

Заключение

Из рассмотренного материала можно сделать следующие выводы:

1. Обоснованы условия взаимоднозначного восстановления исходных элементов трехмерных полиадических чисел. Доказана теорема о существовании системы выражений, обеспечивающих трехмерное полиадическое декодирование без внесения погрешности. На основе доказанной теоремы разработано поэлементное последовательное восстановление элементов трехмерных структур данных.
2. Разработано поэлементное рекуррентное трехмерное полиадическое декодирование без внесения погрешности. Восстановление элементов трехмерных полиадических чисел организовывается в направлении по вертикалям сверху – вниз, по столбцам – в глубину параллелепипеда и по строкам слева – направо. Декодирование кода-номера осуществляется с учетом того, что первый элемент ТПЧ является старшим. В отличие от последовательного поэлементного восстановления для получения одного элемента требуется выполнить только одну операцию деления. В этом случае суммарное количество операций дополнительно сокращается на 20%. Суммарное время восстановления трехмерной структуры данных в случае рекуррентной

обработки снижается в 1,25 раза относительно поэлементной обработки.

Литература: 1. *Ватолин В.И., Ратушняк А., Смирнов М., Юкин В.* Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М.: ДИАЛОГ – МИФИ, 2002. 384 с. 2. *Зубарев Ю.В., Дворкович В.П.* Цифровая обработка телевизионных и компьютерных изображений. М.: Международный центр научной и технической информации, 1997. 212 с. 3. *Баранник В.В.* Метод трехмерного кодирования данных // Системы обработки информации. Харків: ХВУ, 2003. Вип. 1. С. 42–46. 4. *Баранник В.В., Королева Н.А.* Иерархически-последовательная организация восстановления изображения // Радиоэлектроника и информатика. 2002. №1(18). С. 77–81. 5. *Королев А.В., Баранник В.В., Гиневский А.М.* Иерархически-конвейерная организация восстановления изображений // Збірник наукових праць ІПМЕ НАНУ. К.: ІПМЕ НАНУ. 2002. Вип. 15. С. 27–33.

Поступила в редколлегию 09.09.2007

Рецензент: д-р техн. наук, проф. Поляков П.Ф.

Баранник Владимир Викторович, д-р техн. наук, старший научный сотрудник Харьковского университета Воздушных Сил. Научные интересы: обработка и передача информации. Адрес: Украина, 61023, Харьков - 23, ул. Сумская, 77/79, тел. 704–96–71.

Карпенко Сергей Владимирович, доцент кафедры безопасности информационных технологий, НАУ. Научные интересы: обработка и передача информации.

УДК519.713

МЕТОД ОБНАРУЖЕНИЯ ОТКАЗОВ МАРШРУТИЗАТОРОВ В КОМПЬЮТЕРНЫХ СЕТЯХ¹

САЕНКО В.И., АЛЕКСЕЕВ Д.И.

Рассматривается метод обнаружения отказов маршрутизаторов в компьютерной сети произвольной топологии с помощью построения логических деревьев компьютерной сети. Метод предназначен для использования в системах непрерывного мониторинга и основан на решении логических уравнений состояния.

1. Описание проблемы и анализ известных результатов исследований

Корпоративные компьютерные сети требуют реализации процедур постоянного мониторинга для обеспечения функциональной поддержки основных ресурсов и сервисов и обеспечения стабильности показателей QoS для всей сети в целом. К сожалению, технологии постоянного мониторинга оказывают большую нагрузку на компьютерную сеть и снижают эффективность ее использования. К задачам постоянного мониторинга относятся задачи обнаружения отказов маршрутизаторов, а также задачи поддержки функций QoS. Реализация процедур экономного мониторинга является актуальной для компьютерной сети в целом.

¹ – Исследования проводились в научно-исследовательской лаборатории «Менеджмент компьютерных сетей» (NM Lab) в ХНУРЭ.

Общая цель мониторинга – сбор пакетов в различных точках сети и получение информации для раскрытия пути любого пакета [7].

При построении систем мониторинга могут использоваться различные технологии и принципы сбора данных. Наиболее распространенными являются технологии на основе SNMP [9], RMON [9], WBEM [10]. Они позволяют применять принципы построения иерархических структур [1, 9] и обеспечивать высокую гибкость систем менеджмента. С помощью этих технологий могут быть реализованы как централизованные, так и децентрализованные модели мониторинга компьютерной сети.

Однако и централизованные, и децентрализованные модели мониторинга сети получают данные для обработки путем опроса каждого узла, в частности, каждого маршрутизатора в сети, что существенно сокращает возможность сети обеспечить необходимые для большого количества потоков ресурсы. Периодические опросы или размещение агентов на высокоскоростных магистральных маршрутизаторах создают высокую нагрузку на сеть и на сами маршрутизаторы. Чрезмерный трафик, генерируемый неисправными службами, изменяет внутренние характеристики компьютерной сети, через которую он проходит.

Возникает проблема синтеза архитектуры процесса мониторинга и формирования соответствующих эффективных методов диагностики [3].

Существуют различные пути ее решения. Например, системы могут быть построены на основе мобильных агентов [7] или с программируемой архитектурой [2]. В [8] предложено создание логических представлений компьютерной сети, покрывающих реальную компьютерную сеть, путем соединения всех граничных маршрутизаторов поставщика услуг.

Другой подход к опросу – это использование механизма событий, который обычно предоставляет информацию для использования после того, как изменилось состояние элемента сети [4].

Хорошие результаты может давать граничная технология построения систем мониторинга, предложенная в [5]. Этот механизм мониторинга работает в доменах QoS, используя так называемую сетевую томографию [5].

Наиболее удачной является технология мониторинга, основанная на реализации процедур выборочного тестирования маршрутизаторов [6]. При этом предлагается формировать специальные логические деревья и тестировать их, решая логические уравнения [6].

Эти методы, к сожалению, позволяют осуществлять только общую диагностику состояния узлов или предъявляют слишком высокие требования к пропускной способности сети или к хранилищам данных, собранных в процессе мониторинга.

Целью исследования является разработка путей реализации экономически эффективного постоянного мониторинга при выявлении отказов передачи данных в корпоративной сети.

2. Постановка задачи и описание объекта исследования

Описание объекта. Пусть существует некоторая многосегментная компьютерная корпоративная сеть. Требования корпоративности означает, что в сети существует единая политика, разрешающая использование специальных агентов и технологий для тестирования маршрутизаторов и рабочих станций. В такую сеть входят рабочие станции, серверы, внутрисегментные маршрутизаторы и внешние относительно сегментов маршрутизаторы.

В качестве логической топологии сети выбирается древовидная структура представления компьютерной сети. Такая структура позволяет определить узлы, которые участвуют в прохождении пакетов (узлы источника, назначения и промежуточные). Для приведения сети сложной топологии к древовидной можно произвести декомпозицию начальной сети на несколько деревьев, не имеющих петель.

Согласно [6], представим компьютерную сеть в виде древовидного графа, узлами которого являются корневые маршрутизаторы – маршрутизаторы,

образующие магистраль $\{C_1, C_2, \dots, C_5\}$, а вершинами (листьями) – граничные маршрутизаторы – маршрутизаторы, находящиеся на границе локальных сетей $\{E_1, E_2, \dots, E_7\}$ (рис.1).

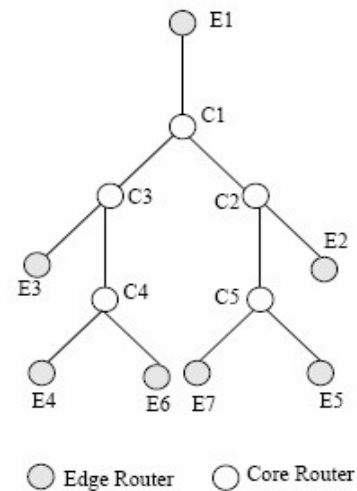


Рис. 1. Граф логической сети

Пусть $\{R_j\}$ – множество всех маршрутизаторов, тогда $\{C_1, C_2, \dots, C_5\} \subset \{R_j\}$, $\{E_1, E_2, \dots, E_7\} \subset \{R_j\}$ и $\{C_1, C_2, \dots, C_5\} \cup \{E_1, E_2, \dots, E_7\} = \{R_j\}$. Введем множество $\{Q^-\}$ – сбойные маршрутизаторы. В случае стабильно работающей сети имеем $\{Q^-\} = O$.

В сети присутствуют сквозные и внутренние потоки данных. Сквозные – потоки, проходящие транзитом сквозь данную сеть; внутренние – входящие и исходящие потоки данных, инициаторами или приемниками которых являются внутренние компьютеры или маршрутизаторы данной сети.

Введем специальные понятия: *путь* $path(i, j)$ – это связь между граничными маршрутизаторами без учета корневых маршрутизаторов; *связь* (i, j) – это соединение двух соседних маршрутизаторов, т.е. ребро графа; *направление связи* – направление передачи данных; *перегрузка связи* – состояние связи, когда потери пакетов на этой связи превышают установленный предел. При этом будем различать передачу данных в разных направлениях и в общем случае $(i, j) \neq (j, i)$.

Структура сети без учета магистральных маршрутизаторов и соответствующие «пути» представлены на рис. 2.

Структура сети с указанными связями между всеми маршрутизаторами представлена на рис.3. Связи между маршрутизаторами и рабочими станциями на данном этапе можно проигнорировать, поскольку потоки, циркулирующие между станциями и маршрутизаторами, не повлияют на работу всей сети в целом.

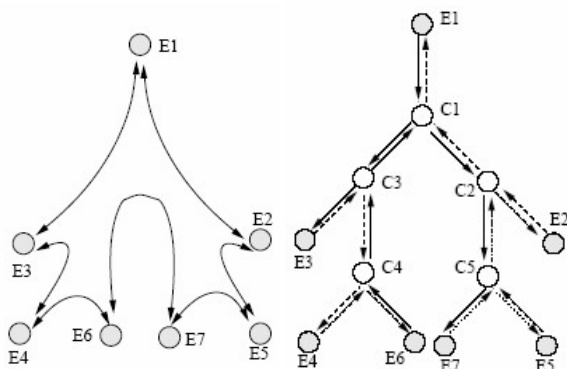


Рис. 2. Пример путей в сети

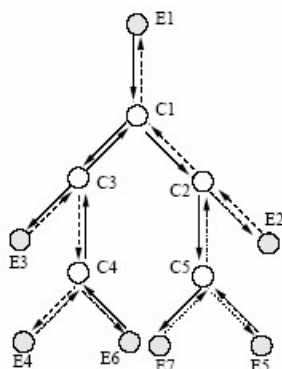


Рис. 3. Связи в сети

Постановка задачи сводится к разработке методов выявления отказов связей (сегментов) и маршрутизаторов в компьютерной сети.

3. Методологические основы проведения мониторинга при выявлении отказов маршрутизаторов

Рассмотрим два метода диагностики: простой метод определения перегруженных связей и метод выявления отказавшего узла в проблемном сегменте (связи).

При этом вводятся переменные состояния.

Простой метод выявления отказавших сегментов (А).

Метод основан на предложениях, представленных в [6]. Пусть в компьютерной сети выделены логические фрагменты, для которых осуществляется тестовая проверка (рис. 4).

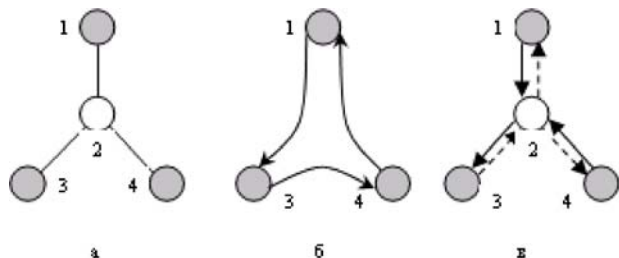


Рис. 4. Схема топологической структуры моделируемого участка сети

Пусть $P_{i,j}$ – булева переменная, которая представляет собой результат проверки состояния пути между граничными маршрутизаторами от i к j . Она равна 1, если связь перегружена, и равна 0 при нормальном состоянии связи. Также $P_{i,j} = 0$, если $i = j$:

$$P_{i,j} = \begin{cases} 1, i \neq j; \\ 0, i = j. \end{cases} \quad (1)$$

Пусть $X_{i,k}$ – булева переменная, представляющая собой состояние перегрузки для связи (i, k) , назовем ее переменной перегрузки. Тогда для переменной $P_{i,j}$ справедлива запись

$$P_{i,j} = X_{i,k} + X_{k,j} \quad (2)$$

(см. рис.4), если для каждого пути имеется один корневой маршрутизатор (k).

В общем случае для h корневых маршрутизаторов справедливо

$$P_{i,j} = X_{i,k} + \sum_{n=k}^{n=h-1} X_{n,n+1} + X_{h,j}. \quad (3)$$

В этой формуле $(+)$ представляет собой булеву операцию «ИЛИ». При этом необходимо отметить, что $X_{i,j}$ не зависит от $X_{j,i}$.

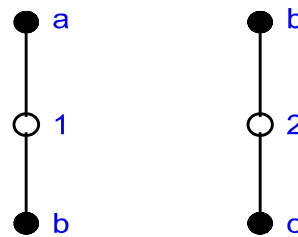


Рис. 5. Схема топологической структуры дополнительного моделируемого участка сети

Для реализации метода будем различать обход логического дерева, отображающего компьютерную сеть, по часовой стрелке и против часовой стрелки.

Обход графа против часовой стрелки является 1-м раундом проверки состояния связей, а обход по часовой стрелке – 2-м раундом проверки. Такой подход позволяет проверить все направления.

Таким образом, переменные перегрузки, полученные в одном раунде проверки, участвуют в выражениях, полученных для другого раунда проверки. В результате тестирования сегментов можно определить сегмент, в котором выявлена переменная перегрузки $P_{i,j} = 1$. Это значит, что этот сегмент является сбойным и принадлежит множеству $\{Q^-\}$.

Метод выявления отказавших маршрутизаторов в проблемном сегменте (В).

В случае если обнаружен перегруженный сегмент, необходимо выявить причину. Такой причиной может быть отказ маршрутизатора в определенном направлении. Следовательно, должна быть определена процедура выявления такого маршрутизатора. Сегмент – это канал, соединяющий два маршрутизатора, следовательно, один из них неисправен, т.е. в сегменте присутствует маршрутизатор, который можно назвать сбойным $R_i \in Q^-$.

Пусть на основании метода (А) выявлен сегмент $X_{j,i}$. Делаем предположение, что в сегменте, обслуживаемом маршрутизатором, есть рабочие станции и они функционируют нормально. Выделим дополнительно по одной рабочей станции в смежных от каждого маршрутизатора сегментах и обозначим эти станции $\{a, b, c\}$ (рис. 5).

Тогда можно сформировать два новых пути: один на основе маршрутизатора 1 {a, b}, а второй на основе маршрутизатора 2 {b, c} (рис. 6).

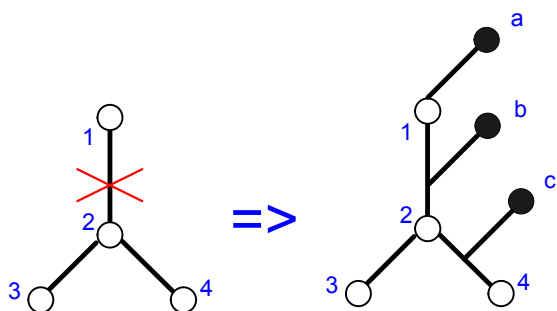


Рис. 6. Схема топологической структуры дополнительного моделируемого участка сети

Введем дополнительные переменные, которые обозначают: состояние маршрутизаторов в определенном направлении – $\{(s_{1a}, s_{a1}); (s_{1b}, s_{b1})\}$, и переменные, которые обозначают состояние рабочих станций $(q_{a1}, q_{1a}); (q_{b1}, q_{1b})$. Тогда состояние связи будет иметь вид $X_{i,j} = s_{ij} + q_{ij}$.

Состояние пути между станциями будет определено в виде системы логических уравнений состояний:

$$\begin{cases} P_{ab} = X_{a1} + X_{1b} , & P_{bc} = X_{b2} + X_{2c} , \\ P_{ba} = X_{b1} + X_{1a} ; & P_{cb} = X_{c2} + X_{2b} , \end{cases} \quad (4)$$

где $X_{a1} = s_{a1} + q_{a1}$, q_{pr} – состояние станции в направлении (p, r); s_{pr} – состояние маршрутизатора в направлении (p, r).

Для рис.5 имеем состояние

$$\{q_{a1}, q_{1a}, q_{1b}, q_{b1}, q_{b2}, q_{2c}, q_{c2}\}, \\ \{s_{a1}, s_{1a}, s_{1b}, s_{b1}, s_{2b}, s_{b2}, s_{2c}, s_{c2}\}.$$

Принимаем допущение, что станции работают нормально, следовательно

$$q_{a1} = q_{1a} = q_{1b} = q_{b1} = q_{b2} = q_{2c} = q_{c2} = 0.$$

Тогда для связей справедливо следующее:

$$\begin{cases} X_{c2} = q_{c2} + s_{c2} , & X_{b2} = q_{b2} + s_{b2} , \\ X_{2b} = q_{2b} + s_{2b} ; & X_{2c} = q_{2c} + s_{2c} ; \end{cases} \quad (5)$$

$$\begin{cases} X_{a1} = q_{a1} + s_{a1} , & X_{b1} = q_{b1} + s_{b1} , \\ X_{1b} = q_{1b} + s_{1b} ; & X_{1a} = q_{1a} + s_{1a} . \end{cases}$$

При наличии в сегменте отказавшего маршрутизатора одно из состояний пути X_{ij} будет иметь значение 1, тогда при принятом допущении о работоспособности рабочих станций ($q_{ij} = 0$) одна из переменных s_{ij} , обозначающих состояние маршрутизатора, примет значение 1. Соответственно этот маршрутизатор и является отказавшим для данного направления.

4. Описание метода обнаружения отказов маршрутизаторов в компьютерной сети

Предлагается метод мониторинга, который слабо нагружает магистральные маршрутизаторы, поскольку не использует их для проведения измерений. Предполагается, что если сеть правильно спроектирована и построена, а также если пользователи ведут себя в соответствии с правилами, то потоки, проходящие сквозь эту сеть, не испытывают больших задержек и потерь.

В предлагаемом методе все граничные маршрутизаторы исследуют своих соседей в направлениях по часовой стрелке и против часовой стрелки.

Суть предлагаемого метода в том, что схема мониторинга имеет три этапа.

На первом этапе все маршрутизаторы (точки прохождения) записывают проходящий трафик в целях исследования задержек на пути следования пользовательских пакетов. Если задержка выше, чем определенный порог, граничные маршрутизаторы запускают процесс измерения потерь. Формируется логическое дерево сети с использованием всех граничных маршрутизаторов. В результате процесса измерения не подсчитывается процент потерь для каждого индивидуального соединения, вместо этого определяются соединения, которые находятся в состоянии перегрузки, т.е. имеют большой процент потерь пакетов.

На втором этапе производится тестирование полученного дерева компьютерной сети в целях получения реальных значений для анализа существующей архитектуры. На этом этапе определяется проблемный участок сети, ограниченный парой маршрутизаторов.

На третьем этапе тестирования компьютерной сети производится уточнение проблемного узла (маршрутизатора) на полученном на втором этапе тестирования проблемном участке. Для уточнения используются рабочие станции, находящиеся по разные стороны маршрутизаторов проблемного участка. Важным условием третьего этапа является то, что рабочие станции исправны, т.е. функционируют без сбоев.

На основании рассмотренных выше решений можно сформулировать метод обнаружения отказов маршрутизаторов.

1. Формируем логическое дерево сети τ , выбираем граничные $\{R\}$ и корневые маршрутизаторы $\{C\}$, логические каналы (пути) $\{P\}$.
2. Проводим мониторинг, накапливаем данные.
3. В соответствии с заданными путями определяем соответствующие логические переменные $P_{i,j}$.
4. Формируем системы уравнений для каждого раунда по часовой стрелке (прямого) и против часовой стрелки (обратного).
5. Решаем полученные системы уравнений и выявляем сбойные сегменты $\{Q^-\}$.

6. Выявляем маршрутизаторы-претенденты в список отказавших $R_i \in Q^-$.

7. Для каждого выбранного маршрутизатора $R_i \in Q^-$ формируем логическое дерево τ_h , состоящее из пары рабочих станций и маршрутизатора.

8. Принимаем допущение, что рабочие станции функционируют нормально.

9. Проводим тестирование для логических деревьев, состоящих из рабочих станций и маршрутизатора.

10. Формируем логические уравнения, решаем их и определяем отказавшие сегменты $\{(a, b)\}$.

11. На основании полученных решений выявляем отказавшие маршрутизаторы $\{R_i\}$.

Область применения полученных результатов. Предлагаемый метод может применяться для оперативной диагностики состояния компьютерных сетей.

5. Примеры использования разработанных методов

Рассмотрим компьютерную сеть, которая состоит из 1-го корневого маршрутизатора и 3-х граничных маршрутизаторов (см. рис.4). Обозначим маршрутизаторы номерами 1,2,3,4, где 1,3,4 – граничные маршрутизаторы и 2 – корневой маршрутизатор. Обозначим пути между граничными маршрутизаторами через $P_{1,3}, P_{3,4}, P_{4,1}$ для первого раунда обхода сети. В этом случае состояние перегрузки для этих связей обозначим через X_{12}, X_{23}, X_{24} .

Пример 1. Пусть связь между 1-м и 2-м маршрутизаторами перегружена, т.е. переменная $X_{1,2}=1$. Полагаем, что это условие нам не известно. Два раунда метода (обход графа против часовой стрелки и по часовой стрелке) дадут следующие уравнения:

$$\begin{cases} X_{12} + X_{24} = P_{14} = 1, & \begin{cases} X_{12} + X_{23} = P_{13} = 1, \\ X_{42} + X_{23} = P_{43} = 0, \\ X_{32} + X_{21} = P_{31} = 0; \end{cases} \\ X_{42} + X_{23} = P_{43} = 0, & \begin{cases} X_{32} + X_{24} = P_{34} = 0, \\ X_{42} + X_{21} = P_{41} = 0. \end{cases} \end{cases}$$

Решая эти две системы уравнений, получаем

$$X_{32} = X_{24} = X_{42} = X_{21} = X_{23} = X_{32} = 0.$$

Следовательно,

$$\begin{cases} X_{12} + X_{23} = X_{12} + 0 = 1, \\ X_{12} + X_{24} = X_{12} + 0 = 1. \end{cases}$$

Отсюда следует, что $X_{12} = 1$, т.е. связь между маршрутизаторами 1 и 2 перегружена (большой процент потерь пакетов).

Пример 2. Пусть перегружено две связи: от маршрутизатора 1 к маршрутизатору 2 и от маршрутизатора 2 к маршрутизатору 3, т.е. переменные $X_{12} = X_{23} = 1$. Полагаем, что это условие нам не известно.

После двух раундов проверки получаем

$$\begin{cases} X_{12} + X_{23} = P_{13} = 1, & \begin{cases} X_{12} + X_{24} = P_{14} = 1, \\ X_{42} + X_{23} = P_{43} = 1, \\ X_{32} + X_{21} = P_{31} = 0. \end{cases} \\ X_{32} + X_{24} = P_{34} = 0, & \\ X_{42} + X_{21} = P_{41} = 0; & \end{cases}$$

Решая эти две системы уравнений, получаем, что $X_{32} = X_{24} = X_{42} = X_{21} = 0$.

Следовательно,

$$\begin{cases} X_{12} + X_{23} = 1 \\ X_{12} + X_{24} = X_{12} + 0 = 1 \\ X_{42} + X_{23} = 0 + X_{23} = 1. \end{cases}$$

Отсюда следует, что $X_{12} = X_{23} = 1$.

Расширение примера 1. Для выявленного отказавшего сегмента $X_{12} = 1$ строим новое логическое дерево. Пусть рис.4 соответствует такому дереву. На рисунке станции из набора $\{a,b,c\}$ – выбранные рабочие станции. Пусть неисправным является маршрутизатор R1. Полагаем, что это условие нам не известно.

Формируем уравнения:

$$\begin{cases} X_{a1} + X_{1b} = P_{ab} = 1, & \begin{cases} X_{b2} + X_{2c} = P_{bc} = 0, \\ X_{c2} + X_{2b} = P_{cb} = 0. \end{cases} \\ X_{b1} + X_{1a} = P_{ba} = 0; & \end{cases}$$

Решая эти две системы уравнений, получаем, что $X_{b2} = X_{2c} = X_{c2} = X_{2b} = X_{b1} = X_{1a} = 0$.

Следовательно, $X_{a1} + X_{1b} = 1$, но согласно (5)

$$\begin{cases} X_{a1} = q_{a1} + s_{a1}, \\ X_{1b} = q_{1b} + s_{1b}. \end{cases}$$

Знания о том, что рабочие станции функционируют нормально $q_{a1} = q_{1b} = 0$, в совокупности с полученным решением систем уравнений дают возможность определить, что в данном случае неисправность присутствует на участке, за который отвечает маршрутизатор R1. Имеем $(s_{a1} \text{ or } s_{1b}) = 1$, следовательно, маршрутизатор R1 является неисправным. Связь между маршрутизатором 1 и станцией (a) перегружена (большой процент потерь пакетов).

6. Выводы

Предложены методы, позволяющие выявить отказы маршрутизаторов компьютерной сети. Основными результатами исследования являются методы обнаружения отказавших маршрутизаторов в сегменте. Это развивает методы, представленные в [6].

Основные научные результаты можно представить в вербальном виде:

-усовершенствован метод выявления отказов сегментов и компонентов компьютерной сети, основанный

на формировании логических деревьев и их тестировании. При этом строятся дополнительные деревья, позволяющие выявить отказавшие узлы. Метод основан на использовании алгебры логики, позволяет получить эффективный (с малым потреблением ресурсов) алгоритм тестирования сети.

Практическая ценность полученных результатов состоит в том, что при использовании предложенного метода уменьшается нагрузка служебным трафиком компьютерной сети, экономится расход полосы пропускания и повышается эффективность использования ресурсов компьютерной сети в целом. Эти результаты позволяют повысить коэффициент утилизации каналов сети и в свою очередь снизить затраты на выполнение служебных функций, повысить качество обслуживания пользователей и доходность от всей сети.

Сравнение с лучшими аналогами. Предложенные методы в сравнении с методом, описанным в [6], позволяют выявлять не только отказавшие сегменты, но и отказавшие узлы. В отличие от методов [2,4,5] они являются экономными с точки зрения затрат на полосу пропускания.

Направление дальнейших исследований. Результаты, полученные в данной работе, касались лишь одного из аспектов улучшения работы протокола управления качеством передачи потока данных. В дальнейшем предполагается рассмотрение реализации полученного метода для сетей со слабоинтеллектуальными узлами и гибридными сегментами Router-Switch.

Литература: 1. Al-Shaer E., Abdel-Wahab H., and Maly K.. HiFi: a new monitoring architecture for distributed systems management. In *Proc. IEEE 19th International Conference on Distributed Computing Systems (ICDCS'99)*. Austin, TX, May 1999. P. 171–178. 2. Anagnostakis K. G., Ioannidis S., Miltchev S., Ioannidis J., Greenwald M., and Smith J. M. Efficient packet monitoring for network management. In *Proc. IEEE Network Operations and Management*

Symposium (NOMS). Florence, Italy, Apr. 2002. P. 50-52. 3. Breitbart Y., Chan C. Y., Garofalakis M., Rastogi R., and Silberschatz A. Efficiently monitoring bandwidth and latency in IP networks. In *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001. P. 72-76. 4. Dilman M. and Raz D. Efficient reactive monitoring. In *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001. P. 34-37. 5. Duffield N.G., Horowitz J., Presti F. Lo, and Towsley D. Network delay tomography from end-to-end unicast measurements. In *Proc. of the 2001 International Workshop on Digital Communications 2001 Evolutionary Trends of the Internet, Italy*. Sept. 2001. P. 121-132. 6. Habib A., Khan M., and Bhargava Bharat. Edge-to-Edge measurement-based distributed network monitoring. *Computer Networks Journal*, 2004. Vol. 44, Issue 2. P. 211-233. 7. Liotta A., Pavlou G., and Knight G. Exploiting agent mobility for large-scale network monitoring. *IEEE Network*. May/June, 2002. P. 112-117. 8. Stone R. Centertrack: An IP overlay network for tracking DoS floods. In *Proc. USENIX Security Symposium*. Denver, CO, Aug. 2000. P. 154-165. 9. Subramanyan R., Miguel-Alonso J., and Fortes J. A. B. A scalable SNMP-based distributed monitoring system for heterogeneous network computing. In *Proc. High Performance Networking and Computing Conference (SC 2000)*, Dallas, TX, 2000. P. 82-90. 10. Hoonsoon Ku; Forslow, J.; Park, J.-G. Web-based configuration management architecture for router networks. *Network Operations and Management Symposium*, 2000. NOMS 2000. 2000 IEEE/IFIP, 2000. P. 171-177.

Поступила в редколлегию 13.06.2007

Рецензент: д-р техн. наук, проф. Поповский В.В.

Саенко Владимир Иванович, канд. техн. наук, доц., проф. каф. информационных управляющих систем ХНУРЭ. Научные интересы: менеджмент компьютерных сетей. Увлечения и хобби: садоводство. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.

Алексеев Дмитрий Игоревич, ассистент кафедры информационных управляющих систем ХНУРЭ. Научные интересы: методы и технологии обнаружения и исправления ошибок в компьютерных сетях. Увлечения и хобби: компьютерные сети. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.

УДК681.325.65:535

СУММАТОРЫ И ПАРАЛЛЕЛЬНЫЕ ЛОГИЧЕСКИЕ ВЕНТИЛИ С ПРОСТРАНСТВЕННО- ПЕРЕСТРАИВАЕМЫМИ ОПТИЧЕСКИМИ СВЯЗЯМИ

ДЕМЬХИН В.В.

Описывается новый простой метод топологической обработки информации (ТОИ) и анализируются особенности построения оптических систем сумматоров и параллельных цифровых логических элементов, использующих при вводе операндов направленное расщепление и дозированное изменение потока входных сигналов («эффект зрачка»), а также определение результата обработки информации по количеству прошедших к выходу этих устройств сигналов.

Проводится сравнение ТОИ-метода с микроэлектронными и оптоэлектронными методами выполнения арифметических и логических операций.

1. Введение. Выполнение арифметических и логических операций с помощью оптических методов, использующих пространственное положение входных переменных и модуляцию передаточной функции пространственного фильтра [1-3], имеет ряд достоинств, недостижимых с помощью средств электронной техники: высокое быстродействие, возможность параллельного выполнения операций над большими массивами данных, отсутствие взаимных помех при передаче данных по оптическим каналам и другие. В то же время методы [1-3] имеют недостатки: большой динамический диапазон уровней входного и выходного сигналов, необходимость специального кодирования входных объектов, дифракционное размывание входных структур и низкая энергетическая добротность.