракции радиоволн. При этом использовались диапазон частот 8 ГГц и энергетические параметры современного радиорелейного оборудования «PasolinkNEO» производства фирмы NEC. Предполагалось также, что на интервалах короче 30 км используются антенны диаметром 1,2 м с усилением 37,2 дБ, на интервалах длиной от 30 до 40 км – антенны диаметром 1,8 м с усилением 40,8 дБ и на интервалах длиннее 40 км – антенны диаметром 2,4 м с усилением 43,3 дБ. Результаты расчета представлены в таблице 1.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

ТАБЛИЦА 1. Расчеты интерференционной неустойчивости на пересеченных интервалах

R, м	M, дБ	dN_1, 1/км	S_a, м	ε_p, мрад	$T_{\text{инт}}$, %	$T^*_{\text{инт}}$, %	Z
26,0	24,2	−321,6	10,0	0,86	0,045	0,035	0,22
33,0	29,1	−338,6	16,5	1,06	0,025	0,020	0,20
45,1	31,1	−303,9	14,9	0,07	0,067	0,054	0,19
45,0	31,1	−303,9	21,0	0,91	0,034	0,027	0,21
54,1	29,4	−303,9	46,8	1,40	0,054	0,049	0,09
22,6	25,2	−302,5	10,1	0,31	0,027	0,023	0,15
38,5	26,8	−302,5	4,8	0,39	0,120	0,078	0,35
18,8	27,1	−262,2	30,5	4,20	0,0012	0,0011	0,08
31,2	28,7	−317,5	4,8	0,34	0,039	0,026	0,33
13,3	30,2	−319,4	11,5	1,20	0,0008	0,0007	0,09
16,3	28,4	−319,4	18,5	0,59	0,003	0,0032	−0,07
23,0	25,3	−319,4	11,3	0,02	0,034	0,032	0,06
47,9	30,4	−299,6	40,6	2,37	0,015	0,012	0,20
52,6	29,6	−307,3	14,9	0,02	0,140	0,107	0,24
53,8	29,4	−298,1	5,9	0,17	0,152	0,090	0,41

Для количественной оценки различия результатов расчета интерференционной неустойчивости на основе выражений (1), (2) и (3), (4) в последнем столбце таблицы 1 приведены значения параметра Z, характеризующего относительную величину различия результатов расчета,

$$Z = \frac{T^*_{\text{инт}} - T_{\text{инт}}}{T_{\text{инт}}}.$$

В таблице 2 представлены статистические характеристики параметра Z.

ТАБЛИЦА 2. Статистические характеристики параметра Z

Объем выборки (число интервалов)	Среднее значение Z	Стандартное отклонение Z	Максимальное значение Z	Минимальное значение Z	Диапазон изменения Z
15	0,18	0,12	0,41	−0,07	0,48

Полученные результаты расчетов позволяют сделать следующие основные выводы:

– неучет неровностей земной поверхности практически всегда приводит к заниженной (неоправданно оптимистической) оценке величины интерференционной неустойчивости;

– в отдельных случаях величина относительного различия результатов расчета может достигать 30–40 %;

– с целью корректной оценки показателей качества передачи при выполнении расчета интерференционной неустойчивости без учета неровностей земной поверхности полученное расчетное значение $T^{*}_{\text{инт}}$ целесообразно увеличить, по крайней мере, на 20–30 %;

– с учетом указанной коррекции величина $T^{*}_{\text{инт}}$ может быть использована в качестве ориентировочной оценки интерференционной неустойчивости на этапе предварительного планирования пересеченных интервалов РРЛ.

В заключение следует отметить, что представленные в таблице 2 статистические характеристики относительной величины различия результатов расчета можно рассматривать лишь как ориентировочные. Для более адекватной оценки различия необходимо проведение расчетов на выборке существенно большего размера.

Библиографический список

1. **ITU-RRecommendationP.530-9.** Propagation data and prediction methods required for the design of terrestrial line-of-sight systems. – 2001. – 41 p.

2. **ITU-R Recommendation P.530-14.** Propagation data and prediction methodsrequired for the design of terrestrial line-of-sight systems. – 2012. – 51 p.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

Аннотация

Исследуется влияние неровности земной поверхности на составляющую неустойчивости пересеченных интервалов радиорелейных линий сантиметрового диапазона, обусловленную многолучевыми замираниями сигналов. Оценка интерференционной неустойчивости производилась на основе Рекомендации Международного союза электросвязи (МСЭ) Р.530-14. При проведении исследования рассматривались разные геоклиматические условия и интервалы разной протяженности.

O. Danilovich

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

EFFECT OF THE EARTH'S SURFACE ROUGHNESS ON THE MULTIPATH INSTABILITY OF THE HOPS OF RADIORELAY COMMUNICATION LINKS

Annotation

Explores the impact of the roughness of the Earth's surface on the multipath part of instability due to atmospheric and surface multipath. Evaluation of multipath instability was based on the International Telecommunication Union (ITU) P.530-14. The study examined different geo-climatic conditions and the rough intervals of different lengths.

Keywords: radio relay links, multipath fading, instability due to multipath, roughness of the Earth's surface.

Данилович Олег Сигизмундович – доктор технических наук, профессор кафедры радиотехнических систем Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», danilovich_spb@rambler.ru

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

УДК 621.315

М. С. Былина, С. Ф. Глаголев

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

ЭКСПЕРИМЕНТАЛЬНОЕ ОПРЕДЕЛЕНИЕ УДЕЛЬНОЙ КОНСТРУКТИВНОЙ ПОСТОЯННОЙ ДВУХПРОВОДНОЙ КАБЕЛЬНОЙ ЦЕПИ

импульсный метод, импульсный прибор, двухпроводная кабельная цепь, рефлектограмма, сигнал обратного потока, удельная конструктивная постоянная, импульсная характеристика.

Введение

Удельная конструктивная постоянная является одним из основных параметров двухпроводной кабельной цепи во временной области. В существующей литературе приводятся выражения, позволяющие рассчитать ее через известные вторичные параметры цепи в частотной области. Однако эти выражения неприменимы для неоднородной цепи. В данной работе предлагаются две методики экспериментального определения удельной конструктивной постоянной по зарегистрированной с помощью цифрового импульсного прибора рефлектограмме кабельной цепи. Предложенные методики расширяют возможности исследования кабельных цепей, включая неоднородные, что является чрезвычайно актуальной задачей.

1 Математическая модель однородной двухпроводной цепи во временной области. Связь между параметрами цепи во временной и частотной областях

Для описания процессов распространения импульсных сигналов произвольной формы по двухпроводным цепям удобно использовать их характеристики во временной области. Математическими моделями двухпроводной цепи во временной области являются ее импульсная $g(t)$ и переходная $h(t)$ характеристики, позволяющие определить ее реакцию на любое входное воздействие.

Следует отметить, что получить аналитическое выражение для $g(t)$ и $h(t)$ даже однородной согласованной двухпроводной цепи возможно только при упрощающих предположениях. Ограничим входные воздействия во временной области только короткими импульсами. Тогда волновое сопротивление можно считать чисто активным, не зависящим от

частоты, а для коэффициента затухания использовать аппроксимацию упрощенного вида:

$$\alpha = b_1\sqrt{f}\ . \qquad (1)$$

Тогда для переходной и импульсной характеристик однородной двухпроводной цепи длиной l, согласованной по входу и выходу, можно получить аналитические выражения [1]:

$$h(t) = \operatorname{erfc}\left(\sqrt{\frac{\tau_0 l^2}{t - t_z}}\right)\cdot 1(\tau_z l),\ g(t) = \frac{l\sqrt{\tau_0}}{\sqrt{\pi}\left(t - \tau_z l\right)^{3/2}}\exp\left[-\frac{\tau_0 l^2}{t - \tau_z l}\right]\cdot 1(\tau_z l), \qquad (2)$$

где $t_z = \tau_z l$ – время задержки сигнала;

τ_z–удельное время задержки;

τ_0 – удельная конструктивная постоянная цепи;

$1(t)$ – функция Хэвисайда.

Удельное время задержки и удельную конструктивную постоянную можно определить через вторичные параметры двухпроводной цепи в частотной области [1]:

$$\tau_z = \frac{\beta(f_1)}{2\pi\cdot f_1}, \qquad \tau_0 = \frac{\alpha^2(f_1)}{4\pi\cdot 8{,}68^2\cdot f_1}, \qquad (3)$$

где $\beta(f_1)$ – коэффициент фазы цепи в рад/км на частоте f_1,

$\alpha(f_1)$ – коэффициент затухания цепи в дБ/км на частоте f_1. Частота f_1 для расчета τ_z и τ_0 должна выбираться достаточно высокая. В [1] рекомендуется выбирать ее из частотного диапазона, в котором коэффициент затухания $\alpha(f)$ пропорционален $\sqrt{f}$.

Отметим, что параметры τ_z и τ_0 вводятся для описания реакции кабельной цепи на импульсное воздействие, спектр которого содержит, как высокие, так и низкие частоты. Поэтому расчет этих параметров с использованием измеренных в частотной области коэффициентов затухания и фазы по выражениям (3) не вполне корректен даже для однородной линии.

2 Описание рефлектограммы двухпроводной цепи замкнутой или разомкнутой на конце

В данной работе предлагаются методики экспериментального определения удельной конструктивной постоянной по зарегистрированной с помощью цифрового импульсного прибора рефлектограмме кабельной

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

цепи известной длины, разомкнутой или имеющей короткое замыкание на конце.

Предлагаемые методики основаны на экспериментально подтвержденном в работах [2–4] предположении, что искажения формы импульса, отраженного от неоднородности, имеющей вещественные коэффициенты отражения и пропускания, обусловлены только прохождением импульсов по кабельной цепи и зависят от ее параметров и удвоенного расстояния до неоднородности. При единственной неоднородности с вещественным коэффициентом отражения r_n, находящейся на расстоянии l_n от начала цепи, импульсная характеристика отражения примет вид:

$$g_r(t) = r_n \cdot \frac{2l_n\sqrt{\tau_0}}{\sqrt{\pi}\left(t-2\tau_z l_n\right)^{3/2}} \exp\left[-\frac{4\tau_0 l_n^2}{t-2\tau_z l_n}\right] \cdot 1\left(2\tau_z l_n\right). \qquad (4)$$

Отметим, что если постоянная времени цепи длиной l_n по отражению $\tau = 4\tau_0 l_n^2$ значительно больше длительности зондирующего импульса, то отраженный импульс приобретает форму импульсной характеристики цепи.

Под рефлектограммой неоднородной линии будем понимать зависимость напряжения сигнала обратного потока от времени (расстояния). Если известна импульсная характеристика цепи $g_r(t)$, напряжение обратного потока $u_r(t)$ можно найти через свертку входного зондирующего сигнала $u_1(t)$ с импульсной характеристикой:

$$u_r(t) = \int_0^t u_1(\tau) \cdot g_r(t-\tau)\, d\tau. \qquad (5)$$

Рефлектограмма, полученная с помощью современного цифрового рефлектометра, представляет собой совокупность отсчетов напряжения сигнала обратного потока u_{ri} в моменты времени t_i:

$$u_{ri} = u_r(t_i), \qquad (6)$$

где $u_r(t)$ определяется выражением (5). Интервал Δt между отсчетами связан с частотой дискретизации $f_d = 1 / \Delta t$. Отсчет времени t_i ведется от начала зондирующего импульса.

Между временем t и расстоянием l на рефлектограмме существует однозначная связь:

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

$$l = \frac{t}{2\tau_z} = \frac{tc}{2k_y}, \qquad (7)$$

где c – скорость света в вакууме, k_y – коэффициент укорочения, который показывает во сколько раз скорость света в вакууме превышает скорость распространения электромагнитного импульса по данной линии.

Цифроваярефлектограмма и/или ее отдельные фрагменты могут быть для последующего анализа сохранены в виде файлов, содержащих информацию об отсчетах напряжения сигнала обратного потока. Каждый отсчет представляется двумя значениями – напряжением u_{ri} и соответствующим ему моментом времени t_i. Однако многие приборы вместо моментов t_i записывают в файл соответствующие этим моментам расстояния l_i. Значения l_i рассчитываются прибором по выражению (7) с использованием значения k_y, установленного при регистрации рефлектограммы.

3 Методика определения удельной конструктивной постоянной, основанная на измерении смещения вершины отраженного импульса

В [2, 5] введено понятие смещения Δt_r вершины отраженного от неоднородности импульса на рефлектограмме относительно истинного положения неоднородности (рис. 1), которое зависит от расстояния до неоднородности, параметров кабельной цепи и длительности зондирующего импульса, и показано, что для его расчета можно использовать выражение:

$$\frac{\Delta t_r}{t_p} = \left(1 + \left(\frac{t_g}{t_p}\right)^Q\right)^{\frac{1}{Q}}, \qquad (8)$$

где Q = 1,371 – коэффициент аппроксимации;

t_p – длительность зондирующего импульса;

t_g – положение вершины импульсной характеристики кабельной цепи относительно времени задержки, определяемое выражением:

$$t_g = \frac{8}{3}\tau_0 l_n^2. \qquad (9)$$

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

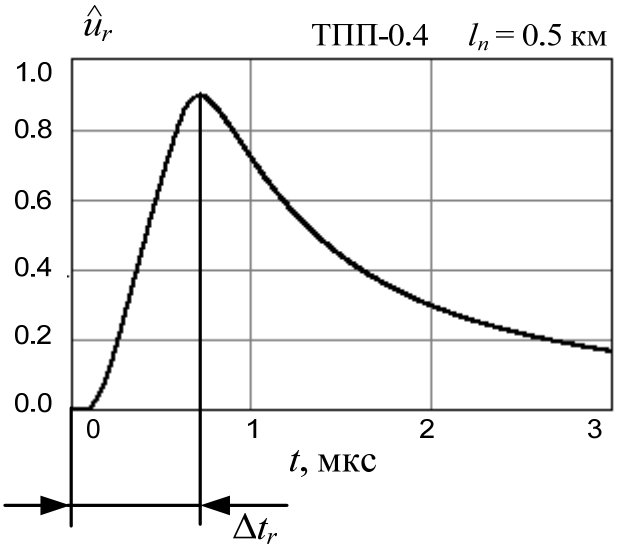


Рис. 1. Определение смещения вершины отраженного
от неоднородности нормализованного импульса

Отметим, что выражение (8) было получено для зондирующего импульса прямоугольной формы – традиционной для современных рефлектометров. Однако в [2] показано, что форма униполярного зондирующего импульса влияет на форму отраженного от неоднородности импульса только при длительности импульсов, большей или сравнимой с постоянной времени линии по отражению τ_{line}:

$$\tau_{line} = 4\tau_0 l_n^2, \qquad (10)$$

то есть при относительно небольших расстояниях до неоднородностей. При выполнении условия:

$$t_p << \tau_{line} \qquad (11)$$

формы отраженных сигналов практически не отличаются, а определяются только площадью зондирующих импульсов. Поэтому выражение (8) может быть использовано для любых униполярных зондирующих импульсов при выполнении условия (11).

Соотношение (8) позволяет предложить следующий алгоритм определения удельной конструктивной постоянной:

1. Зарегистрировать цифровым импульсным прибором рефлектограмму кабельной цепи известной длины l_n, разомкнутой или имеющей короткое замыкание на конце, и сохранить ее на компьютере в виде файла.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

2. Выделить участок рефлектограммы (группу отсчетов сигнала обратного потока), содержащий импульс, отраженный от обрыва или короткого замыкания цепи.

3. Найти отсчет $u_{r\max} = u_r(l_{\max})$, соответствующий вершине отраженного импульса. Рассчитать величину $t_{\max}$ по выражению:

$$t_{\max} = \frac{l_{\max} k_y}{c}. \tag{12}$$

4. Рассчитать смещение вершины отраженного импульса:

$$\Delta t_r = 2\left(t_{\max} - t_z\right), \tag{13}$$

где t_z определяется следующим образом:

$$t_z = \frac{l_n k_y}{c}. \tag{14}$$

В (12) и (14) k_y – коэффициент укорочения, установленный при регистрации рефлектограммы. Отметим, что выражения (12) и (14) позволяют правильно определить $t_{\max}$ и t_z даже при неверной установке коэффициента укорочения в момент регистрации рефлектограммы.

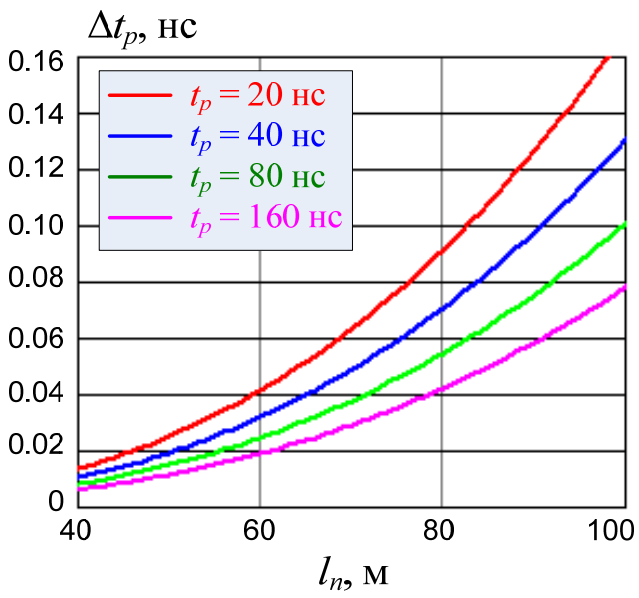
5. Рассчитать удельную конструктивную постоянную по выражению, следующему из (8):

$$\tau_0 = \frac{3t_p}{8l_n^2}\left(\left(\frac{\Delta t_r}{t_p}\right)^Q - 1\right)^{\frac{1}{Q}}, \tag{15}$$

Достоинством данной методики является ее простота, а недостатком – быстрое ужесточение требований к точности задания длительности зондирующего импульса с уменьшением длины линии и точности измерения Δt_r. На рисунке 2 показаны результаты расчета допустимого отклонения длительности зондирующего импульса от истинного значения. При проведении расчетов предполагалось, что погрешность определения удельной конструктивной постоянной не превышает 2 %.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

а



б

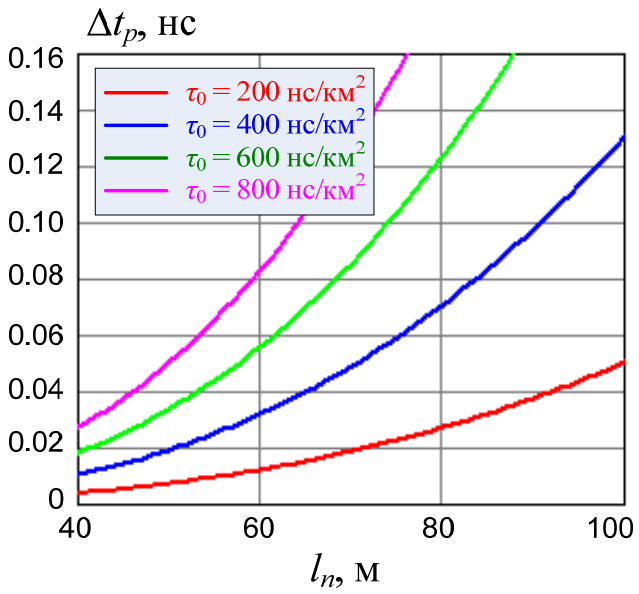


Рис. 2. Допустимое отклонение длительности зондирующего импульса от истинного значения: а – при $\tau_0 = 400$ нс/км^2, б – при $t_p = 40$ нс

Проведенные расчеты показали, что для успешного применения методики рекомендуется выбирать линии относительно большой длины и регистрировать рефлектограммы при малой длительности зондирующего импульса. Рекомендуется также по зарегистрированной рефлектограмме провести отдельное измерение длительности зондирующего импульса, которая может отличаться от номинальной.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

4 Методика определения удельной конструктивной постоянной, основанная на сравнении отсчетов зарегистрированной и теоретически рассчитанной рефлектограмм

Лучших результатов можно достичь, если использовать для анализа всю совокупность отсчетов фрагмента рефлектограммы, содержащего импульс, отраженный от конца разомкнутой или короткозамкнутой линии. Предлагаемая методика основана на сравнении отсчетов экспериментально зарегистрированной рефлектограммы с теоретически рассчитанными отсчетами сигнала обратного потока. Для ее реализации необходимо:

1. Зарегистрировать цифровым импульсным прибором рефлектограмму кабельной цепи известной длины l_n, разомкнутой или имеющей короткое замыкание на конце, и сохранить ее на компьютере в виде файла.

2. Выделить участок рефлектограммы (группу отсчетов сигнала обратного потока), содержащий импульс, отраженный от обрыва или короткого замыкания цепи.

3. Рассчитать значения t_i для отсчетов выделенного в п. 2 участка по выражению:

$$t_i = \frac{l_i k_y}{c}, \tag{16}$$

где k_y – коэффициент укорочения, установленный при регистрации рефлектограммы. Выражение (16) позволяет правильно определить t_i даже при неверной установке k_y в момент регистрации рефлектограммы.

4. Найти отсчет $u_{r\,\max} = u_r(t_{\max})$, соответствующий вершине отраженного импульса. Рассчитать начальное значение удельной конструктивной постоянной τ_{0b} по выражениям (13)–(15).

5. По выражениям (4)–(5) рассчитать теоретическую форму отраженного импульса $u_r(t_i)$ для неоднородности, имеющей коэффициент отражения, равный 1, и расположенной на расстоянии l_n от начала кабельной цепи с удельной конструктивной постоянной $\tau_0 = \tau_{0b}$.

6. Рассчитать оценку расхождения теоретической и экспериментально зарегистрированной рефлектограмм, используя метод наименьших квадратов:

$$D(\tau_0) = \frac{1}{N} \sum_{i=i_{\min}}^{i_{\max}} \left(u_r(t_i, \tau_0) - u_{ri} \right)^2, \tag{17}$$

где $i_{\min}$ и $i_{\max}$ – номера первого и последнего отсчетов в выделенной в п. 2 группе.

7. Найти такое значение τ_0, при котором оценка (17) минимальна. Это значение принимается за истинную удельную конструктивную постоянную кабельной цепи.

Отметим, что вертикальная шкала импульсного прибора может градуироваться не в единицах напряжения, а в так называемых условных единицах. Поэтому для расчета оценки (17) необходимо перевести значения отсчетов зарегистрированной рефлектограммы в единицы напряжения. Для этого следует:

1. Зарегистрировать опорную рефлектограмму кабельной цепи при той же длительности зондирующего импульса, при которой регистрировалась сопоставляемая с теоретическим расчетом рефлектограмма, и небольшом усилении A, которое не приводит к насыщению регистрирующего тракта зондирующим импульсом.

2. По зарегистрированной опорной рефлектограмме измерить амплитуду зондирующего импульса $Y_{\max}$ в условных единицах.

3. Рассчитать величину условной единицы Δu по выражению:

$$\Delta u = \frac{U_1}{Y_{\max} \cdot 10^{(A_{\exp} - A)/20}}. \qquad (18)$$

где A_{exp} – усиление, при котором регистрировалась рефлектограмма, сопоставляемая с теоретическим расчетом в децибелах, U_1 – амплитуда зондирующего импульса в вольтах.

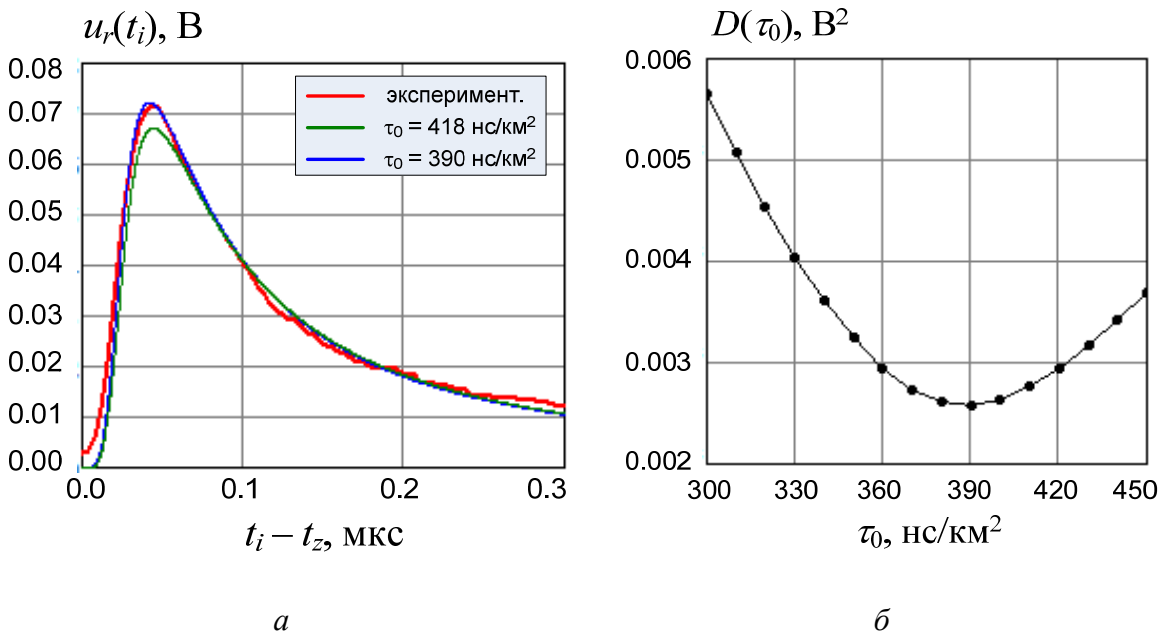
4. Выраженные в условных единицах отсчеты y_{ri} рефлектограммы, сопоставляемой с теоретическим расчетом, перевести в отсчеты, выраженные в вольтах:

$$u_{ri} = y_{ri} \cdot \Delta u. \qquad (19)$$

5 Экспериментальное определение удельной конструктивной постоянной по предложенным методикам

Предложенные методики были применены к определению удельной конструктивной постоянной кабельной цепи MAXILANCat. 5e. Для измерений использовался отрезок кабеля длиной 248 м. На рисунке 2 *а* показан фрагмент экспериментально зарегистрированной рефлектограммы разомкнутой кабельной цепи MAXILANCat. 5e, содержащий отраженный от ее конца импульс. Рефлектограмма была зарегистрирована импульсным прибором РЕЙС-205 при длительности зондирующего импульса 31,25 нс. Определенная по положению вершины отраженного импульса удельная конструктивная постоянная составила 418 нс/км^2. Оценка (17) оказалась минимальной при значении удельной конструктивной постоянной 390 нс/км^2. На рисунке 3 *б* показана зависимость

этой оценки от значения удельной конструктивной постоянной. Для сравнения на рисунке 3 *а* показаны формы отраженного импульса, теоретически рассчитанные при двух указанных значениях конструктивной постоянной.



а *б*

Рис. 3. Эффективность предлагаемых методик для определения удельной конструктивной постоянной кабельной цепи

Из рисунка 3 видно, что обе методики позволяют определить удельную конструктивную постоянную кабельной цепи. Также видно, что точность второй методики несколько выше.

Заключение

В работе предложены две методики определения удельной конструктивной постоянной кабельной цепи поеерефлектограмме, пригодные как для однородных, так и для неоднородных линий. Их эффективность подтверждена проведенными экспериментальными исследованиями.

Библиографический список

1. **Андреев, В. А.** Временные характеристики кабельных линий связи / В. А. Андреев. – М. : Радио и связь, 1986.

2. **Былина, М. С.** Исследование импульсного метода измерений параметров двухпроводных кабельных цепей: Автореф. дис. … канд. техн. наук. СПб., 2006.

3. **Былина, М. С.** Определение характера повреждения или неоднородности по рефлектограмме кабельной цепи / М. С. Былина, С. Ф. Глаголев // Труды учебных заведений связи. – № 168. – СПб., 2002.

4. **Былина, М. С.** Повышение точности и информативности импульсных измерений путем компьютерной обработки зарегистрированной рефлектограммы / М. С. Былина, С. Ф. Глаголев // Труды пятой всероссийской конференции «Современные технологии проектирования, строительства и эксплуатации линейно-кабельных сооружений – СТЛКС». – СПб., 2006.

5. **Былина, М. С.** Повышение точности и информативности импульсного метода измерений / М. С. Былина, С. Ф. Глаголев // Труды четвертой Всероссийской конференции «Современные технологии проектирования, строительства и эксплуатации линейно-кабельных сооружений – СТЛКС». – СПб., 2005.

Аннотация

Удельная конструктивная постоянная является одним из основных параметров двухпроводной кабельной цепи во временной области. В существующей литературе приводятся выражения, позволяющие рассчитать ее через известные вторичные параметры цепи в частотной области. Однако эти выражения неприменимы для неоднородной цепи. В данной работе предлагаются две методики экспериментального определения удельной конструктивной постоянной по зарегистрированной с помощью цифрового импульсного прибора рефлектограмме кабельной цепи. Их эффективность подтверждена проведенными и описанными в работе экспериментальными исследованиями. Предложенные методики расширяют возможности исследования кабельных цепей, включая неоднородные, что является чрезвычайно актуальной задачей.

M. Bylina, S. Glagolev

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

EXPERIMENTAL DETERMINATION OF A SPECIFIC STRUCTURAL CONSTANT OF A CABLE CIRCUIT

Annotation

The specific structural constant is one of the most important transmission line parameters in the time domain. Formulas in the literature on this topic allow us to calculate the specific structural constant using the known transmission line parameters in the frequency domain. However, these formulas are not applicable for the transmission line with multiple discontinuities. This paper contains two experimental procedures for the determination of the specific structural constant directly from the TDR trace. The efficiency of the procedures is confirmed by our experimental research described in this paper. The exper-

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

imental procedures expand opportunities for researching transmission lines including lines with multiple discontinuities.

Keywords: time domain reflectometry, time domain reflectometer, TDR trace, transmission line, specific structural constant, pulse response characteristic.

References

1. **Andreev, V. A.** Vremennye harakteristiki kabel'nyh linij svjazi / V. A. Andreev. – M. : Radio i svjaz', 1986.

2. **Bylina, M. S.** Issledovanie impul'snogo metoda izmerenij pa-rametrov dvuhprovodnyh kabel'nyh cepej: Avtoref. dis. … kand. tehn. nauk. SPb., 2006.

3. **Bylina, M. S.** Opredelenie haraktera povrezhdenija ili neod-norodnosti po reflektogramme kabel'noj cepi / M. S. Bylina, S. F. Glagolev // Trudy uchebnyh zavedenij svjazi. – № 168. – SPb., 2002.

4. **Bylina, M. S.** Povyshenie tochnosti i informativnosti im-pul'snyh izmerenij putem komp'juternoj obrabotki zaregistrirovan-noj reflektogrammy / M. S. Bylina, S. F. Glagolev // Trudy pjatoj vse-rossijskoj konferencii «Sovremennye tehnologii proektirovanija, stroitel'stva i jekspluatacii linejno-kabel'nyh sooruzhenij – STLKS». – SPb., 2006.

5. **Bylina, M. S.** Povyshenie tochnosti i informativnosti im-pul'snogo metoda izmerenij / M. S. Bylina, S. F. Glagolev // Trudy chetvertoj Vserossijskoj konferencii «Sovremennye tehnologii pro-ektirovanija, stroitel'stva i jekspluatacii linejno-kabel'nyh sooruzhenij – STLKS». – SPb., 2005.

Былина Мария Сергеевна – кандидат технических наук, доцент кафедры фотоники и линии связи Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», ittsut@gmail.com

Глаголев Сергей Федорович – кандидат технических наук, доцент, заведующий кафедрой фотоники и линии связи Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», ittsut@gmail.com

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ОБРАБОТКИ ДАННЫХ

УДК 004.438

Ю. В. Скворцов

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

ПРОБЛЕМАТИКА ЯЗЫКА С РАСШИРЯЮЩИМСЯ СИНТАКСИСОМ

язык программирования, синтаксис, расширение.

Введение

Язык программирования с расширяющимся синтаксисом – концепция языка программирования, позволяющая вводить новые ключевые слова и методы обработки данных. Традиционно система программирования (далее – Система) состоит из ядра и окружения (библиотек, функций и т. п). Ядро языка позволяет применять расширения для выявления констант пользовательских типов (например, 40m сделать временным промежутком в 40 минут), организации хранилищ (для использования объектов передачи данных *protobuf* без преобразований), введения новых конструкций, упрощающих написание программы (например, для многопоточного программирования) etc. Это делает написанный код более выразительным и понятным для предметной области, нежели код, написанный на обычном языке программирования общего назначения.

Однако, следует учитывать и проблемы, которые могут возникнуть при расширении языка. Рассмотрим их.

Проблема понимания кода

Может показаться, что расширения языка будет значительно усложнять сам язык. Так, например, происходит с *C++11/14*: вводимые в язык конструкции, такие как шаблоны с переменным числом параметров (от англ. *variadic template*), семантика передвижения (от англ. *move semantic*), и другие нововведения делают и без того сложный язык ещё сложнее, а значит, труднее к восприятию. С другой стороны, эти нововведения обусловлены сложностью тех задач, которые предполагается решать с их помощью, и без нововведений решение было бы ещё сложнее; некоторые нововведения помогают увеличить производительность приложений. Существуют и интуитивно понятные нововведения,

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

например, поддержка именованных аргументов и аргументов по умолчанию, появившихся в *C#* версии 3. Они позволяют сократить объём кода необходимый для вызова методов и убирает необходимость в написании функций с меньшим числом параметров, которые лишь вызывают одноимённые функции, подставляя стандартные значения.

Язык программирования с расширяющимся синтаксисом позволяет ввести множество новых конструкций и использовать их в различных ситуациях. Предполагается, что это позволит решать многие задачи быстрее и нагляднее. Однако, если довести число расширений до абсурдного, например, более 9000, может оказаться, что программный текст перестал быть «си со вставками» и стал представлять нечто совершенно иное. Иным может быть как текст в запутаннейших Perl-скриптах, так и повесть «Война и Мир», из которой расширения попытаются сделать мультипликацию. В результате получаются различные диалекты языка, пользователи которых не в состоянии понять друг друга.

Но понимают ли друг друга программисты традиционных языков программирования? Существует большое количество книг, содержащих методики написания «понимаемого» кода. Часто в этих книгах описываются шаблоны проектирования (от англ. *design pattern*), соглашения о внешнем виде кода (как следует оформлять код, как называть переменные, методы, классы, области имен, и другие). Но, не смотря на всё это, по-прежнему пишется код, который проще написать заново, чем разобраться в нём. Существуют конкурсы программирования, в которых победителям объявляется автор самого непонятно написанного кода, часто представленного в виде *ASCII*-арта (например, Мисака – http://ioccc.org/2013/misaka/misaka.c). Глядя на некоторый код можно подумать, что авторы как будто специально готовились к этому конкурсу, настолько там всё непонятно.

Но даже если код написан довольно хорошо, то всегда ли он может быть понят другим программистом? Как показывают требования к кандидатам на вакансию программиста – едва ли. Очередным препятствием на пути к всеобщему пониманию является большое количество различных каркасов (от англ. *framework*) для создания приложений. Например, для создания приложений с графическим интерфейсом под ОС *Windows* для *C++* существуют *WTL* (*Windows Template Library*) и *qT*. Подходы к заполнению форм, обработке событий, манипуляциям с виджетами (контролами) у них отличаются настолько, что они не имеют практически ничего общего. Поэтому код, написанный на одном из каркасов, будет понятен специалисту, использующему другой каркас, лишь интуитивно.

Пусть код написан довольно хорошо, используются только известные каркасы. Как теперь обстоит дело с его пониманием? Существуют

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

довольно сложные для понимания кода области деятельности (например, шифрование), в которых используются лишь базовые возможности языка, но для понимания большего, чем «тут произошло возведение в степень и получение остатка от деления», «а это присвоение результата сложения по модулю два», требуется глубокое знание предметной области (например, криптографии).

Большинство программистов знакомо с несколькими языками программирования и привычно к тому, что в разных языках код отличается, поэтому ход выполнения может быть интуитивно понятен. Исходя из вышеизложенного, следует вывод, что красиво оформленный код будет понятен большинству с первого взгляда, в то время как действительно сложные задачи требуют большого погружения для понимания.

Проблема при добавлении новых операторов

Разборщик выражений считает применённым оператором самую длинную запись записанную слева направо. То есть, `*a*+++*b*` без каких либо расширений будет разобрано как `*(a++) + b*`. Существует возможность добавить оператор `+++`, который делает, например, сложение по модулю 3. После его введения и применения для текущей единицы компиляции, выражение `*a*+++*b*` перестанет рассматриваться как `*(a++) + b*`, и станет рассматриваться как `*a* +++ *b*`. Хотя это именно то, что ожидается, но старый код мог быть не готов к такому повороту.

Решением является такой стиль написания кода, в котором каждый оператор должен быть отбит пробелом, то есть изначальная запись должна принять вид `*a*++_ + *b*`.

Проблема сосуществования расширений

Получение выполняемого кода можно разделить на несколько этапов: разбиение текста на части (такие как константы, символы, операторы, скобки), формирование из этих частей логически связанных объектов (таких как класс, метод, член класса), компиляцию методов и линковку. Следует отметить, что этап линковки назван так скорее по историческим причинам, как последний этап перед получением программы, нежели является действительно линковкой. Структура реализации Системы предусматривает возможность повлиять на каждый из этих этапов. Хотя эти этапы практически автономны друг от друга, но они связаны: на вход каждого следующего этапа поступают результаты предыдущего, на своём этапе расширение может пользоваться такими элементами, которые сочтёт нужными.

Изначально предполагалось, что имеющихся объектов и их комбинаций будет достаточно для покрытия всех случаев, но потом оказалось, что это может быть не так. Например, семантически циклы *While* и *For*

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

разные, но можно представить *While* как вырожденный случай цикла For, в котором блоки инициализации и итерации пусты (то есть `*while (condition)*` эквивалентен `*for (; condition;)*`). Поэтому для минимизации числа сущностей ядро Системы не содержит такого выражение как *While*, но содержит *For*. В реализациях линковщика, результатом которого является программа, это не играет никакой роли, более того – позволяет упростить его реализацию. Но если линковщиком является транслятор на другой язык, то такая семантическая потеря может быть нежелательной, хотя и мало повлияет на результат компиляции оттранслированного кода. Это можно исправить, добавив дополнительную метаинформацию к узлам синтаксического дерева.

Существуют ситуации, в которых результат может быть выражен не так полно, как на примере циклов *While* и *For*. Такой ситуацией, в частности, является результат выполнения оператора *SizeOf*. Этот оператор возвращает число байт, занимаемых указанным далее типом. В первых реализациях Системы оператор `*sizeof*` возвращал константу: так как класс после своего формирования уже не может изменить свою длину, она остаётся постоянной на всё время выполнения программы. Первоначально байт-код не сохранялся на диск, так как для этого требовалось существенно усложнить виртуальную машину и увеличить время разработки, а хотелось получить прототип Системы в сжатые сроки, поэтому программа компилировалась и выполнялась в памяти. При разработке следующей версии, которая байт-код сохраняет, оказалось, что под размером типа следует считать не константу, но некоторое число, которое может различаться на различных платформах и архитектурах. Это происходит из-за варьирующегося размера указателя (часто 4–8 байт) и из-за выравнивания (например, `*struct { int64 a; byte b; }*` может занимать от 9 байт до 16), но после запуска программы оно неизменно. Для этого самого случая был добавлен новый тип узла синтаксического дерева.

Диспетчер и общее ядро языка, как и ожидалось, восприняли появление ещё одного узла «бесшовно». Но старый линковщик и часть встроенных расширений не понимали новый элемент *SizeOf* и заканчивали работу с сообщением об ошибке. Необходимость добавлять код к ядру системы убивает идею расширяемого языка; расширения, даже такие сложные как преобразователи метода в сопрограмму, должны быть готовы к появлению новых элементов в абстрактном синтаксическом дереве (АСД). Но как обеспечить такую готовность?

Новые типы узлов могут иметь специальный метод, который позволит преобразовать их в такой тип узла, который гарантировано поддерживается (входит в ядро Системы). Можно заменить цикл `*foreach*` на цикл `*for*`, выражения `*sizeof* ` в константы. Такие преобразования позволяют сохранить АСД в том виде, в котором оно полноценно отражает

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

код, но на финальном этапе (генерации программного кода) привести его к эквивалентному с точки зрения выполнения кода.

Реализация шаблона программирования «посетитель» (от англ. *visitor*) для узлов АСД позволяет прочитать все внутренние объекты, что необходимо, например, для реализации замыканий (от англ. *closure*) и сопрограмм.

Возможно, существуют ещё проблемы, которые могут возникнуть при расширении АСД, но пока с ними сталкиваться не приходилось.

Заключение

Будучи применяемыми разумно, расширения базового синтаксиса позволят сохранить код интуитивно понятным, при этом уменьшая количество кода, необходимого для реализации такого же функционала средствами языка программирования общего назначения. Взаимодействие расширений друг с другом и с ядром системы можно реализовать с помощью объектно-ориентированного подхода (виртуальные методы, наследование, поддержка интерфейсов).

Аннотация

Рассмотрены и описаны проблемы языка программирования с расширяющимся синтаксисом. Даны рекомендации по их устранению или минимизации.

Y. Skvortsov

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

PROBLEMS OF LANGUAGE WITH SYNTAX EXPANSION

Annotation

Problems of programming language with syntax expansion are studied and described. Recommendations for their removal or minimization are given.

Keywords: programming language, syntax expansion.

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

Статья представлена рецензентом д-ром техн. наук, профессором Буйневичем М. В.

УДК 004.627

К. Р. Мудрак, С. Л. Федоров

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

ICER: ЦЕЛОЧИСЛЕННЫЙ КОМПРЕССОР, ОСНОВАННЫЙ НА ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИИ

компрессия изображений, вейвлет, адаптивный энтропийный кодер.

ICER – это компрессор статических изображений, использующий вейвлет-преобразование и обеспечивающий прогрессивное кодирование. Он создан специально для нужд космических каналов связи, в то же время обеспечивая максимально возможную степень сжатия изображений.

В 2004 году марсианские роверы «Spirit» (MER-A) и «Opportunity» (MER-B) прибыли на Марс [1]. Более половины информации, передаваемой роверами – это информация с девяти камер, установленных на каждом из них. Все изображения, передаваемые роверами, закодированны при помощи ICERa [2].

В этой статье будут описаны методы, используемые в ICERe, позволяющие использовать его в глубоком космосе.

Под прогрессивным кодированием подразумевается организация процесса кодирования таким образом, что чем больше закодированных бит декодер получит, тем выше будет качество декодированного изображения. Рисунок 1 иллюстрирует увеличение качества, как функцию количества бит на пиксель, с помощью тестового изображения:

а) б) в) г)

Рис. 1. Зависимость качества изображения от количества декодированных бит на пиксель: а) 0,125 бит/пиксель, б) 0,25 бит/пиксель, в) 0,5 бит/пиксель, г) 1 бит/пиксель

Для сравнения, не прогрессивные компрессоры типично кодируют одну область изображения за другой. Потеря фрагментов данных, закодированных таким образом, ведет к полной потере части изображения.

Прогрессивное кодирование заключается в кодировании не пространственных составляющих друг за другом, а спектральных составляющих [3]. Для последнего используется двухмерное вейвлет-преобразование. Фактически вейвлет-преобразование – это два фильтра, НЧ и ВЧ, с совпадающими частотами среза. Двухмерность преобразования заключается в последовательном применении преобразования к строкам, а затем столбцам изображения. Иллюстрация двухмерного вейвлет-преобразования первого порядка приведена на рисунке 2.

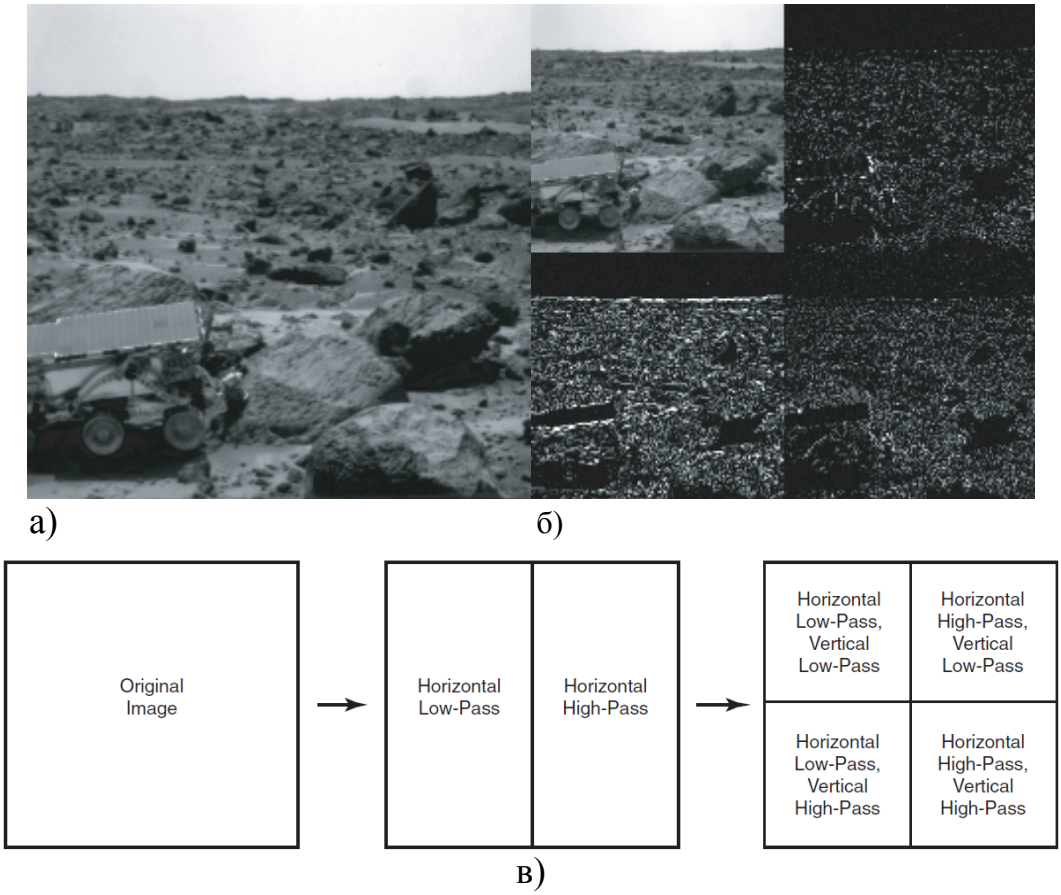


Рис. 2. Иллюстрация двухмерного вейвлет-преобразования первого порядка: а) изначальное изображение, б) вейвлет-преобразованное изображение, в) последовательность преобразования

Применяя вейвлет-преобразование несколько раз последовательно к НЧ части результирующего изображения, можно по-октавно разделить изображения на спектральные составляющие, максимально декоррелированные друг с другом (рис. 3).

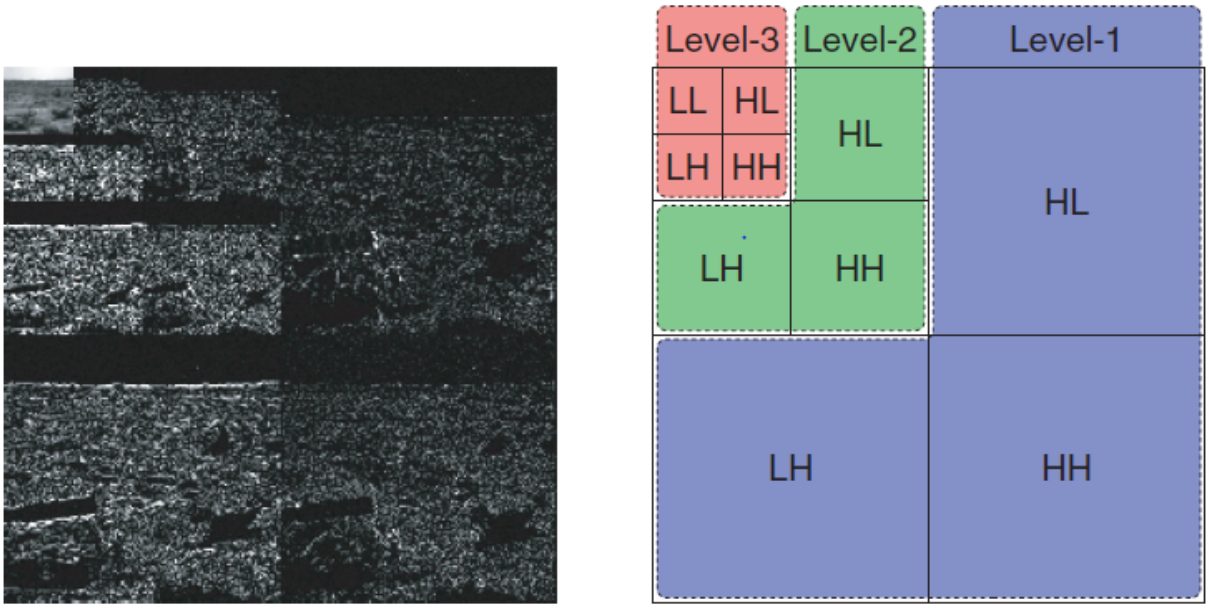


Рис. 3. Десять спектральных поддиапазонов, полученных благодаря применению вейвлет-преобразований над низкочастотной составляющей предыдущих преобразований
(на изображении повышен контраст)

Типично в ICERe используется от трех до девяти последовательных преобразований.

После выполнения всех преобразований, ICER компрессирует последовательно битовые поля, другими словами, сначала компрессии подвергаются все старшие биты (*MSB*) всех пикселей в одном вейвлет-преобразованном участке, затем (*MSB-1*) биты, и т. д. до (*MSB-N*), где *N* – разрядность изображения.

Перед тем, как кодировать бит, кодер определяет вероятность того, что этот бит равен нулю. Вероятность определяется только на основе ранее закодированной информации. Ранее закодированная информация представляется в ICERe таким понятием, как «контекстная схема». По этой схеме, бит, подлежащий кодированию, сначала классифицируется в один из нескольких «контекстов», в зависимости от значений соседних закодированных бит. Эта схема позволяет выделить пиксели, принадлежащие различным объектам и, впоследствии, кодировать их независимо друг от друга. Битам, находящимся в различных контекстах, присваиваются различные вероятности, что они равны нулю, и, как следствие, объекты на изображении кодируются независимо друг от друга.

Благодаря частотному разбиению и разнесению объектов по контекстам, появляется возможность максимально эффективного, объектно-ориентированного кодирования.

После определения контекста и вероятности бита, сам бит и его вероятность передаются в адаптивный перемежающий энтропийный кодер

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

[4]. Данный кодер кодирует информацию в зависимости от сопряженной с ней вероятностью. В кодере содержится семнадцать интервалов вероятности с границами от 0,5 до 1. Бит, поступающий в кодер, в зависимости от сопряженной с ним вероятности, классифицируется в один из этих интервалов. После классификации происходит проверка, не является ли набор бит в этом интервале кодовым словом. В случае если кодовое слово найдено, оно удаляется из этого интервала, а соответствующие биты (чье количество меньше, чем размер кодового слова, например, в 17-м интервале кодовое слово в 512 нулей превращается в одну единицу в 16-м интервале) записываются в интервалы, чьи пределы вероятности лежат ближе к 0,5, чем у исходного интервала. Повторяя данную операцию для всех бит в изображении, представляется возможным закодировать все биты в некую последовательность в первом интервале, чьи границы вероятности 0,5–0,56. Битовый поток с вероятностью, что следующий элемент равен нулю, равной 0,5 невозможно компрессировать сильнее. Таким образом, достигается максимально возможная компрессия информации.

ICER, по своему алгоритму работы, близок к компрессору JPEG 2000:

– прогрессивное кодирование;

– сжатие без потерь;

– сжатие с потерями;

– коррекцию ошибок, позволяющую ограничить эффект потери данных в канале связи;

– обеспечивают разбиение изображения на блоки для увеличения эффективности сжатия, позволяя более эффективно использовать канал связи, оперативную память и процессорное время;

– позволяют варьировать степень сжатия в зависимости от размера изображения (в байтах);

– позволяют варьировать степень сжатия в зависимости от качества (хотя ICER варьирует степень сжатия с 1 % погрешностью).

Но и имеет ряд различий:

– JPEG 2000 использует арифметику с плавающей запятой, ICER – только целочисленную арифметику (для упрощения использования формата на простых процессорах, предназначенных для космических применений);

– ICER использует модифицированный LOCO-компрессор для сжатия без потерь;

– JPEG 2000 использует несколько разных моделей сжатия без потерь, с помощью переключения вейвлет-компрессора в режим сжатия без потерь;

– ICER и JPEG 2000 используют разные цветовые пространства;

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

Подводя итог, можно сформулировать, что ICER, фактически, состоит из двух основных блоков: блок декорреляции изображения, где максимально простым образом разделяется информация о различных объектах на изображении и блок компрессии, где достигается степень компрессии информации, близкая к теоретическому пределу. Грамотно настроенные параметры обоих блоков дают возможность к чрезвычайно эффективной компрессии.

Библиографический список

1. **Mars** Exploration Rover Project / A. Haldemann, J. Crisp, J. Gelles. – URL: http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/12837/1/01-1164.pdf NSS ISDC 27/05/2001. – P. 15.
2. **The ICER** Progressive Wavelet Image Compressor / A. Kiely, M. Klimesh. – URL: http://ipnpr.jpl.nasa.gov/progress_report/42-155/155J.pdf 15/11/2003. – PP. 3–40.
3. **Progressive** Decoding Overview. – URL: http://msdn.microsoft.com/en-us/library/ee720036(v=vs.85).aspx.
4. **A New** Entropy Coding Technique for Data Compression / A. Kiely, M. Klimesh. – URL: http://ipnpr.jpl.nasa.gov/tmo/progress report/42-146/146G.pdf 15/08/2001. – PP. 3–42.

Аннотация

Алгоритмы сжатия изображений, на основе пространственного сегментирования, не смотря на широкое применение и простоту, не подходят для передачи научно значимых изображений. Новые алгоритмы, основанные на разбиении изображений по частотам представляют особый интерес для отраслей, где важна устойчивость к пакетным ошибкам и отсутствие блочных помех.

K. Mudrak, S Fedorov

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

ICER AN INTEGER COMPRESSOR, BASED ON WAVELET DECOMPOSITION

Annotation

Compression algorithms, based on spatial segmentation are ineffective for transmission of science significant images, despite of their wide use and simplexes. New algorithms, based on frequency segmentation are quite interesting for science regions, where packet errors stability and block artifacts are significant.

Keywords: image compression, wavelet, adaptive entropy coder.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

Мудрак Константин Русланович – аспирант кафедры «Телевидение и видеотехника» Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», zzzmurdoczzz@bk.ru

Федоров Сергей Леонидович – кандидат технических наук, доцент кафедры «Телевидение и видеотехника» Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», sergf7@mail.ru

Статья представлена рецензентом д-ром техн. наук, профессором Гоголем А. А.

электронный научный журнал ISSN 2307-1303
© информационные технологии и телекоммуникации, выпуск 1 за 2014 год

УДК 004.94

Д. И. Корольков

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

АЛГОРИТМ ФОРМИРОВАНИЯ ЗАМКНУТЫХ ПОВЕРХНОСТЕЙ АЛГЕБРАИЧЕСКИХ ТЕЛ И ЕГО ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

компьютерное моделирование, алгебраическая модель тела.

Алгебраический полином имеет вид

$$a_0 \times \mathrm{m}_0 + a_1 \times \mathrm{m}_1 + ... + a_k \times \mathrm{m}_k, \quad (1)$$

где a – коэффициент при комплексной переменной, m – комплексная переменная, k – номер члена в алгебраическом полиноме.

Коэффициент a может принимать любые вещественные положительные и отрицательные значения.

Комплексная переменная имеет вид

$$\mathrm{m} = x^n \times y^n \times z^n, \quad (2)$$

где x, y, z – переменные декартового пространства, n – степенной показатель, который целочисленный и может меняться от нуля до n.

В ходе работы алгоритма формирования алгебраического полинома тела создается два симметричных массива с числом ячеек k+2: массив коэффициентов a и массив комплексных переменных m.

Алгоритм выполняет перемножение двух алгебраических полиномов ограничивающих поверхностей тела последовательно до получения окончательного решения в виде алгебраического полинома тела.

Порядок работы алгоритма следующий:

1. Вычисление показателя степени результирующего алгебраического полинома. Степень результирующего алгебраического полинома равна сумме степеней исходных алгебраических полиномов.

2. Формирование массива комплексных переменных результирующего алгебраического полинома с учетом лексографического порядка записи комплексных переменных. Лексографический порядок [1] определяет повышение показателя степени от младшего члена полинома к старшему в последовательности

z-y-x. Возможно использование библиотечных шаблонов алгебраического полинома соответствующего показателя степени.

3. Вычисление коэффициентов *a* при комплексных переменных *m*, полученных при перемножении исходных алгебраических полиномов, и запись в массив с учетом лексографического порядка.

4. Для архивного хранения алгебраического полигона тела и дальнейшей его обработки достаточно иметь массив коэффициентов *a* с записью в первой ячейки массива значения степенного показателя алгебраического полигона тела.

В качестве примера рассмотрим перемножение алгебраических полиномов (3) и (4), используя выше описанный алгоритм перемножения и лексикографической сортировки.

$$-1+0\times z+0\times y+0\times x+5\times yz+5\times xz+5\times xy+3\times z^2+3\times y^2+3\times x^2 \qquad (3)$$

$$-10+0\times z+0\times y+0\times x+0\times yz+0\times xz+0\times xy+2\times z^2+2\times y^2+2\times x^2 \qquad (4)$$

Определяем показатель результирующего алгебраического полинома. Полином (3) 2-й степени и полином (4) 2-й степени. Результирующий полином будет 2 + 2 = 4-й степени.

Формируем лексикографически отсортированный массив комплексных переменных для алгебраического полинома 4-й степени (5).

$$\begin{aligned}&1+z+y+x-yz-xz-xy-z^2-y^2-x^2+xyz+z^2y+z^2x+y^2z+\\&+y^2x+x^2z+x^2y+z^3+y^3+x^3+z^2xy+y^2xz+x^2yz+y^2z^2+\\&+x^2z^2+x^2y^2+x^3y+z^3x+y^3z+y^3x+x^3z+x^3y+z^4+y^4+x^4\end{aligned} \qquad (5)$$

Далее вычисляем коэффициенты при комплексных переменных, полученные при перемножении исходных алгебраических полиномов (3) и (4), результат после последовательного приведения показан в (6), (7) и (8).

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

$$
\begin{aligned}
&(-1\times -10)+(-1\times 0z)+(-1\times 0y)+(-1\times 0x)+(-1\times 0yz)+(-1\times 0xz)+\\
&+(-1\cdot 0xy)+(-1\cdot 2z^2)+(-1\cdot 2y^2)+(-1\cdot 2x^2)+(0z\cdot -10)+(0z\cdot 0z)+\\
&+(0z\times 0y)+(0z\times 0x)+(0z\times 0yz)+(0z\times 0xz)+(0z\times 0xy)+(0z\times 2z^2)+\\
&+(0z\times 2y^2)+(0z\times 2x^2)+(0y\times -10)+(0y\times 0z)+(0y\times 0y)+(0y\times 0x)+\\
&+(0y\times 0yz)+(0y\times 0xz)+(0y\times 0xy)+(0y\times 2z^2)+(0y\times 2y^2)+(0y\times 2x^2)+\\
&+(0x\times -10)+(0x\times 0z)+(0x\times 0y)+(0x\times 0x)+(0x\times 0yz)+(0x\times 0xz)+\\
&+(0x\times 0xy)+(0x\times 2z^2)+(0x\times 2y^2)+(0x\times 2x^2)+(3z^2\times -10)+(3z^2\times 0z)+\\
&+(3z^2\times 0y)+(3z^2\times 0x)+(3z^2\times 0yz)+(3z^2\times 0xz)+(3z^2\times 0xy)+(3z^2\times 2z^2)+\\
&+(3z^2\times 3z^2)+(3z^2\times 3x^2)+(3y^2\times -10)+(3y^2\times 0z)+(3y^2\times 0y)+(3y^2\times 0x)+\\
&+(3y^2\times 0yz)+(3y^2\times 0xz)+(3y^2\times xy)+(3y^2\times 2z^2)+(3y^2\times 2y^2)+(3y^2\times 2x^2)+\\
&(3x^2\times -10)+(3x^2\times 0z)+(3x^2\times 0y)+(3x^2\times 0x)+(3x^2\times 0yz)+(3x^2\times 0xz)+\\
&+(3x^2\times 0xy)+(3x^2\times 2z^2)+(3x^2\times 2y^2)+(3x^2\times 2x^2)
\end{aligned}
\tag{6}
$$

$$
\begin{aligned}
&10+0z+0y+0x-50yz-50xz-50xy-30z^2-30y^2-30x^2+\\
&+0z+0z^2+0yz+0xz+0z^2y+0z^2x+0xyz+0z^3+0y^2z+\\
&+0x^2z+0y+0yz+0y^2+0xy+0y^2z+0xyz+0y^2x+0z^2y+0y^3+\\
&+0x^2y+0x+0xz+0xy+0x^2+0xyz+0x^2z+0x^2y+0z^2x+0y^2x+\\
&+0x^3+0yz+0z^2y+0y^2z+0xyz+0y^2z^2+0z^2xy+0x^2z^2+0x^2yz+\\
&+0z^3x+0y^2xz+15x^3z+0xy+0xyz+0y^2x+0x^2y+0y^2xz+0x^2yz+\\
&+0x^2y^2+0z^2xy+0y^3x+0x^3y-2z^2+0z^3+0z^2y+0z^2x+10z^3y+\\
&+10z^3x+10z^2xy+6z^4+6y^2z^2+6x^2z^2-2y^2+0y^2z+0y^3+0y^2x+\\
&+10y^3z+10y^2xz+10y^3x+6y^2z^2+6y^4+6x^2y^2-2x^2+0x^2z+\\
&+0x^2y+0x^3+10x^2yz+10x^3z+10x^3y+6x^2z^2+6x^2y^2+6x^4
\end{aligned}
\tag{7}
$$

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

$$
\begin{aligned}
&10+(0+0)z+(0+0)y+(0+0)x+(0+0+0-50)yz+(0+0+0-50)xz+\\
&+(0+0+0-50)xy+(-2+0-30)z^2+(-2+0-30)y^2+(-2+0-30)x^2+\\
&+(0+0+0+0+0+0)xyz+(0+0+0+0)z^2y+(0+0+0+0)z^2x+\\
&+(0+0+0+0)y^2z+(0+0+0+0)y^2x+(0+0+0+0)x^2z+\\
&+(0+0+0+0)x^2y+(0+0)z^3+(0+0)y^3+(0+0)x^3+\\
&+(0+0+10+0)z^2xy+(0+0+10+0)y^2xz+(0+0+10+0)x^2yz+\\
&+(0+6+6)y^2z^2+(0+6+6)x^2z^2+(0+6+6)x^2y^2\\
&+(10+0)x^3y+(10+0)z^3x+(10+0)y^3z+(10+0)y^3x+\\
&+(10+0)x^3z+(10+0)x^3y+6z^4+6y^4+6x^4
\end{aligned}
\tag{8}
$$

Затем необходимо установить коэффициенты из алгебраического полинома (8) в шаблон алгебраического полинома (5). Результат произведения представлен в (9).

$$
\begin{aligned}
&10+0z+0y+0x-50yz-50xz-50xy-32z^2-32y^2-32x^2+\\
&+0xyz+0z^2y+0z^2x+0y^2z+0y^2x+0x^2z+0x^2y+0z^3+\\
&+0y^3+0x^3+10z^2xy+10y^2xz+10x^2yz+12y^2z^2+\\
&+12x^2z^2+12x^2y^2+10x^3y+10z^3x+10y^3z+10y^3x+\\
&+10x^3z+10x^3y+6z^4+6y^4+6x^4
\end{aligned}
\tag{9}
$$

Массив записи коэффициентов a при комплексных переменных m показан в (10).

$$
\begin{aligned}
&4;10;0;0;0;-50;-50;--50;-32;-32;-32;0;0;0;0;0;0;0;0;0;0;\\
&10;10;10;12;12;12;10;10;10;10;10;10;6;6;6.
\end{aligned}
\tag{10}
$$

На рисунке показано алгебраическое тело 4-го порядка, составленное в данном примере из двух поверхностей 2-го порядка.

Необходимо отметить, что полученный результат занимает крайне мало компьютерной памяти и может быть сжат без потери данных приблизительно в пять раз.

Это открывает возможности создавать библиотеки большого числа алгебраических полигонов тел сложной геометрической формы.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

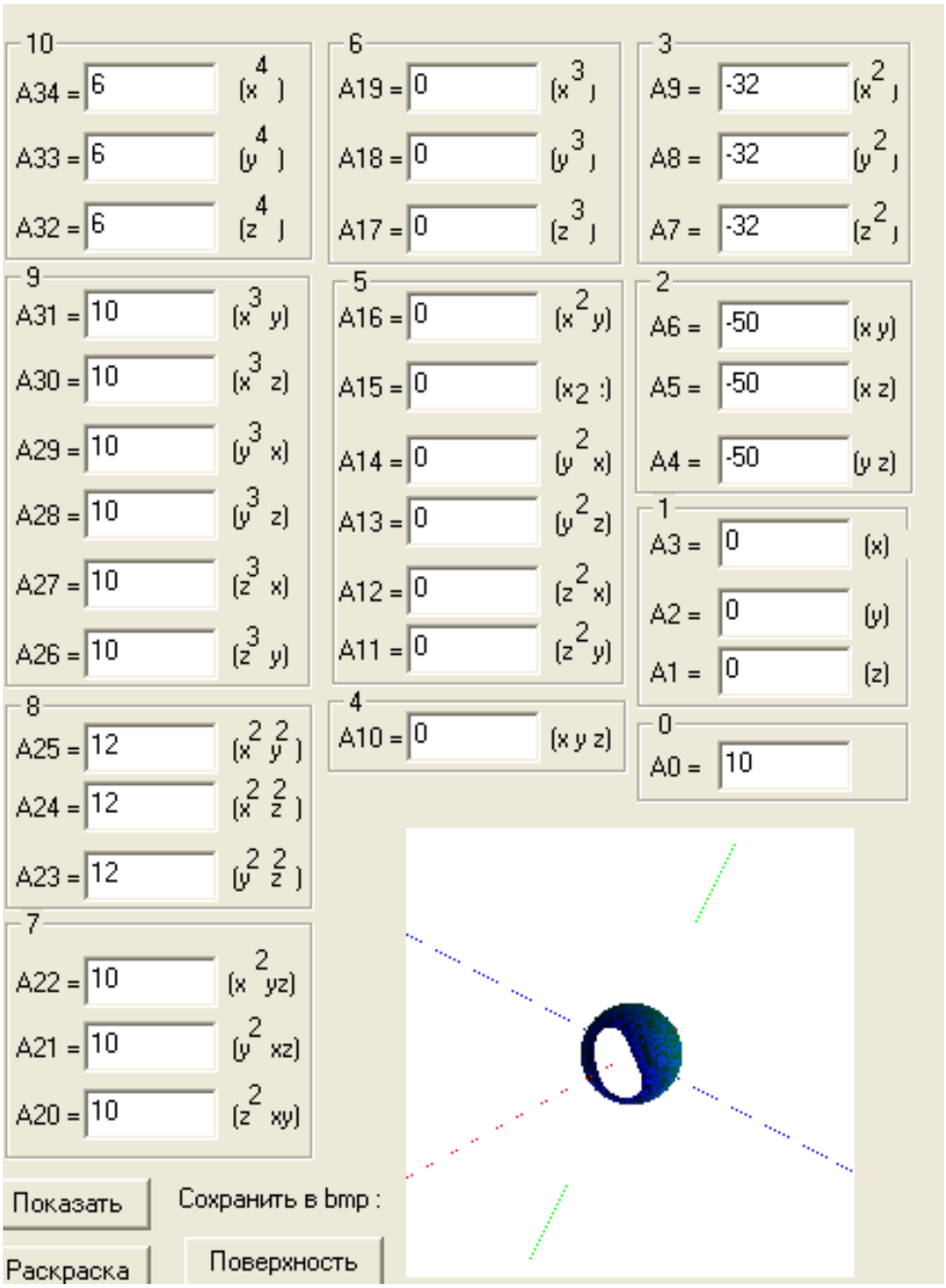


Рисунок. Алгебраическое тело 4-го порядка,
ограниченное двумя поверхностями 2-го порядка

Представленный алгоритм позволяет вычислять коэффициенты при комплексных переменных для алгебраических полиномов любой степени и заданной точности.

Библиографический список

1. **Компьютерная** геометрия и графика: учебник для студ.учреждений высш. проф. образования / В. М. Дегтярев. – 2-е изд., стер. – М. : Издательский центр «Академия», 2011. – 192 с.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

Аннотация

Рассматривается математическое моделирование геометрического тела в виде алгебраического полинома (алгебраического тела), который дает однозначное полное геометрическое описание этого алгебраического тела с любой точностью и сложностью геометрической формы. Алгебраическое тело формируется из отдельных алгебраических полиномов, описывающих ограничивающие поверхности геометрического тела (от плоскости до поверхностей любых порядков), путем последовательного перемножения отдельных алгебраических полиномов. Моделирование осуществляется в трехмерном Декартовом пространстве. Полиномы записываются в определенном лексографическом порядке.

D. Korolkov

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

THE ALGORITHM AND SOFTWARE TO CREATE ALGEBRAIC SURFACES

Annotation

We consider the mathematical modeling of geometric body as an algebraic polynomial (algebraic body), which gives an unambiguous complete geometric description of this algebraic body with any accuracy and complexity of geometric shapes. Algebraic body is formed from separate algebraic polynomials describing the bounding surfaces of a geometric body (from the plane to surfaces of any rate), by sequentially multiplying the individual algebraic polynomials. We are modeling in three-dimensional Cartesian space. Polynomials are written in a certain lexicographical order.

Keywords: CAD, geometric models, optimization of a design.

Reference

1. **Komp'juternaja** geometrija i grafika: uchebnik dlja stud.uchrezhdenij vyssh. prof. obrazovanija / V. M. Degtjarev. – 2-e izd., ster. – M. : Izdatel'skij centr «Akademija», 2011. – 192 s.

Корольков Дмитрий Игоревич – аспирант кафедры информатики и компьютерного дизайна Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», xsau@mail.ru

Статья представлена рецензентом д-ром техн. наук,
профессором Дегтяревым В. М.

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 519.68

С. И. Штеренберг

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

ИССЛЕДОВАНИЕ И АНАЛИЗ ОСОБЕННОСТЕЙ ФОРМАТОВ ИСПОЛНИМЫХ ФАЙЛОВ ПОД LINUX ДЛЯ СКРЫТОГО ВЛОЖЕНИЯ ИНФОРМАЦИИ

стеганография, контейнер, стеганокодер, стеганосистема, передача информации.

Считается, будто бы обмен исполняемыми файлами в мире Linux намного ниже, чем в Windows, что большинство пользователей скачивает исходники и компилирует их самостоятельно. Однако это не так. Исходники занимают намного больше места, а модем столько свободного пространства сколько требуется не имеет. Компиляция больших проектов занимает довольно продолжительное время, зачастую намного превышающее время скачивания (десятки минут или даже часы). Важно одно – очень многие пользователи предпочитают сливать готовые бинарники, скомпилированные для своей оси. Часто такие файлы лежат прямо на официальном сайте производителя [1–9].

Есть и другая проблема. Linux программисты не заостряют внимания с интерактивными конфигураторами и серьезно злоупотребляют директивами условной компиляции. Например, для однопроцессорной машины создается одна сборка, для двух- или четырехпроцессорной – другая. Таких опций может быть очень много и выложить все разновидности сборок на официальный сайт просто нереально. А компилировать самостоятельно – слишком долго. Вот и приходится рыскать по сети в поисках готовых сборок, откомпилированных независимыми разработчиками, и скачивать их. При этом возникает естественная угроза нарваться на вирус, закладку или троян, и такие происшествия уже случались! Доработать исходные тексты проще всего, но что делать, если есть только исполняемый файл и больше ничего? Берем hex-редактор и в самых ответственных местах правим yes на no! А еще круче внедрить «часовой механизм», который в определенный момент выведет приветственное сообщение на экран или выполнит некоторое событие.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

Изначально Unix поддерживали множество исполняемых форматов, ожесточенно конкурирующих между собой, но теперь остался один ELF, ставший стандартом де-факто для Linux и BSD. Аббревиатура **ELF** расшифровывается как **Execution & Linkable Format** (формат исполнения и компоновки). Он состоит в определенном родстве с win32 PE, поэтому у них много общего. В начале ELF-файла расположен служебный заголовок (ELF-header), описывающий основные характеристики файла – тип (исполнения или линковки), архитектура ЦП, виртуальный адрес точки входа, размеры и смещения остальных заголовков. За ELF-header'ом следует таблица сегментов (program header table), перечисляющая имеющиеся сегменты и их атрибуты. В формате линковки она необязательна. Линкер не обращает внимания на сегменты, так как работает исключительно на уровне секций. Напротив, системный загрузчик, загружающий исполняемый ELF-файл в память, игнорирует секции и оперирует целыми сегментами. Ближайший аналог ELF-сегментов – PE-секции, но в PE-файлах секция – это наименьшая структурная единица, а вот в ELF-файлах сегмент может быть разбит на один или несколько фрагментов – секций. В частности, типичный кодовый сегмент состоит из секций .init (процедуры инициализации), .plt (секция связок), .text (основой код программы) и .finit (процедуры финализации). Секции нужны линкеру для комбинирования, чтобы он мог отобрать секции с похожими атрибутами и оптимальным образом растасовать их по сегментам при сборке файла, то есть «скомбинировать»[1].

Несмотря на то, что системный загрузчик игнорирует таблицу секций, линкер все-таки помещает ее копию в исполняемый файл. Место тратится совсем немного, зато отладчикам и дизассемблерам так приятнее. По не совсем понятным причинам gdb и многие другие программы отказываются загружать файл с поврежденной или отсутствующей таблицей секций, чем часто пользуются для защиты программ от постороннего вмешательства.

Структуру и назначение полей служебных заголовков здесь разбирать не будем. Этим займется hex-редактор, и нам эти подробности не понадобятся. Интересующиеся могут обратиться к файлу /usr/include/elf.h – там все подробно расписано. По умолчанию ELF-заголовок находится по адресу 8048000h. Это и есть базовый адрес загрузки. На стадии линковки он может быть свободно изменен на другой, но большинство программистов оставляют его «как есть». Все сегменты проецируются в память в соответствии с виртуальными адресами, прописанными в таблице сегментов, причем, виртуальная проекция образа всегда непрерывна, и между сегментами не должно быть незаполненных «дыр».

[1] http://www.gfs-team.ru

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

Начиная с адреса 40000000h, располагаются совместно используемые библиотеки ld-linix.so, libm.so, libc.so и другие, которые связывают операционную систему с прикладной программой. Ближайший аналог из мира Windows – KERNEL32.DLL, реализующая win32 API, но при желании программа может вызывать функции операционной системы и напрямую. В NT за это отвечает прерывание INT 2Eh, в Linux, как правило, INT 80h.

Для того чтобы сделать обнаружение факта вложения информации в исполняемый код максимально трудным, требуется, чтобы статистика появления определённых инструкций в коде с вложением минимально отличалась от такой же статистики в коде без вложений. В докладе рассматривается статистика появления инструкций в исполняемом коде без вложений и анализируется наиболее подходящие для вложения инструкции.

Как правило, обработчики двоичных файлов распознают файлы по каким-либо «магическим последовательностям» (специальным последовательностям байтов), внедренным в начало файла, а иногда – по какому-либо свойству имени файла. Например, обработчик Java убеждается, что имя файла заканчивается символами .class и что первыми четырьмя байтами (в шестнадцатиричном формате) являются 0xcafebabe, как определено стандартом Java.

Ядро версии 2.2 обеспечивает следующие обработчики двоичных файлов (в системе на базе процессора Intel; порты Linux для других платформ, таких как PowerPC и SPARC, обеспечивают дополнительные обработчики):

− *a.out* (в файле *fs/binfmt_aout.c*). Для двоичных файлов в старом формате Linux. Этот обработчик все еще требуется для обратной совместимости в некоторых системах, но в целом, формат a.out быстро отмирает.

− *ELF* (в файле *fs/binfmt_elf.c*). Для двоичных файлов в новом формате Linux. Этот формат широко используется как для исполняемых файлов, так и для библиотек совместного использования. Большинство современных систем Linux (например, Red Hat 5.2) поставляются только с двоичными файлами в формате ELF, хотя, как правило, они поддерживают также загрузку двоичных файлов в формате a.out, на тот случай, если пользователь решит их установить. Обратите внимание, что несмотря на то, что формат ELF считается собственным форматом Linux, он использует обработчик двоичных файлов, подобно остальным форматам — ядро не оказывает предпочтение ни одному из форматов. Исключение особых случаев способствует упрощению кода ядра.

− *EM86* (в файле *fs/binfmt_em86.c*). Помогает выполнять двоичные файлы Intel Linux на компьютерах Alpha, как если бы они были собственными двоичными файлами Alpha.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

− *Java* (в файле *fs/binfmt_java.c*). Позволяет выполнять файлы .class Java без явного указания интерпретатора байт-кода Java. При этом используется механизм, аналогичный используемому для сценариев; обработчик запускает интерпретатор байт-кода, передавая ему имя файла .class в качестве аргумента. С точки зрения пользователя двоичные файлы Java обрабатываются как собственные исполняемые файлы. Этот обработчик будет рассмотрен подробнее далее в этой главе.

− *Misc* (в файле *fs/binfmt_misc.c*). Наиболее разумный на данный момент из обработчиков двоичных файлов, этот обработчик может распознавать ряд двоичных форматов по специальным внедренным цифрам или по суффиксам имен файлов; но его главное преимущество заключается в том, что он может конфигурироваться во время выполнения, а не только во время компиляции. Следовательно, поддержку новых форматов двоичных файлов можно добавлять «на лету», не выполняя перекомпиляцию ядра и перезагрузку. (Это действительно прекрасная идея!). Комментарии в исходном файле указывают, что со временем обработчики двоичных файлов Java и EM86 будут заменены этим обработчиком.

− *Scripts* (в файле *fs/binfmt_script.c*). Предназначен для сценариев оболочки, сценариев Perl и т.п. Допуская некоторую вольность, можно сказать, что любой исполняемый файл, первыми двумя символами которого являются #!, обрабатываются этим обработчиком двоичных файлов.

Для исследования была использована программа gzip (GNU Zip). Исходный код программы был скомпилирован с помощью компилятора GCC в операционной системе Linux в трёх версиях: без оптимизации (ключ – O0), с максимальной оптимизацией по времени (ключ – O3) и с оптимизацией по размеру (ключ – Os). Для исследования были выбраны инструкции прибавления и вычитания константы с помощью add/sub, обнуления регистра с помощью xor/sub и противоположные условные переходы. Диаграммы распределения каждого из вариантов эквивалентных инструкций приведены на рисунках 1, 2, 3.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

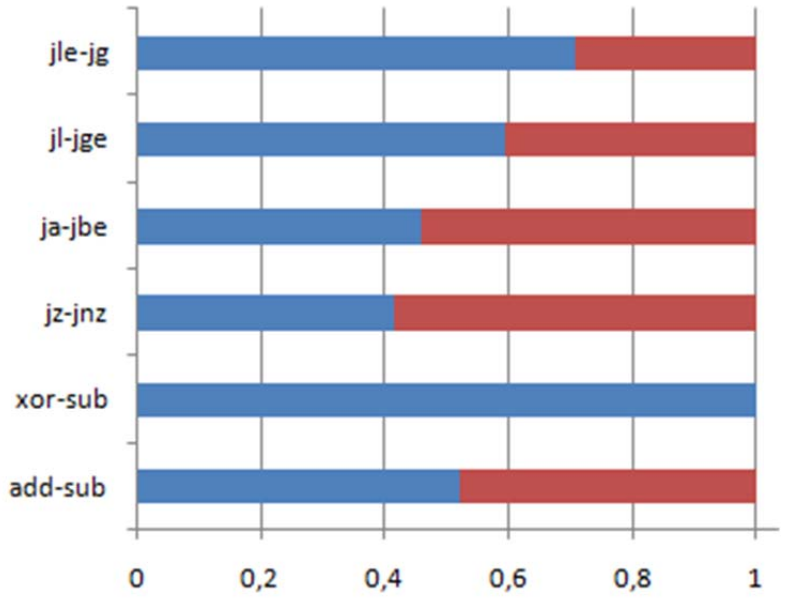


Рис. 1. Распределение вариантов эквивалентных инструкций в исполняемом коде без оптимизации

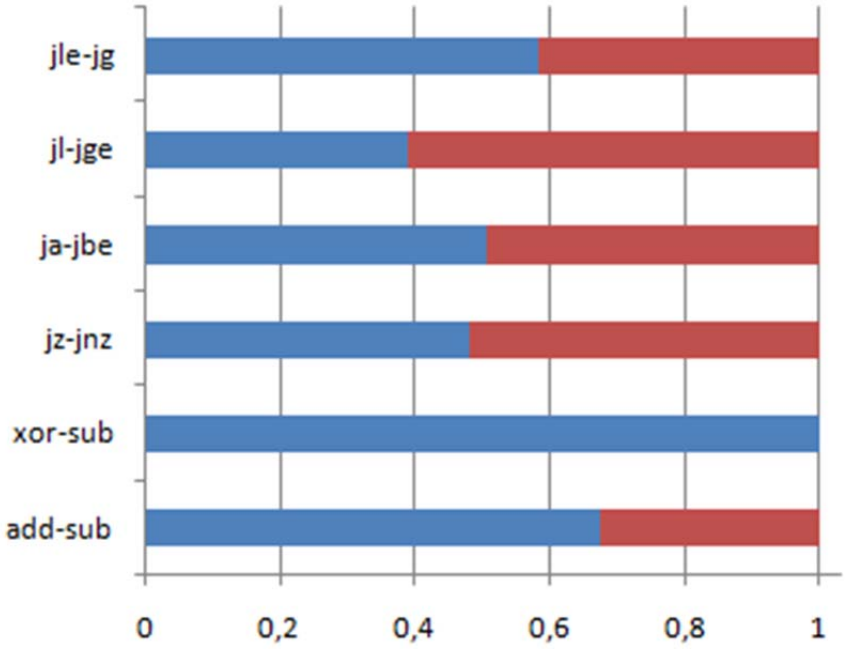


Рис. 2. Распределение вариантов эквивалентных инструкций в исполняемом коде с оптимизацией по времени

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

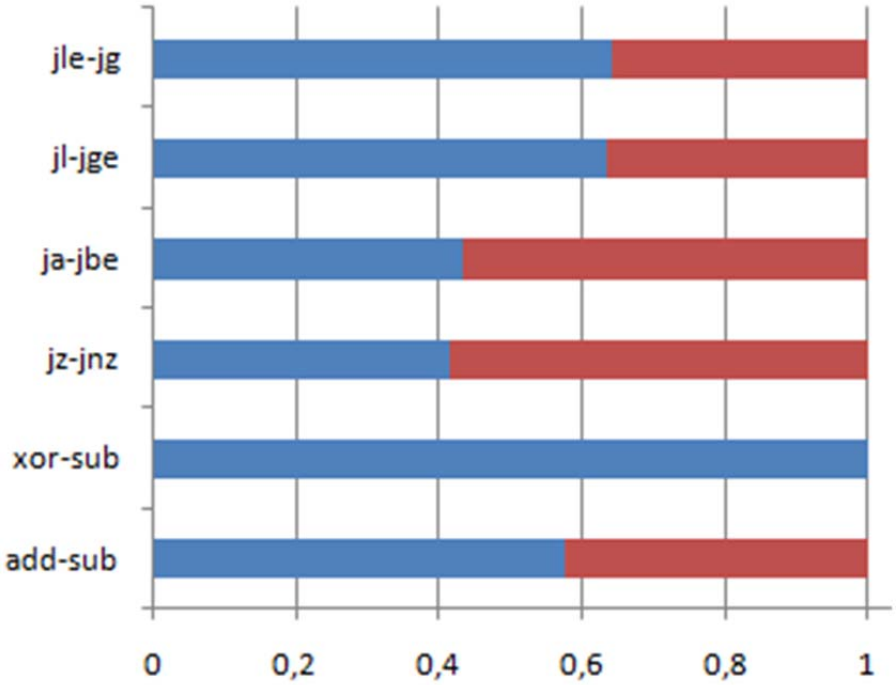


Рис. 3. Распределение вариантов эквивалентных инструкций
в исполняемом коде с оптимизацией по размеру

Из диаграмм видно, что для обнуления регистра компилятор GCC всегда использует инструкцию xor. Это значит, что вложение с помощью замены инструкций xor/sub является легко обнаруживаемым. Остальные синонимы инструкций распределены относительно равномерно и могут быть использованы для вложения информации.

Также была проанализирована частота появления инструкций, использующих в качестве обоих операндов регистры, в исполняемом коде, скомпилированном для различных операционных систем (рис. 4). Больше всего для вложения информации подходят наиболее часто встречающиеся инструкции mov, xor и cmp.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

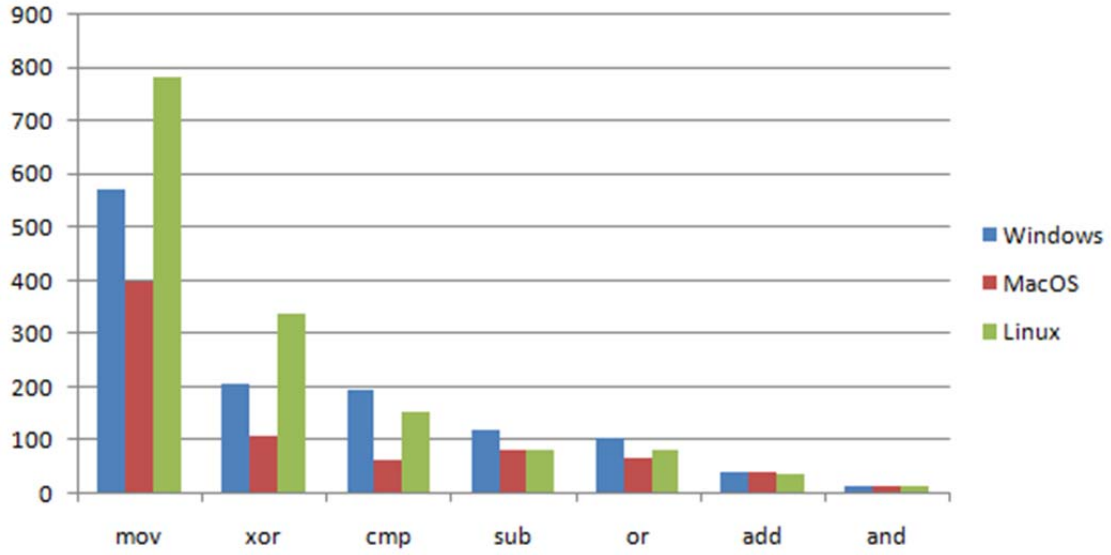


Рис. 4. Количество инструкций, использующих
в качестве обоих операндов регистры, в исполняемом коде
для различных операционных систем

Для вызова функций типа открытия файла можно обратиться либо к библиотеке libc, либо непосредственно к самой операционной системе. Первый вариант – самый громоздкий, самый переносимый и наименее приметный. Последний – прост в реализации, но при первом же взгляде на дизассемблерный листинг тут же бросается в глаза (правильные программы INT 80h не вызывают!), к тому же он испытывает проблемы совместимости с различными версиями Linux.

Последний гигабайт адресного пространства (от адреса C0000000h и выше) занимают код и данные операционной системы, к которым мы будем обращаться только посредством прерывания INT 80h или через разделяемые библиотеки. Стек находится в нижних адресах. Он начинается с базового адреса загрузки и «растет вверх» по направлению к нулевым адресам.

Для экспериментов по имплантации нам потребуется живой исполняемый файл, который при помощи компилятора и текстового редактора мы сможем изготовить и самостоятельно (рис. 5).

Электронный научный журнал ISSN 2307-1303
© Информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

```
#include <stdio.h>

main()

{

printf("LORDI - the best group in the world!\n"\

"(www.lordi.org)\nmonsters, bondage and sado-maso\n");

}
```

Рис. 5. Демонстрационная программа,
в которую мы будем внедрять посторонний код

Давай для разминки просто поменяем первые две команды местами: xor ebp,ebp/pop esi на pop esi/xor ebp,ebp. Подведем курсор к первой машинной команде (она расположена по адресу 80482C2h) и нажмем <Ctrl-A> (Assemble), вводим pop esi. Редактор предложит несколько вариантов ассемблирования на наш выбор: 5Eh и 8Fh C6h. Выбираем 5Eh, как самый короткий (8Fh C6h просто не влезет в отведенное место), затем точно так ассемблируем команду xor ebp,ebp[2].

Полей контрольной суммы в ELF-заголовке нет, и потому обращаться к ее пересчету не нужно. Linux контрольную сумму файла не считает! В операционных системах Windows происходит все иначе. PE-файл содержит заголовке много лишней и часто неиспользуемой информации. Приходится подписывать каждую секцию, легче чтобы программа сама понимала что за секция. Linux и Windows поддерживают механизм отложенной загрузки по требованию. Страницы образа проецируются в память тогда и только тогда, когда к ним происходит обращение, в результате чего, немедленно после запуски, файл готов к работе, а все недостающие страницы дозагружаются уже потом (или не загружаются вообще, например, часть программы, ответственная за печать, вообще не будет загружена, если ни разу не был выбран пункт print). Процесс загрузки как бы «размазывается» во времени, не привлекает внимания никакими песочными часами, которые так любит демонстрировать Windows. Однако, при подсчете контрольной суммы происходит неизбежное обращение ко всем страницам, и все они загружаются в память, даже если не нужны. Получается, что у нас есть два механизма – один оптимизирует загрузку, другой ее «пессимизирует».

А вот разработчики Linux переложили подсчет контрольной суммы на устройства ввода/вывода, которые ее действительно считают. Конеч-

[2] http://www.xakep.ru/magazine/xa/083/106/3.asp

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

но, это не страхует от искажений. В частности, жесткие диски контролируют только физические дефекты, но не обращают внимания на логические искажения (вирусы). Тем не менее, особого смысла в контрольной сумме, хранящейся непосредственно в самой файле, все равно нет. Если вирус может модифицировать файл, он модифицирует и контрольную сумму. Контрольные суммы нужно хранить в отдельном «защищенном хранилище», и их подсчетом должна заниматься файловая система или антивирусные ревизоры. Ни того, ни другого в мире Linux не наблюдается.

Единственную проблему представляют протекторы и упаковщики исполняемых файлов, контролирующие собственную целостность. С каждым годом их становится все больше и больше. UPX, протектор от shiv'ы... В них на первых порах лучше не внедряться!

Выводы

В ближайшее время, по-видимому, следует ожидать большого роста численности ELF-вирусов, ибо для этого имеются все условия. Всплеск интереса к Linux пошел не на пользу этой операционной системе. Чем больше специалистов перейдет на UNIX, тем больше среди них окажется хакеров и вирусописателей и тогда с UNIX произойдет то же, что в свое время произошло с MS-DOS. За основу контейнера для вложения можно взять наиболее подходящий и самый оптимальный исполнимый файл формата .elf. Ближайший аналог ELF-сегментов – PE-секции, но в PE-файлах секция – это наименьшая структурная единица, а вот в ELF-файлах сегмент может быть разбит на один или несколько фрагментов – секций. Формат ELF почти полностью заменил собой более ранний, так называемый формат a.out, который был далеко не столь гибок – среди прочих недостатков формат a.out плохо подходил для динамического связывания, затрудняя реализацию библиотек совместного использования. Linux по прежнему сохраняет обработчик двоичных файлов для формата a.out, но ELF является предпочтительным.

Библиографический список

1. **Свидетельство** о государственной регистрации программы для ЭВМ № 2013612237. Программа для внедрения цифровых водяных знаков в исполняемые и библиотечные файлы / Красов А. В., Верещагин А. С. ; правообладатель: ФГОБУ ВПО «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича». Дата поступления 25 декабря 2012 г. Зарегистрирована в Реестре программ для ЭВМ 18 февраля 2013 г.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

2. **Красов, А. В.** Методы скрытого вложения информации в исполняемые файлы / А. В. Красов, А. С. Верещагин, В. С. Абатуров, М. В. Резник // Известия СПбГЭТУ «ЛЭТИ». – 2012. – № 8. – С. 51–55.

3. **Красов, А. В.** Аутентификация программного обеспечения при помощи вложения цифровых водяных знаков в исполняемый код / А. В. Красов, А. С. Верещагин, А. Ю. Цветков // Телекоммуникации. – 2013. – Спецвыпуск. – С. 27–30.

4. **Немет, Эви** Unix и Linux : руководство системного администратора. Как установить и настроить Unix и Linux = Unix and Linux System Administration Handbook / Эви Немет, Гарт Снайдер, Трент Хейн, Бэн Уэйли. – 4-е изд. – М. : Вильямс, 2012. – 1312 с.

5. **Лав, Роберт** Ядро Linux: описание процесса разработки = Linux Kernel Development / Роберт Лав. – 3-е изд. – М. : Вильямс, 2012. – 496 с.

6. **Блум, Ричард** Командная строка Linux и сценарии оболочки. Библия пользователя = Linux Command Line and Shell Scripting Bible / Ричард Блум, Кристина Бреснахэн. – 2-е изд. – М. : Диалектика, 2012. – 784 с.

7. **Далхаймер, Маттиас Калле** Запускаем Linux / Маттиас Калле Далхаймер. – М. : Символ-Плюс, 2008. – 992 с.

8. **Колисниченко, Д. Н.** Linux. От новичка к профессионалу / Д. Н. Колисниченко. – 2-е изд. – СПб. : БХВ-Петербург, 2010. – 764 с.

9. **Уэлш, Мэтт** Запускаем Linux / Мэтт Уэлш, Маттиас Калле Далхаймер, Терри Доусон и Лар Кауфман. – 4-е изд. – СПб-М. : Символ-Плюс, 2004. – 730 с.

Аннотация

В статье рассматриваются особенности форматов исполнимых файлов под операционные системы Linux и методика выбора наиболее подходящих форматов файлов для скрытого вложения информации.

S. Shterenberg

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

INVESTIGATION AND ANALYSIS OF FEATURES EXECUTABLE FILE FORMATS ON LINUX FOR HIDDEN ASSETS INFORMATION

Annotation

The aim of the article is to study the methods for selecting a container for embedding hidden information. Surge of interest in Linux has gone not to the benefit of this operating system. The more professionals move to Linux, the more of them will be hackers and virus writers with Linux and then what happens is the same as that at the time happened to MS-DOS.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

Keywords: steganography, container steganokoder, steganosistema transferring information.

References

1. **Svidetel'stvo** o gosudarstvennoj registracii programmy dlja JeVM № 2013612237. Programma dlja vnedrenija cifrovyh vodjanyh znakov v ispolnjaemye i bibliotechnye fajly / Krasov A. V., Vereshhagin A. S. ; pravoobladatel': FGOBU VPO «Sankt-Peterburgskij gosudarstvennyj universitet telekommunikacij im. prof. M. A. Bonch-Bruevicha». Data postuplenija 25 dekabrja 2012 g. Zaregistrirovana v Reestre programm dlja JeVM 18 fevralja 2013 g.

2. **Krasov, A. V.** Metody skrytogo vlozhenija informacii v ispolnjaemye fajly / A. V. Krasov, A. S. Vereshhagin, V. S. Abaturov, M. V. Reznik // Izvestija SPbGJeTU «LJeTI». – 2012. – № 8. – S. 51–55.

3. **Krasov, A. V.** Autentifikacija programmnogo obespechenija pri pomoshhi vlozhenija cifrovyh vodjanyh znakov v ispolnjaemyj kod / A. V. Krasov, A. S. Vereshhagin, A. Ju. Cvetkov // Telekommunikacii. – 2013. – Specvypusk. – S. 27–30.

4. **Nemet, Jevi** Unix i Linux : rukovodstvo sistemnogo administratora. Kak ustanovit' i nastroit' Unix i Linux = Unix and Linux System Administration Handbook / Jevi Nemet, Gart Snajder, Trent Hejn, Bjen Ujejli. – 4-e izd. – M. : Vil'jams, 2012. – 1312 s.

5. **Lav, Robert** Jadro Linux: opisanie processa razrabotki = Linux Kernel Development / Robert Lav. – 3-e izd. – M. : Vil'jams, 2012. – 496 s.

6. **Blum, Richard** Komandnaja stroka Linux i scenarii obolochki. Biblija pol'zovatelja = Linux Command Line and Shell Scripting Bible / Richard Blum, Kristina Bresnahjen. – 2-e izd. – M. : Dialektika, 2012. – 784 s.

7. **Dalhajmer, Mattias Kalle** Zapuskaem Linux / Mattias Kalle Dalhajmer. – M. : Simvol-Pljus, 2008. – 992 s.

8. **Kolisnichenko, D. N.** Linux. Ot novichka k professionalu / D. N. Kolisnichenko. – 2-e izd. – SPb. : BHV-Peterburg, 2010. – 764 s.

9. **Ujelsh, Mjett** Zapuskaem Linux / Mjett Ujelsh, Mattias Kalle Dalhajmer, Terri Douson i Lar Kaufman. – 4-e izd. – SPb-M. : Simvol-Pljus, 2004. – 730 s.

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

Статья представлена рецензентом д-ром техн. наук, профессором Буйневичем М. В.

УДК 004.492.3

И. А. Федянин

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

О ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ МЕТОДА ВЫЧИСЛЕНИЯ ДИВЕРГЕНЦИИ, ОСНОВАННОГО НА ПОИСКЕ БЛИЖАЙШЕГО "СОСЕДА", ДЛЯ ОПТИМИЗАЦИИ СТЕГОСИСТЕМ

стеганография, дивергенция, обработка изображений, относительная энтропия.

Стеганография является активно развивающимся направлением современной науки о защите информации. Стеганография позволяет скрыть сам факт передачи информации, в отличие от криптографии, которая делает сообщение “нечитаемым” но при этом факт передачи зашифрованного сообщения является известным. Для вложения используется покрывающее сообщение (ПС) которое может представлять быть изображением, видео, звуковыми файлами, текстом и другими объектами. В частности, в данной работе, в качестве ПС рассматриваются изображения, однако, следует отметить, что рассматриваемые методы можно обобщить и на остальные мультимедийные объекты. ПС в который произведено вложение – называется стеганограммой (СГ)

Стегоанализ – направление науки, задачей которого является классификация объектов на ПС и СГ. В данной работе рассматриваются вопросы возможности расчета вероятности ошибки “идеального” стегоанализатора. Для этого используется математический аппарат относительной энтропии.

Пусть P, Q d-мерные случайные распределения, $P, Q \in \left(\Re^d, B_{\Re^d}\right)$. Тогда дивергенция(относительная энтропия) определяется следующим образом [1]:

$$D(P \| Q) = \begin{cases} \int\limits_{\Re^d} dP \cdot \log \dfrac{dP}{dQ}, \text{ когда } P \text{ абсолютно непрерывно по отношению к } Q \\ +\infty, \text{ в остальных случаях} \end{cases}$$

Если плотности P и Q существуют, и обозначаются $p(x)$ и $q(x)$ соответственно, тогда

$$D(p \| q) = \int_{\Re^d} p(x) \cdot \log \frac{p(x)}{q(x)} dx. \quad (1)$$

Дивергенция показывает насколько сильно различаются распределения двух случайных величин, и удобной является мерой разницы двух случайных распределений.

В работе [2] доказывается, что для любого метода стегоанализа выполняется следующие неравенства:

$$P_{fa} \log(\frac{P_{fa}}{1-P_m}) + (1-P_{fa}) \log(\frac{1-P_{fa}}{P_m}) \le D(P_w \| P_c), \quad (2)$$

$$P_m \log(\frac{P_m}{1-P_{fa}}) + (1-P_m) \log(\frac{1-P_m}{P_{fa}}) \le D(P_c \| P_w), \quad (3)$$

где P_{fa}и P_m – вероятности ложной тревоги и пропуска цели соответственно, P_w, P_c – соответственно распределения СГ и ПС.

Таким образом, формулы (2), (3) позволяют вычислить вероятность ошибки с которой может наилучший стегоанализатор различить СГ и ПС. Однако, что бы определить вероятность ошибки по формулам (2), (3) необходимо вычислить дивергенцию, для чего необходимо знать распределения ПС и СГ – P_c. И P_w. Распределения реальных изображений получить практически невозможно, что ограничивает применение формул (2), (3) только для простейших случаев известных распределений.

В работе [3] описывается метод эмпирического вычисления дивергенции на основе поиска ближайшего «соседа».

Пусть $\{X_1, \dots, X_n\}$ и $\{Y_1, \dots, Y_m\}$ это вектора отсчетов с плотностью распределения p и q соответственно, при этом эти два набора векторов независимы друг от друга. Определим расстояние от X_i до его ближайшего соседа в $\{X_j\}_{j \ne i}$ как

$$\rho_n(i) = \min_{j=1,\dots,n, j \ne i} \left\| X_i - X_j \right\|,$$

где L^2 это норма в $\Re^d$. Аналогично, введем расстояние от X_i до его ближайшего соседа в $\{Y\}$

$$\nu_m(i) = \min_{j=1,\dots,m} \left\| X_i - Y_j \right\|.$$

Тогда дивергенцию можно приближенно оценить следующим образом

$$\hat{D}_{n,m}(p \| q) = \frac{d}{n}\sum_{i=1}^{n} \log \frac{\nu_m(i)}{\rho_n(i)} + \log \frac{m}{n-1}. \qquad (4)$$

Формула (4) дает возможность оценить дивергенцию между двумя случайными многомерными изображениями. Для того, что бы воспользоваться ей, нужно выбрать некие функционалы, описывающие изображение. В качестве таких функционалов, в данной работе выбраны двумерные функционалы SPAM [4], которые представляют собой оценки переходных вероятностей между соседними пикселями по 8-ми направлениям: $\{\leftarrow,\rightarrow,\uparrow,\downarrow,\nwarrow,\searrow,\nearrow,\swarrow\}$. Далее приведена формула вычисления функционалов для одного направления, для изображения I размером $n_1 \times n_{12}$:

$$C^{X,\rightarrow}_{d_1,d_2} = \Pr(D^{\rightarrow}_{i,j} = d_1, D^{\rightarrow}_{i,j+1} = d_{12}), \qquad (5)$$

где $-T \le d_1 \le T, -T \le d_{12} \le T$, а $D^{\rightarrow}_{i,j} = I_{i,j} - I_{i,j+1}$, $1 \le i \le n_1, 1 \le i \le n_2 - 1$.

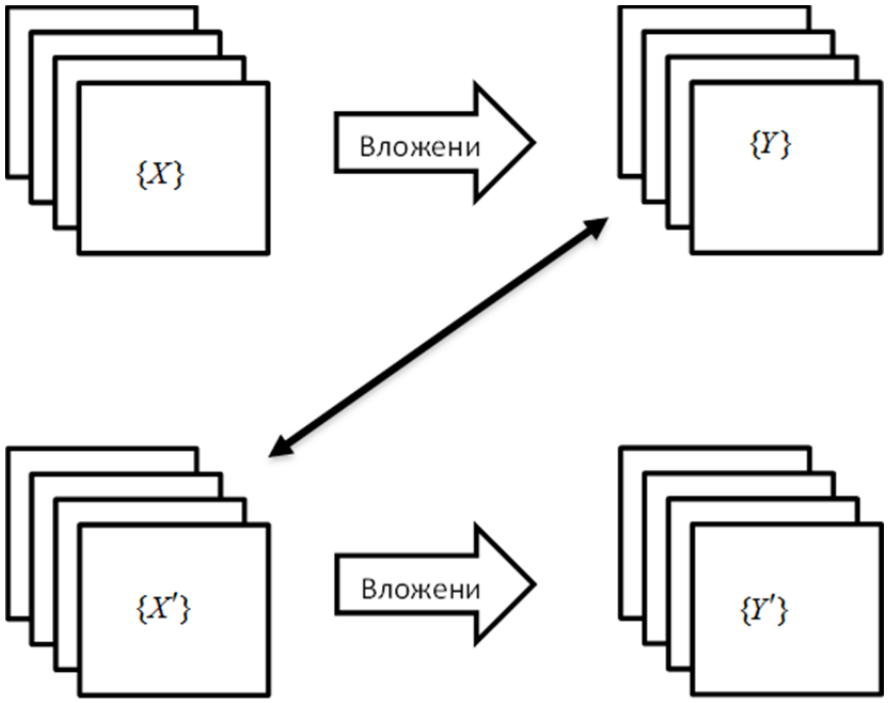


Рисунок. Схема эмпирического вычисления дивергенции набора изображений

В данной работе, предлагается такая модель вычисления дивергенции изображений, при которой некий набор изображений, делится случайным образом на два множества {X} и {X'}, с равным количеством изображений в каждом множестве. Затем, в оба множества производится вложение, и как показано на рисунке, дивергенция вычисляется между двумя совершенно разными множествами

изображений: ПС из множества {X'} и СГ множества {Y}. При этом {Y} представляет собой {X} после вложения стегосообщения.

На основе модели, приведенной на рисунке, производился эксперимент со следующим параметрами: использовалось 40000 изображений в формате pgm размером 128x128 пикселей. Вложение производилось при помощи таких широко известных алгоритмов, как LSB replacing, LSB matching и ШПС. Для ШПС использовалось 3 различных варианта амплитуды, с которой производится вложение. Для каждой стегосистемы, вероятность вложения варьировалась от 0 до 1.

ТАБЛИЦА. Значения дивергенции для различных стегосистем с различными вероятностями вложения

Вероятность вложения	D(X'||Y) при LSB replacing	D(X'||Y) при LSB matching	D (X'||Y) при ШПС (1)	D (X'||Y) при ШПС (3)	D (X||,Y) при ШПС (5)
0	63.1827	63.1827	63.1827	63.1827	63.1827
0.1	81.2901	77.7251	99.7108	181.1013	147.8558
0.5	212.4426	180.7910	297.7161	538.7343	459.8349
1	366.9323	297.8719	396.9119	510.5257	518.1709

Как видно из таблицы, вычисленные значения дивергенции ведут себя в соответствии с теорией. Таким образом, работа показывает, что вычисление дивергенции по методу ближайшего соседа может быть использовано в целях расчета вероятности ошибки оптимального стегоанализатора. Однако следует отметить, что для реального использования дивергенции, необходимо изменить модель расчета дивергенции таким образом, что бы при нулевой вероятности вложения, дивергенция была бы близка к нулю. Этому будет посвящено дальнейшее исследование.

Библиографический список

1. **Cover, T. M.** Elements of Information Theory / T. M. Cover, J. A. Thomas. – John Wiley & Sons, Inc. – 1991. – 541 p.

2. **Cachin, C.** An information – theoretic model for steganography / C. Cachin // LNCS, 1525. 1998, – PP. 306–318.

3. **Wang, Qing** A Nearest-Neighbor Approach to Estimating Divergence between Continuous Random Vectors / Qing Wang et al. // ISIT, Seattle – 2006. – PP. 242–246.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

4. **Pevny, T.** Steganalysis by substractive pixel adjacency matrix / T. Pevny, P. Bas, J. Fridrich// Proceedings of the 11th ACM Multimedia & Security Workshop, 2009. – PP. 75–84.

Аннотация

В работе рассматриваются вопросы практической оценки эффективности стегосистем. Для этого применяется математический аппарат вычисления дивергенции по методу ближайшего "соседа". Данный метод, позволит в перспективе, оптимизировать стегосистему против идеального стегоанализатора, несмотря на то что такого стегоанализатора еще не существует.

I. Fedyanin

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

ABOUT POSSIBILITY OF APPLICATION DIVERGENCE ESTIMATION METHOD BASED ON NEAREST-NEIGHBOR APPROACH FOR STEGOSYSTEMS OPTIMIZATION

Annotation

Questions of stegosystem performance estimation are considered in this paper. It is impossible to use standard methods of divergence calculation because real images have very complex distribution. In this work, application of nearest-neighbor approach towards calculation divergence of real images is investigated. In prospective, this method will make possible to optimize stegosystem against the best possible steganalyzer, even though such steganalyzer does not exist yet.

Keywords: steganography, divergence, image processing, relative entropy.

Федянин Иван Алексеевич – аспирант кафедры защищенных систем связи Федерального государственного образовательного бюджетного учреждения высшего профессионального образования «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича», ivan.a.fedyanin@gmail.com

Статья представлена рецензентом д-ром техн. наук, профессором В. И. Коржиком

UDC 004.722; 004.715

A. O. Tiamiyu

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications

SELECTION AND RATIONALE OF ALGORITHM FOR NETWORK TOPOLOGY DETERMINATION IN THE INTEREST OF TRUSTED ROUTING

network topology, route discovery, georeferencing, routing table, host.

Introduction

Understanding, development and application of secured ways of delivering data over IP networks from the sender to the recipient mitigating attacks of all kinds to maintain data integrity and confidentiality as well as its availability are nowadays of topmost priority considering the ever-increasing number of Internet users, web applications and industrial automation. All these and others expose hundreds of millions of people and organizations worldwide to treats and risks since the Internet is vulnerable to various type of attacks like cryptographic attacks, injection attacks etc.

Though routing is the process of selecting paths in a network along which to send data, yet under an unmonitored scenario, the network essentially breaks into components with similar interest groups. And as such, securing data is inherently coupled with securing the network from self-interest groups [1]. Thus, there is necessity for trusted routing (TR), which is the process of planning the path that data follows when traversing across multiple networks from its source to its destination without possibility of the node(s) within any of the networks tampering with the data in any form [2]. However, method of TR consists of many stages among which is network topologies determination. Network topologies can be determined using routing tables (RT) obtained from different routing devices (RD) over a network in conjunction with results from various network-tracing algorithms (NTA) which are also systematic approach to obtain other information about routers within a network (e.g. route, georeferencing as well as other information of interest that is contained therein). Furthermore, the results of the tracing can be used for running additional discoveries and/or decision-making that are required in the processes of finding trusted route. Among many NTA and RT access methods that can be used jointly or independently in determining network topologies are NTA with the help of

standard tools, NTA with the help of IP-packet option, RT access algorithm using SNMP, RT access algorithm using BGP and NTA by analyzing RT. Each of these NTA or RT access methods is implemented to get enough information into the database for analytical network mapping approach for determining topology of a network. It is assumed that TRM (trusted routing manager, a software program for trusted routing management and traffic flow control) is installed on the management automated workstation, and that TRA (trusted routing agent that is normally embedded into RD in the interests of TR) is present (i.e. already embedded) in some of the RD of the network which topology is to be determined.

1. Network-tracing algorithms

1.1. NTA with the help of standard tools

There exists many standard utilities for tracing path to the source of data in a network, e.g. PING, traceroute, tracert etc. Among them traceroute is to some extent the most popular mechanism for computing the topology of a network in the Internet [3]. The technique is using ICMP messages to determine the route. Traceroute, like other standard tools, uses two techniques: small values of TTL and invalid port number (33434) to trace packet route to the destination. Though the TTL field of IPv4 has been renamed to Hop Limit in IPv6, the message format, basically, is the same for ICMPv4 and ICMPv6. Regardless of the name, the field still has the same basic purpose of keeping a datagram from wandering the Internet forever. Diagram of the algorithm is shown in Figure 1.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

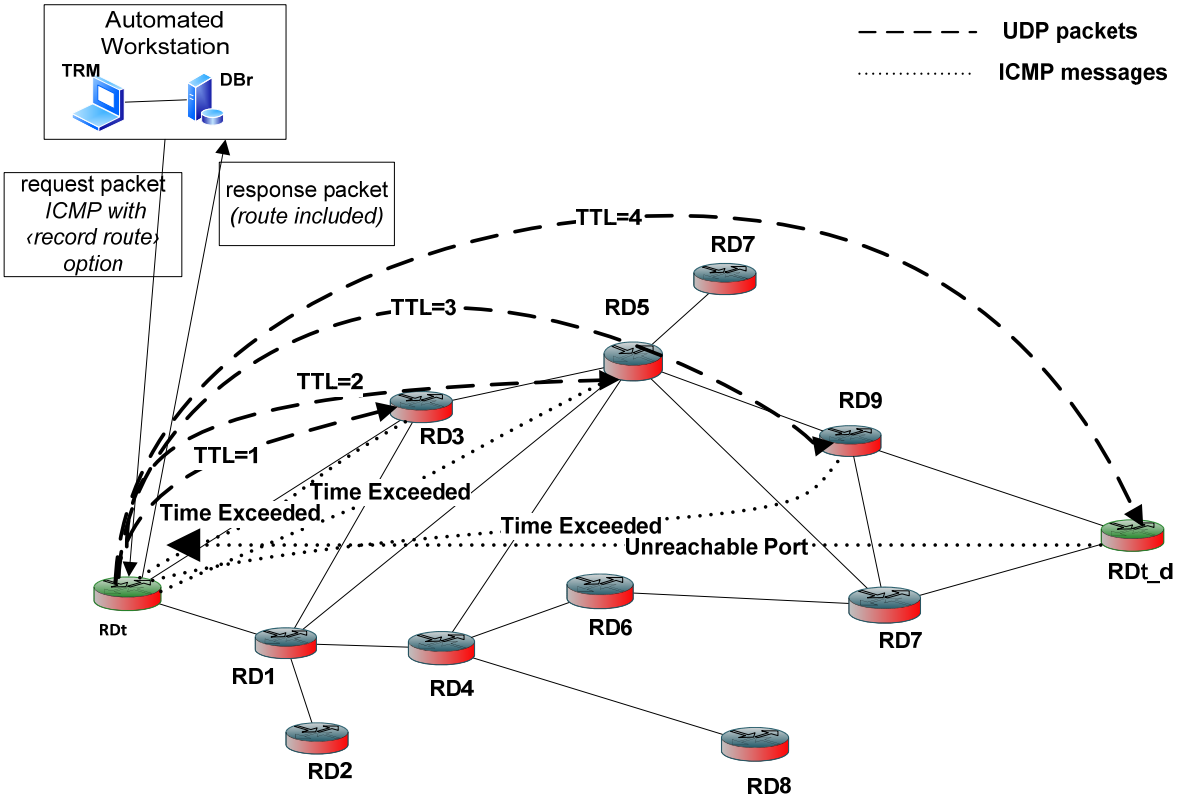


Figure 1. Diagram of the Information Flow of NTA using a Standard Tracing Tool

NTA with the help of standard tools is implemented as follows:

1) TRM sends to RDt a special packet when the standard tool like, for instance traceroute on UNIX, runs with the IP-address of the recipient and option ‹record route› (router RDt_d in this case, as seen in Figure 1). RDt then starts sending packets with a small lifespan. The counter value starts at 1 and increases by 1 for each group of three UDP packets. The first router, RD3, that receives the packet, decrements the TTL value to zero and returns ICMP-message ‹Time Exceeded› to RDt.

2) Having received a packet, each transit router decreases TTL value by 1. Whenever TTL=0, packet is not sent further and RDt receives ICMP message ‹Time Exceeded› from the transit router. From this ICMP message, ‹Time Exceeded›, TRA discovers the IP address of the router.

3) This cycle is repeated until the receiving host (router RDt_d) receives a packet from RDt, to which it returns an ICMP-message ‹Unreachable Port›, indicating that the packet reached the destination, thus, end of the trace.

4) Having obtained information about the IP-addresses of all the routers located between the source and the specified destination, a list of intermediate

routers is built, starting from a distance of one transition and ending this with the destination host. Response packet containing the route (for example, route RDt⇒RD3⇒RD5⇒RD9 ⇒RDt_d as seen in Figure 1) is generated and sent to TRM by RDt.

5) TRM records the information into the database of active routes (DBr) for further analysis and preparation of the active route.

1.2. NTA with the help of IP-packet option

IP-packet NTA is for finding the source of a packet by probabilistically marking packets with partial path information as they are forwarded by routers [4]. In Ipv4, this is based on parameter 7 ‹record route› field option. The ‹record route› option, when included in the IP packet option, provides a means to record the route, and this gives, along with the packet, the recipient a list of all transit routers, traversed by the packet. As such, the algorithm is suitable for obtaining the list of IP-addresses of all the transit routers between two RDt (RD with TRA embedded i.e. as implant), for example, between RDt and RDt_d (RDt_d is the recipient) as shown in Figure 2, the diagram of realization of the algorithm. Obvious that IPv4 options perform a very important role in the IP protocol operation, therefore the capability is preserved in IPv6. However, the functionality of options is removed from the main header and implemented through a set of additional headers called extension headers (EH) [5]. All recorded data is written into the Data Space that is preallocated in the ipv6 record route (RR6) option header. Packets, in which the RR6 option is set, because the Data Space exists in it, carry the recorded data.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

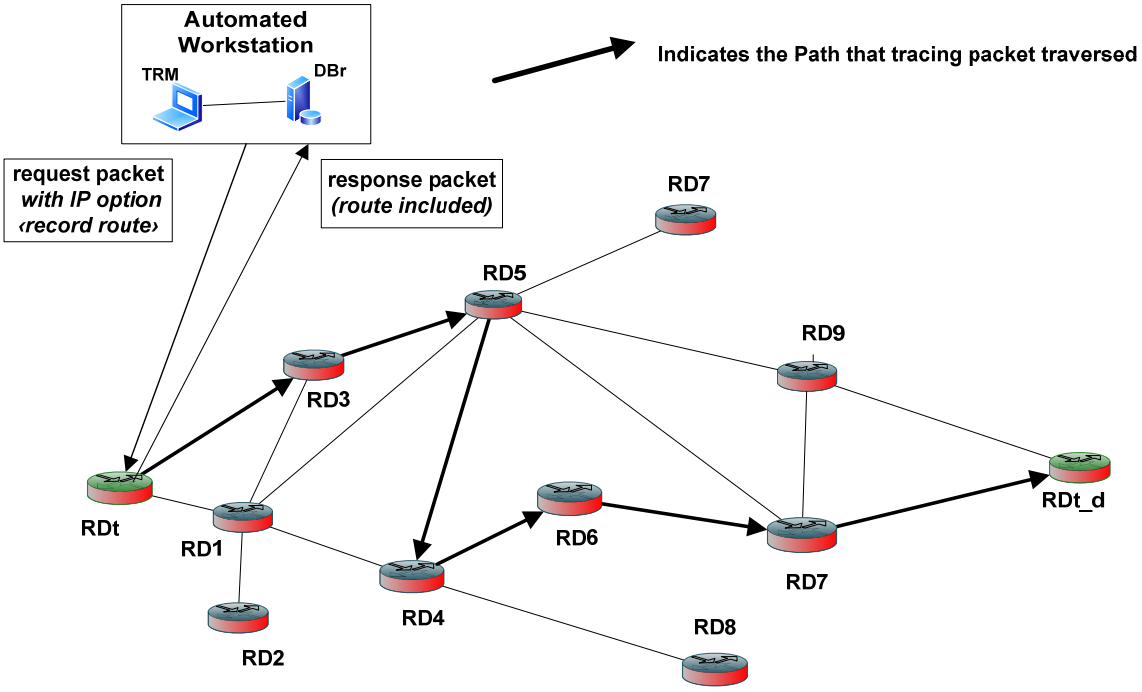


Figure 2. Diagram of Realization of NTA using IP-packet option

This NTA using IP-packet option is implemented as follows:

1) After should-be controlled part of the network is defined, trusted routing manager (TRM) sends to RDt special packet (in which IP-packet option ‹record route› is enabled) requesting to record route from itself to a specified destination IP address, e.g. a particular RDt_d.

2) RDt, having received the ‹request› packet, sends it to RDt_d. Each router through which the packet traverses records its IP-address into the packet and sends the packet further to the next router, until the packet reaches the destination router RDt_d.

3) RDt_d preprocesses the trace result i.e. records of the resulting route (for example, route RDt⇒RD3⇒RD5⇒RD4⇒RD6⇒RD7⇒RDt_d as seen in Figure 2), generates and sends a special response packet containing the result via a special communication channel to RDt.

4) On getting this response that contains IP addresses of all transit routers and that of RDt_d from RDt_d, RDt sends it to TRM where it is recorded into the DBr for further analysis and processing.

5) Stages 2 through 4 are repeated as many times as active routes are needed.

1.3. RT Access Algorithm using SNMP Protocol

Though manufacturers can independently choose protocols that are used to configure network devices, however for inter-operability among devices, SNMP agent is usually embedded in the network devices. These SNMP agents are sometimes addressed using MIB-II as information available from them is available in a standardized form. Route is an object of MIB and SNMP can be configured over IPv6 transport so that an IPv6 host can perform SNMP queries and receive information about RT from router running IPv6 software since SNMP agent and related MIBs have been enhanced to support IPv6 addressing [6].

This RT access algorithm using SNMP protocol is for obtaining information contained in the RT of router and it is implemented algorithmically as follows:

1) TRM sends to RDt packet ‹Enable SNMP›. Having received the packet, RDt generates request packet according to the standard procedure of SNMP protocol, (1.3.6.1.2.1.4.21 – ipRouteTable and 1.3.6.1.2.1.4.21.1 – ipRouteEntry) where iso(1), identified-organization(3), dod(6), internet(1), mgmt(2), mib-2(1), ip(4), ipRouteTable(21) and ipRouteEntry(1).

2) RDt then connects with router via SNMP and starts recording information about RT from the router. On starting, RDt connects with the router and receives response from the router with information about the destination, ‹ipRouteDest›, and then sends ‹get-next› request to the router to get the next parameter: interface index ‹ipRouteIfIndex›. By sequentially sending request ‹get-next› and receiving response from the router, RDt receives detailed information of all the records (for example, IP-address destination, destination interface, next-hop, protocol code, route mask etc) that are located in the RT of the router.

3) RDt then generates a response message containing information about RT of the router, and sends it to TRM.

4) This cycle is repeated for other routers.

5) In TRM, received messages containing information about RT are recorded into the DBr for further analysis and for preparation of the active route of a particular packet. Diagram of the method is shown in Figure 3.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

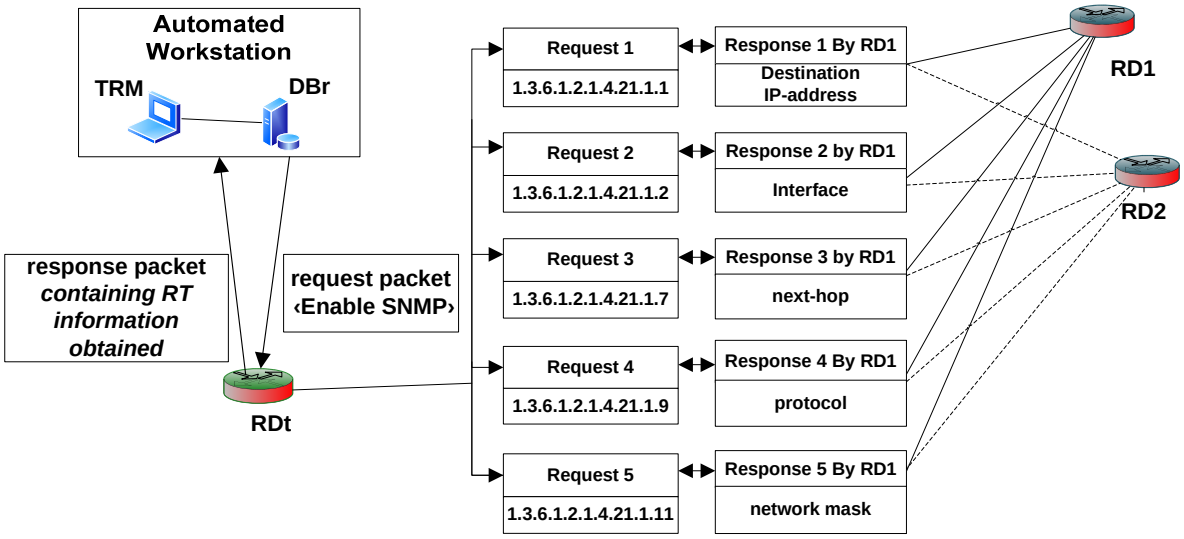


Figure 3. Diagram of RT Access Algorithm using SNMP protocol

1.4. RT Access Algorithm using BGP

BGP protocol is usually used between gateway routers on the Internet for exchanging information like a list of known routers, available routes etc. RT Access Algorithm using BGP is designed to obtain information contained in the RT of routers by implementing access to them via BGP. The RT Access Algorithm using BGP is implemented as follows:

1) TRM sends packet ‹Enable BGP› to RDt requesting it to open BGP-session.

2) After RDt opens BGP-session with router successfully (for example, with any of eBGP routers shown in Figure 3), RT information exchange with RDt begins with the help of packet ‹update›. On the RDt, the control fields are filled: IP – address of the router; Version – BGP version; AS – autonomous system; hold time – 180.

3) RDt effects primary assessment and analysis of RT information contained in the packets «update» and generates a special response packet that contains information obtained (information like Host BGP IP, Unfeasible Routes, Length NextHop, Next Hop, Length ASPPath, ASPPath, Networks etc.), and sends it to TRM.

4) BGP-session with router is maintained until signal to close the BGP-session comes from TRM

5) Information from RDt about RT of all the routers is sent to TRM for further processing and realization of analytical method for detection of route.

BGP-sessions with several routers can be opened simultaneously thereby start getting their RT information as well. Diagram of the algorithm is shown in Figure 4.

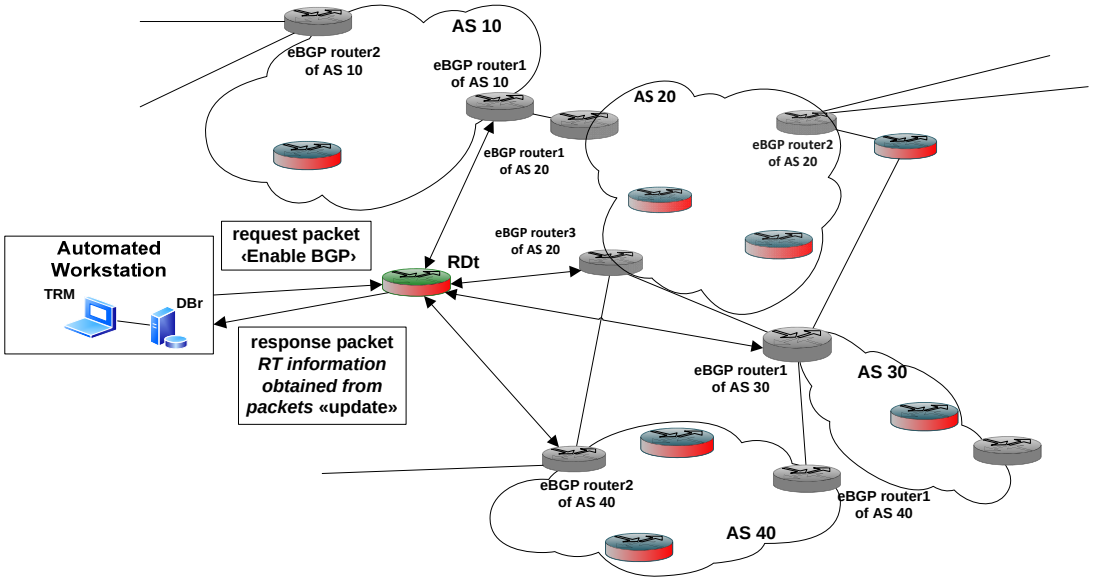


Figure 4. RT Access Algorithm using BGP

1.5. NTA by Analyzing RT

NTA by Analyzing RT is designed to trace information flow analyzing RT obtained through the implementation of RT access methods using different protocols. Thus, this algorithmic method is designed for the analytical determination of active routes of information flows (list of active IP-addresses of transit routers) by analyzing information contained therein RT.

NTA by Analyzing RT is implemented as follows:

1) A sequential access to RD using a given protocol in order to obtain information about the RT is performed, and information about RT obtained through a variety of access methods to RT re-presented in a unique formalized view and then recorded into DBr;

2) Analysis of RT in order to obtain route of a specific packet is implemented by selecting initial IP-address (fixing starting point for the route), and then identify a row in the formalized RT using the selected initial IP-address;

3) Analyzing RT, precisely by the starting point of the route, IP-address (Next hop) of other transit routers is determined;

4) Search for active IP-addresses of intermediate routers is performed iteratively; and this gives route of a particular packet, which is then recorded into the DBr.

Flowchart of this analytical tracing algorithm method is shown in Figure 5.

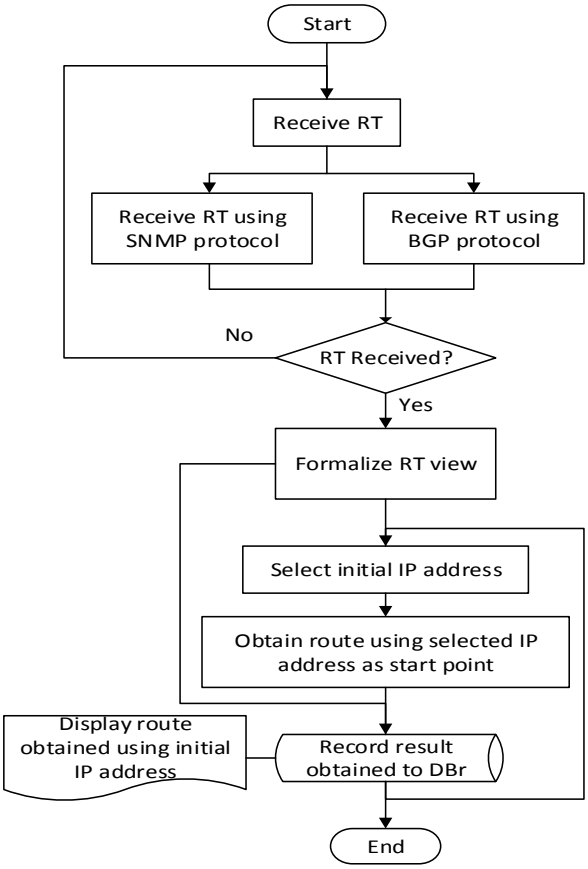


Figure 5. Flowchart of Analytical NTA using RT

1.6. Network Topology Mapping

Though all these NTAs and RT access algorithms discover links between interfaces on Internet routers, however, some of these IP addresses belong to the same router. Thus, topology mapping, for example, Internet topology mapping, requires IP aliases resolution. IP aliases resolution is the process of recognizing interfaces that belong to the same router and it helps revealing true topology [7–9]. Many techniques of aliases resolution are discussed in [7] that can produce the most accurate and complete IP-to-router mapping. According to [7], some of

the techniques e.g. common source address, graph analysis, TTL constraints and missing middle produces a better set of results when combined. Also combination of standard utility like traceroute and IP option ‹record route› probes improves the accuracy of the determined network topology. To have a true view of the network topology, the existing IP aliases among the results that are stored in DBr are resolved.

2. Pros and Cons of the NTAs

2.1. Pros and Cons of NTA with the help of Standard Tools

Realization of NTA with the help of standard tools is fast. Its usage is simple and it allows determining constituting elements of the network that is being traced, as well as determining its topology that results in having a map of links with which the situation in the network can be successfully diagnosed. Notwithstanding, the usage of this algorithm is possible only if «block incoming ICMP-messages» is not enabled or the outgoing ICMP-messages ‹Time Exceeded› is not being filtered.

2.2. Pros and Cons of NTA with the help of IP-option

NTA with the help of IP-option is designed to obtain information about the IP-addresses of all the routers located between the source and the specified destination, which allows getting the record of the active route of traffic flow. Furthermore, there is possibility of implementing this algorithm using standard methods and realization is also fast. Using this algorithm the received information is reliable and can be transferred via a secure communication channel. In addition, it allows TRA working on request as well working as programmed by ‹timer›. Its drawbacks are that agreement between the source and the receiver on enabling the option ‹record route› must exist, and that the quantity of recorded IP-addresses is limited by the size of option field ‹data› parameter in IPv4 unlike in IPv6 where Data Space (though can accommodate far more compared to the case of IPv4) of the option header limits it.

2.3. RT Access Algorithm using BGP

RT Access Algorithm using BGP makes it possible to get full active RT of connected router, detailed information of which allows its usage as input for the implementation of analytical method for obtaining route of a particular packet and network topology mapping. RT information of some connected routers can simultaneously be gotten and transferred from RDt to TRM. However, all these are possible provided BGP protocol is enabled on the router and BGP-session with routers is supported.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

2.4. RT Access Algorithm using SNMP

Pros and cons of RT Access Algorithm using SNMP is very similar to that of RT Access Algorithm using BGP. However, in addition to enabling of SNMP on the participating routers, RT access algorithm using SNMP also faces challenge of route information obsolescence. Furthermore, route information accuracy in this algorithm is determined based on the authenticity of MIB information.

2.5. NTA by Analyzing Routing Table

NTA by Analyzing RT is for analytical determination of active routes of information flows and it gives complete, both active and backup, routes of a specific packet. Its disadvantages is that complete information about all RT exists not always and obsolescence of information in RT occurs. Furthermore, effectiveness of the algorithm depends on effectiveness of RT access algorithm that is being used.

Conclusion

Despite the disadvantages associated with these NTAs and RT access methods described, they can be used in the process of getting trusted route since in TR mechanism it is far more important to have valid trusted route(s) than to have a complete map of the network e.g. map of Internet. Notwithstanding they help in network topology mapping and they can be used as components of the methodology or software development for effective determination of the paths of traffic flows and georeferencing. Furthermore, they can be used to obtain other information contained therein the RD, thus, aiding in the determination of trust level of RD, one of the stages in the processes of TR. As such, they are vital in the process of TR.

References

1. **Path** Optimization and Trusted Routing in MANET: An Interplay Between Ordered Semirings / K. K. Somasundaram, J. S. Baras // Advances in Networks and Communications. – Springer, 2011. – PP. 88–98.

2. **Overview** of Modern Telecommunication Networks Security Challenges / A. O. Tiamiyu // Scientific Journal: National Security and Strategic Planning, 2013. – № 2 (2). – PP. 37–43.

3. **Detection**, understanding, and prevention of traceroute measurement artifacts / F. Viger, B. Augustin, X. Cuvellier, C. Magnien, M. Latapy,

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

T. Friedman & R. Teixeira // Computer Networks, 2008. – № 52 (5). – PP. 998–1018.

4. **IP traceback** based on deterministic packet marking and logging / X. J. Wang, Y. L. Xiao // Eighth International Conference on Embedded Computing ‹‹Scalable Computing and Communications››. – IEEE, 2009. – PP. 178–182.

5. **Record** Route for IPv6 (RR6) Hop-by-Hop Option Extension / H. Kitamura // Internet Engineering Task Force. – 2000. – Mode of access: http://tools.ietf.org/pdf/draft-kitamura-ipv6-record-route-00.pdf. (Date accessed 20.02.2014).

6. **IPv6** Configuration Guide: Cisco IOS XE Release 3S / Cisco Headquarters. – Cisco, 2012. – 54 p.

7. **Internet-scale** IP alias resolution techniques / K. Keys // ACM SIGCOMM Computer Communication Review, 2010. – № 40 (1). – PP. 50–55.

8. **Yet another** active probing technique for alias resolution / P. Marchetta, V. Persico & A. Pescapé // ACM CoNEXT, 2013. – PP. 229–234.

9. **Analytical** IP alias resolution / M. H. Gunes & K. Sarac // IEEE International Conference ‹‹ICC'06››. – IEEE Communications, 2006. – № 1. – PP. 459–464.

Annotation

One of the stages in the process of application of trusted routing method at getting information across global data network safely and securely is determination of the network topology. This paper describes the selection and rationale of algorithm for network topology determination in the interest of trusted routing.

Тиамийу А. Осуолале

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

ВЫБОР И ОБОСНОВАНИЕ АЛГОРИТМА ОПРЕДЕЛЕНИЯ ТОПОЛОГИИ СЕТИ В ИНТЕРЕСАХ ДОВЕРЕННОЙ МАРШРУТИЗАЦИИ

Аннотация

Одним из этапов в процессе применения метода доверенной маршрутизации при передаче информации через глобальную сеть данных надежно и безопасно является определение топологии сети. Эта статья описывает выбор и обоснование алгоритма для определения топологии сети в интересах доверенной маршрутизации.

электронный научный журнал ISSN 2307-1303
© информационные технологии
и телекоммуникации, выпуск 1 за 2014 год

Ключевые слова: топология сети, определение маршрута, геопривязка, таблица маршрутизации, хост.

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

Статья представлена рецензентом д-ром техн. наук,
профессором Буйневичем М. В.

НАУЧНОЕ ИЗДАНИЕ

Информационные технологии и телекоммуникации

Электронный научный журнал

Выпуск 1-2014

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 01.03.2014
Вышло в свет 30.03.2014
Уст. объем 4,25 печ. л. Заказ № 006-ИТТ-2014
191186, СПб, наб. р. Мойки, 61
E-mail: telecomsut@gmail.com
www.itt.sut.ru