

**Калачев А.А., Краснов А.Е., Надеждин Е.Н., Никольский Д.Н., Репин Д.С.**

Федеральное государственное автономное учреждение «Государственный научно-исследовательский институт информационных технологий и телекоммуникаций», г. Москва, Россия

## **МОДЕЛЬ ГЕТЕРОГЕННОЙ СЕТИ ДЛЯ СИМУЛЯЦИИ НЕЙРОДИНАМИЧЕСКИХ ЗАДАЧ**

### **АННОТАЦИЯ**

*Для симуляции нейродинамических задач предложен подход, основанный на функционально-структурной модели гетерогенной сети, состоящей из информационной и управляющей подсетей, и её объектно-ориентированной программной реализации. Для каждой из подсетей описаны активные структурные элементы, а также способ их объединения в общую гетерогенную многосвязную сеть. Определены основные функции активных структурных элементов такой сети. Представлено краткое описание архитектуры программного комплекса, предназначенного для работы на кластерах универсальных и/или графических процессоров. Выполнена апробация разработанной модели на основе решения ряда известных нейродинамических задач высокой размерности.*

### **КЛЮЧЕВЫЕ СЛОВА**

*Функционально-структурная модель; гетерогенная сеть; граф; активный элемент; управляемая связь; объектно-ориентированное программирование; кластер; универсальный процессор; графический процессор; программный комплекс; нейродинамика.*

**Kalachev AA, Krasnov A.E., Nadezhdin E.N., Nikolskii D.N., Repin D.S.**

Federal state autonomous institution "State Research Institute of Information Technologies and Telecommunications", Moscow, Russia

## **MODEL OF THE HETEROGENEOUS NETWORK FOR THE SIMULATION OF NEURO DYNAMIC PROBLEMS**

### **ABSTRACT**

*The approach based on the functional-structural model of a heterogeneous network, consisting of information and control subnets and its object-oriented software implementation for the simulation of neuro dynamic problems is proposed. For each of subnets active structural elements, as well as a way to join them in a common multiply heterogeneous network are described. The main structural features of active elements of the network are determined. A brief description of the software architecture of the complex, designed to run on clusters of universal and/or graphics processors is presented. The approbation of the developed model on the basis of the solution of a number of known neurodynamic problems of high dimension is executed.*

### **KEYWORDS**

*Functional and structural model; heterogeneous network; count; active element; operated communication; object-oriented programming; cluster; universal processor; graphic processor; program complex; neurodynamics.*

### **Введение**

Для исследования ряда проблемных задач естественных наук и программной реализации специализированных моделей, например, нейронных сетей, часто используют промышленные нейропакеты (Statistica Neural Networks, NeuroShell, нейромодуль Circuit simulator в Matlab, GENESIS, NetLab, CONNECT [1]). По мнению экспертов, перспективным инструментарием при создании универсальных моделей сложных сетей следует считать объектно-ориентированное программирование [2, 3], а также использование кластеров универсальных и графических процессоров [4-6].

В последние годы повышенное внимание уделяется вопросам управляемости сложными сетями в классической постановке [7], при которой управляющие сигналы через, так называемые, ведущие узлы, воздействуют на другие узлы, определяющие функциональное состояние сети [8-10]. Так, например, в работе [11] при исследовании влияния атак на управляемость вычислительной сети обнаружено, что целенаправленная атака на связи (ребра) узлов является более эффективной, чем случайная атака на сами узлы. В дополнение к этому в работах [12, 13] предложен новый подход к моделированию, в котором рассматривается и динамика связей, с которыми соотносится вектор состояния сети.

Следует отметить, что вопросам организации управления динамикой сети через настройку весов связей посвящено большое количество теоретических работ. При этом для естественнонаучных приложений наиболее часто привлекается аппарат дифференциальных (рекуррентных) сетей [14].

Представляется целесообразным предложить новый подход к построению формальной модели сети, в архитектуре которой объединены информационные компоненты и элементы управления связями, в совокупности поддерживающие её функциональность и гибкость при реализации на кластерах универсальных и/или графических процессоров. Предпосылками для создания такой универсальной сетевой модели, ориентированной на решение достаточно широкого круга задач сетевой динамики, служат известные результаты ранее опубликованных работ [4-6].

В связи с этим, в настоящей работе решается задача моделирования на функционально-структурном уровне гетерогенной многосвязной сети, состоящей из информационной и управляющей подсетей, образованных большим числом активных элементов, а также – построения объектно-ориентированного программного комплекса, реализующего данную модель на кластерах универсальных и/или графических процессоров.

Конечной целью работы является создание гибкого инструмента симуляции задач сетевой динамики на кластерах универсальных и/или графических процессоров.

Развиваемый в статье подход сложился в результате обобщения многолетнего опыта собственных прикладных исследований, а также под влиянием идей, получивших обоснование в нейробиологических работах зарубежных авторов [15-17].

Однако, по сравнению с существующими подходами, где описана лишь идея обмена потоками кальция и нейромедиаторов между синапсами и астроцитами, в работе предлагается обобщение в виде взаимодействия двух подсетей, одна из которых образована коммутаторами и связывающими их коннекторами, а другая – контроллерами. В модели каждый контроллер опрашивает несколько коннекторов, получая информацию об их состояниях, после чего настраивает их параметры.

Введение подсети контроллеров позволит экспериментально изучать вопросы управления связями сетей, формирования различных сетевых кластеров, их влияние на процессы обучения сетей, а также – повысить надежность их функционирования. Использование нескольких контроллеров позволяет также распараллелить процесс управления для сетей с большим количеством связей ( $\geq 10^8$ ).

Основу методологии настоящего исследования составили базовые положения и принципы системного анализа, методы теории графов, а также методы функционально-структурного моделирования и объектно-ориентированного программирования.

## **Результаты**

### **Структурные элементы сети и ее топология**

Информационная подсеть включает: коммутаторы («*Comm*»,  $N$ ) и коннекторы («*Conn*»,  $N \times N$ ).

Коммутатор «*Comm*» имеет входные и выходные информационные порты, соединенные с внешними и внутренними входами и выходами концентраторами, а также – порты внешнего управления.

К входному порту коммутатора через концентраторы подключено множество входов, а к выходному порту – множество выходов.

Коннектор «*Conn*» имеет входные и выходные информационные порты, а также – порт внешнего управления и порт внутреннего управления.

Управляющая подсеть образована контроллерами («*Contr*»,  $M \geq N$ ).

В управляющей подсети контроллеры «*Contr*» имеют входные и выходные информационные порты, а также – порт внешнего управления и порт внутреннего управления. Данные порты соединяются коммуникационными каналами через концентраторы.

Фрагмент структуры предлагаемой сети показан на рисунке 1.

Настройка всех активных структурных элементов гетерогенной сети осуществляется через магистраль внешнего управления.

В информационной подсети происходит обмен информационными потоками  $I_{comm}$  и  $I_{conn}$  данных, преобразуемых коммутаторами и коннекторами по соответствующим внутренним магистралям (связям) коммутаторов и коннекторов.

На вход каждого коммутатора «Comm» через внутреннюю магистраль коммутаторов подключены выходы группы коннекторов «Conn».

Информационный выход каждого коммутатора Comm соединен через внутреннюю магистраль коммутаторов с группой коннекторов. Так, например, выход коммутатора Comm1 – информационный поток  $I_{comm1}$  данных воздействует на входы коннекторов Conn1,1, ..., Conn1,N.

В управляющей подсети происходит обмен потоками  $S_{conn}$  и  $C$  данных через магистраль внутреннего управления (Trunk of the internal control). Поток  $S_{conn}$  данных описывает состояние коннекторов, а  $C$  – управление коннекторами со стороны контроллеров.

Межконтроллерный обмен информационными потоками  $I_{contr}$  данных осуществляется через внутреннюю магистраль (Trunk of the communication exchange).

Так как число  $M$  контроллеров намного меньше числа  $N \times N$  коннекторов, то каждый контроллер управляет группой (множеством) коннекторов. Например, коннекторами группы  $\{1,1\}, \dots, \{1,N\}$  (Conn1,1, ..., Conn1,N) управляет контроллер Contr1, который поочередно опрашивает состояния  $\{S1,1, \dots, S1,N\}$  соответствующих коннекторов и вырабатывает сигналы управления  $\{C1,1, \dots, C1,N\}$  управления. Несколько коннекторов получают сигнал управления от одного контроллера через магистраль внутреннего управления.

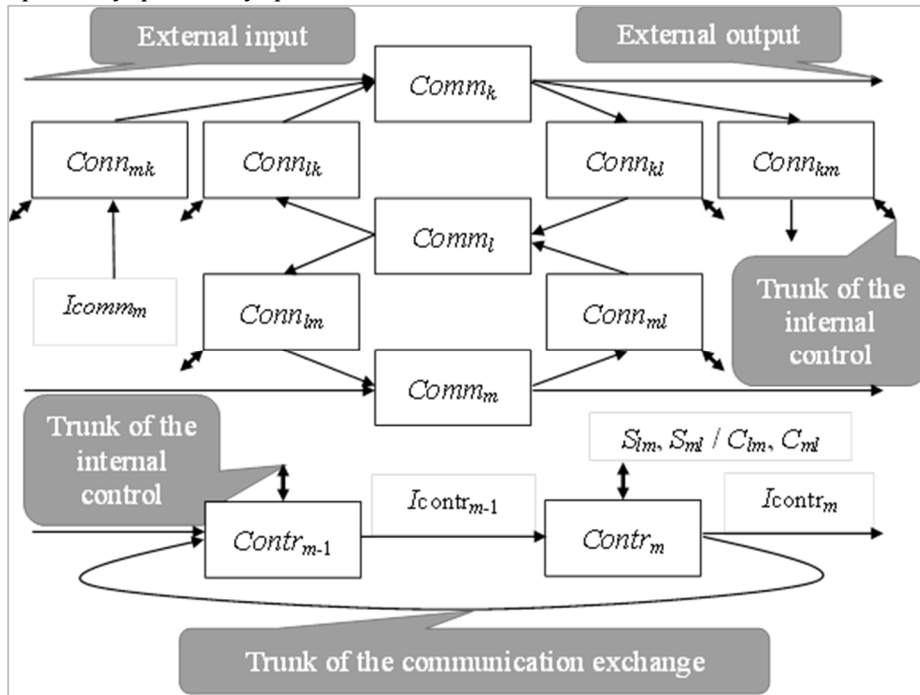


Рисунок 1 – Структура гетерогенной многосвязной сети

### **Функциональные модели компонентов сети**

Активные структурные элементы рассмотренной выше сети обладают следующими функциями.

Коммутатор («Comm») осуществляет функции пространственной и временной интеграции входных информационных сигналов, а также формирования выходного информационного сигнала.

Для описания передаточной функции элемента могут быть использованы как статические логические выражения, так и динамические уравнения, рассмотренные, например, в [10, 14]. В последнем случае в  $n$ -ом коммутаторе  $Comm_n$  осуществляется интегрирование системы динамических уравнений, например, вида:

$$\frac{dI_{comm}(n)}{dt} = F\{I_{comm}(n), I_{conn}(1), \dots, I_{conn}(K), I_{in}, Z_{comm}, Par_1\}, \quad (1)$$

$$\frac{dZ_{comm}(n)}{dt} = \mathbf{G}\{I_{comm}(n), I_{conn}(1), \dots, I_{conn}(K), I_{in}, \mathbf{Z}_{comm}(n), \mathbf{Par}_z\},$$

где  $I_{comm}(n)$  – выходной информационный сигнал  $n$ -го коммутатора ( $n = 1, \dots, N$ );  $I_{conn}(1), \dots, I_{conn}(K)$  – входные информационные сигналы от  $K$  коннекторов;  $I_{in}$  – внешний входной информационный сигнал;  $\mathbf{Z}_{comm}(n)$  – внутренняя векторная переменная. В (1) конкретный вид скалярной функции  $F$ , векторной функции  $\mathbf{G}$ , векторов параметров  $\mathbf{Par}_1$  и  $\mathbf{Par}_z$  задаются через магистраль внешнего управления.

Для задач нейродинамики условие формирования выходного значения сигнала  $I_{comm}$  может быть задано, например, в виде: if  $I_{comm} \geq I_{thr}, I_{out}$ ; otherwise  $I_{comm} = 0$ , где  $I_{thr}$  – пороговое значение сигнала.

Коннектор («Conn») осуществляет функции временной интеграции входного информационного сигнала и формирования выходного сигнала, формирование функции состояния, а также функции запоминания параметров настройки со стороны контроллера.

Для описания передаточных функций элемента могут быть использованы как статические логические выражения, так и динамические уравнения, рассмотренные, например, в работах [15, 16]. В последнем случае в  $k,l$ -ом коннекторе  $Conn_{k,l}$  ( $k, l = 1, 2, \dots, N \times N$ ), соединяющем  $k$ -й и  $l$ -й коммутаторы (см. рисунок 1), осуществляется интегрирование динамических уравнений, описывающих скорость изменения выходного информационного сигнала  $I_{conn}(k,l)$  и скорость изменения функции  $S_{conn}(k,l)$  его состояния в зависимости от входного информационный сигнала  $I_{comm}(k)$  от  $k$ -го коммутатора и управляющего сигнала  $C_{contr_m}$  от  $m$ -го контроллера ( $m = 1, \dots, M; N \ll M \ll N \times N$ ).

Контроллер («Contr») реализует вычислительные и управляющие функции для группы коннекторов. Для описания его работы могут быть использованы как статические логические выражения, так и динамические уравнения, рассмотренные, например, в статьях [16, 17]. В последнем случае в  $m$ -м контроллере  $Contr_m$  осуществляется интегрирование динамических уравнений.

Входным управляющим сигналом для контроллера является функция  $S_{conn}(k,l)$  состояния опрашиваемого  $k,l$ -го коннектора, а выходным – управляющий сигнал  $C_{contr}$ .

Каждый  $m$ -ый контроллер вырабатывает выходной информационный сигнал  $I_{contr}(m)$  в зависимости от информационных сигналов  $I_{contr}(m-k), I_{contr}(m+k)$ , связанных с ним через магистраль Trunk of the communication exchange. Связь по информационным каналам необходима для резервирования управляющих функций в случае отказа каких-либо контроллеров. Использование нескольких контроллеров позволяет осуществить распараллеливание процесса управления в сетях с большим количеством коммутаторов.

Сигналы управление  $C_{contr}(m)$  формируются в зависимости от функций  $S_{conn}(1), \dots, S_{conn}(K)$  состояния  $K$  коннекторов, связанных с ним через магистраль Trunk of the internal control внутреннего управления.

Рассмотренная функционально-структурная модель гетерогенной сети активных элементов позволяет осуществить их физическую реализацию на базе интегральных микросхем. В частности, для сети с большим количеством активных элементов ( $N \geq 10^5$ ), представляет интерес универсальная технология на основе графических процессоров (GPU) и кластера универсальных процессоров (CPU).

Ниже рассмотрен объектно-ориентированный подход к программной реализации предложенной модели на кластере универсальных процессоров и видеокарт.

### **Объектно-ориентированный подход к программной реализации модели**

Библиотека для моделирования элементов гетерогенной многосвязной сети активных элементов содержит следующие группы классов:

Communicator – содержит классы, моделирующие коммутаторы «Comm»: Comms, CommsCPU, CommsGPU, ... .

Connector – содержит классы, моделирующие коннекторы (соединения) «Conn»: Conns, ConnsCPU, ConnsGPU, ... .

Controller – содержит классы, моделирующие контроллеры «Contr»: Contrs, ContrCPU, ContrsGPU, ... .

Каждый класс использует программно-аппаратные решения для реализации параллельной обработки данных.

Иерархии классов построены с применением принципов объектно-ориентированного программирования. Группы Communicator, Connector и Controller инкапсулируют базовые интерфейсы для своих классов.

Для создания классов, содержащих конкретные реализации методов, связанных с реализацией конкретных математических моделей для конкретной архитектуры, используется схема наследования, представленная на рисунке 2.

На рисунке 2 представлено создание конкретных наследников класса Comms для выполнения на кластерах CPU и GPU.

Класс Comms предоставляет общий интерфейс для всех классов, моделирующих работу коммутатора. Он является предком для классов CommCPU и CommGPU, содержащих специализированные структуры для хранения данных для универсальных и графических процессоров.

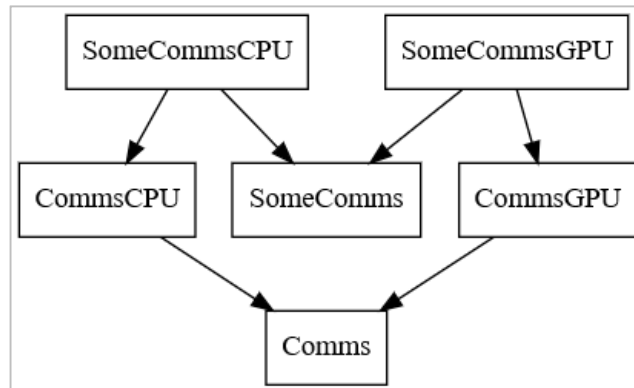
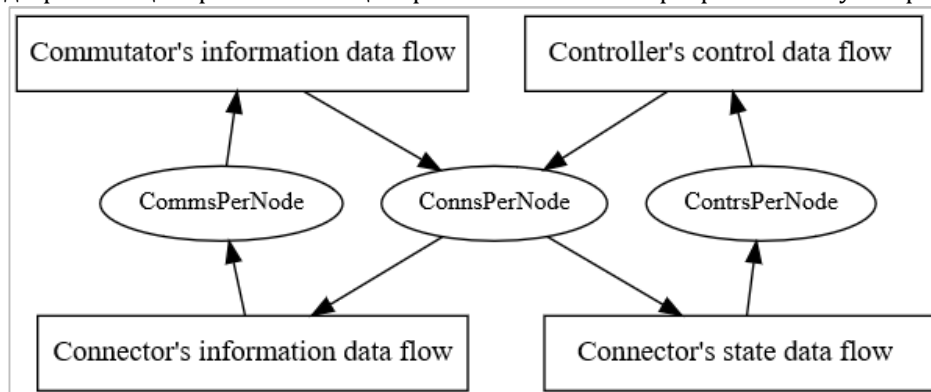


Рисунок 2 – Схема создания конкретных наследников для выполнения на кластерах CPU и GPU

Для разработки конкретных классов, реализующих конкретные математические модели, создается базовый класс SomeComms, содержащий поля и методы, необходимые для реализации необходимой математической модели. Затем, разрабатываются классы SomeCommsCPU и SomeCommsGPU, выполняющие расчеты на универсальных и графических процессорах.

Принцип полиморфизма, принятый в объектно-ориентированном программировании, позволяет разрабатывать алгоритмы, работая с указателями на базовые классы. Так, сеть, представленная на рисунке 3, программируется с использованием указателей на базовые классы, которым в ходе реализации различных сценариев выполнения программы могут передаваться



любые наследники.

Рисунок 3 – Схема обработки потоков данных узлами программных модулей

Рисунок отражает циклический обмен информационными ( $I_{comm}$ ,  $I_{conn}$ ) и управляющими ( $S_{conn}$ ,  $S_{contr}$ ) потоками данных.

Каждый из классов CommsPerNode, ConnsPerNode и ContrsPerNode занимается обработкой порции объектов, приходящих на один узел кластера. Последнее позволяет выполнять симуляцию работы больших сетей на кластере универсальных и (или) графических процессоров.

Далее представлены результаты компьютерного эксперимента, демонстрирующие, что на одном узле сети одновременно можно обрабатывать порядка  $10^6$  объектов. Это позволяет в  $10^2$  узлах кластера одновременно обрабатывать до  $10^8$  объектов. Таким образом, следуя изложенным рекомендациям, можно реализовать процесс моделирования одновременной работы  $10^4$  нейронов,

объединенных в сеть  $10^8$  числом связей. Заметим, что указанная мощность множества нейронов соответствует количеству нейронов в одной кортикальной колонке головного мозга [17, 19].

### Примеры решения нейродинамических задач

В качестве примера ниже рассмотрено решение задач моделирования нейродинамического процесса с использованием кластера на основе центральных процессоров Intel Core i5-2300 и графических процессоров GM206 (NVidia Geforce GTX 950).

Рассмотрим результаты вычислительных экспериментов с сетью коммутаторов.

Для проверки разработанного объектно-ориентированного подхода в качестве коммутатора «Conn» был выбран нейрон с динамическим уравнением (1) в виде модели Е.М. Ижикевича [19], в качестве коннектора «Conn» – синапс с безынерционным откликом и регулируемой амплитудой и временем  $\tau_{conn}$  спада его отклика, в качестве контроллера «Contr» – астроцит с функцией регулятора амплитуды времени  $\tau_{conn}$  спада.

В ходе проведения расчетов на кластере, были выполнены замеры времени выполнения одного шага численного интегрирования уравнения (1) на CPU и GPU для различного числа нейронов, приходящихся на один узел кластера  $N_{\text{NeursPerNode}} = N_{\text{neurs}} / \text{Nodes\_CPU}$ . Здесь  $N_{\text{neurs}}$  – число нейронов в моделируемой сети,  $\text{Nodes\_CPU}$  – число узлов кластера.

Исследовалась зависимость времени расчета одного шага решения уравнения (1) от числа  $N_{\text{neurs}}$  моделируемых нейронов на одном CPU и GPU. Данные зависимости приведены на рисунке 4.

Для построения графика зависимости использовалось среднее время работы универсальных или графических процессоров кластера при выполнении 1000 шагов интегрирования.

В ходе эксперимента исследован также процесс генерации возбуждений (спайков) 100 нейронов при их связи в виде полного графа и разных настройках времен  $\tau_{conn}$  релаксации откликов синапсов.

На рисунке 5 показана диаграмма активности нейронов при одинаковых для всех  $10^4$  синапсов настройках времен спада  $\tau_{conn} = 4$  мс.

Из диаграммы видно, что все нейроны генерируют спайки (импульсы возбуждения) практически синхронно.

На рисунке 6 приведена диаграмма активности нейронов, разделенных на группы (34, 33, 33), с интервалами времени релаксации синаптических откликов 0,4 мс, 4 мс и 40 мс соответственно.

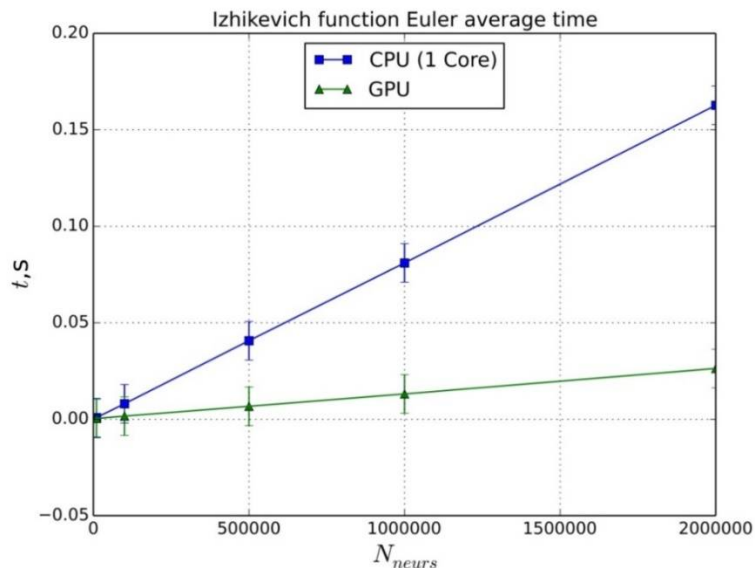


Рисунок 4 – Зависимость времени расчета одного шага решения уравнения (1) на CPU (1 процессорное ядро) и GPU от числа  $N_{\text{neurs}}$  моделируемых нейронов

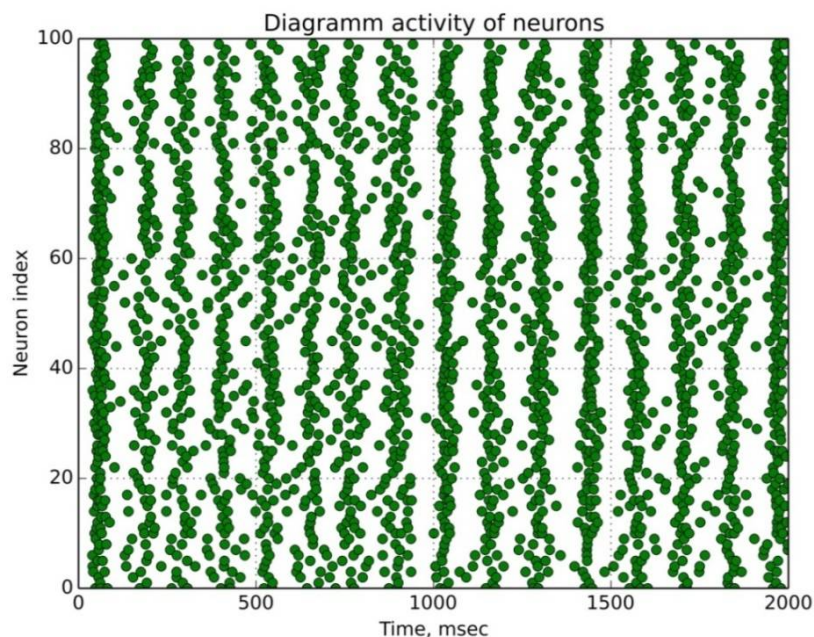


Рисунок 5 – Диаграмма активности нейронов при одинаковых для всех  $10^4$  синапсов настройках времен  $\tau_{conn} = 4$  мс релаксации их откликов (кружками отмечены спайки нейронов в соответствующие моменты времени)

Из диаграммы видно, что нейроны с большими временами релаксации синаптических откликов генерируют спайки чаще.

При дополнительном тестировании использовалась полносвязная сеть из 1000 нейронов, построенная на основе математической модели, реализованной в пакете Matlab численным методом Эйлера [20]. Результаты работы этой модели были сравнены с результатами, полученными нами на GPU и CPU (рисунок 7).

В имитационном эксперименте также как и в [21], наблюдались альфа-ритм (10 Гц) и гамма-ритм (40 Гц). Тем самым, подтверждена правильность функционирования разработанной модели и ее реализации, как на графическом, так и универсальном процессорах.

Представим результаты эксперимента по взаимодействию подсетей коммутаторов и контроллеров.

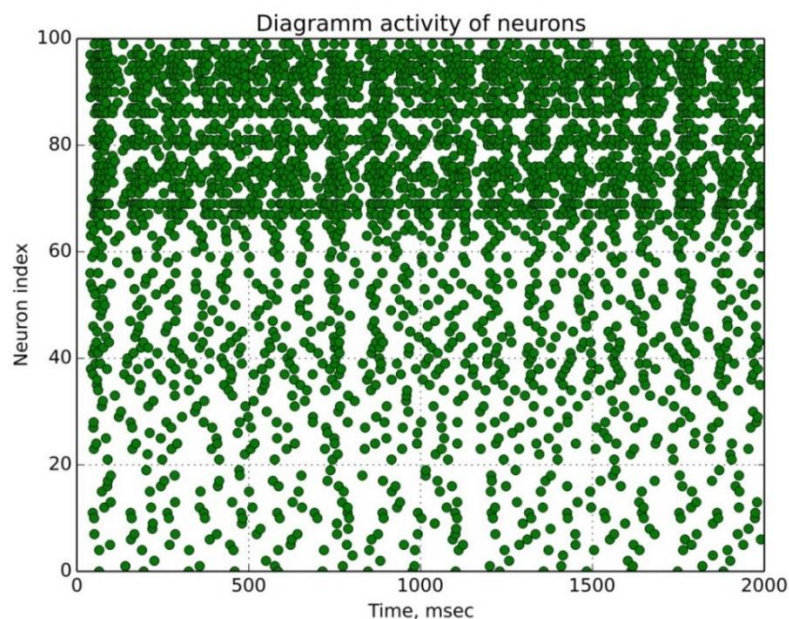


Рисунок 6 – Диаграмма активности нейронов неоднородной сети из трех групп различных нейронов (0,4 мс, 4 мс и 40 мс)



В процессе апробации разработанной модели на качественном уровне был проведён анализ взаимодействия коннекторов сети Conns с контроллерами Contr. В эксперименте в качестве модели (1) коммутаторов Comms была взята устойчивая к помехам модель нейрона «Накопление-Сброс» («Integrate and Fire») [18], а в качестве моделей регуляторного взаимодействия синапса и астроцита – известные модели обмена нейро- и глиапередатчиками [15–17].

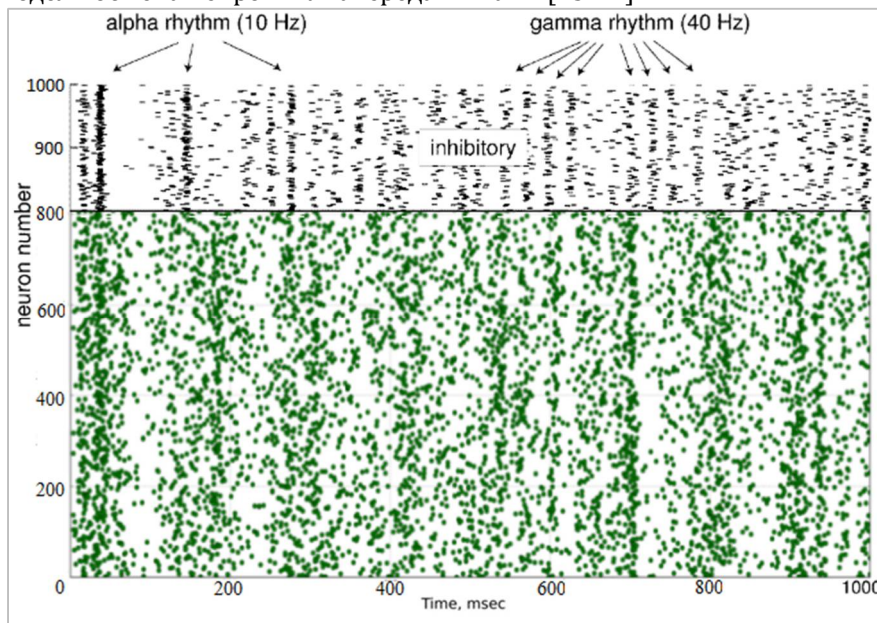


Рисунок 7 – Диаграмма активности полностью связанной нейронной сети (1000 нейронов), реализованной на GPU

В эксперименте нейроны-коммутаторы соединялись друг с другом через коннекторы, к которым были подключены контроллеры.

На внешние информационные входы коммутаторов подавалась случайная последовательность возбуждающих импульсов, а к их выходам была добавлена аддитивная равномерно распределенная помеха, которая через коннекторы передавалась на внутренние информационные входы коммутаторов.

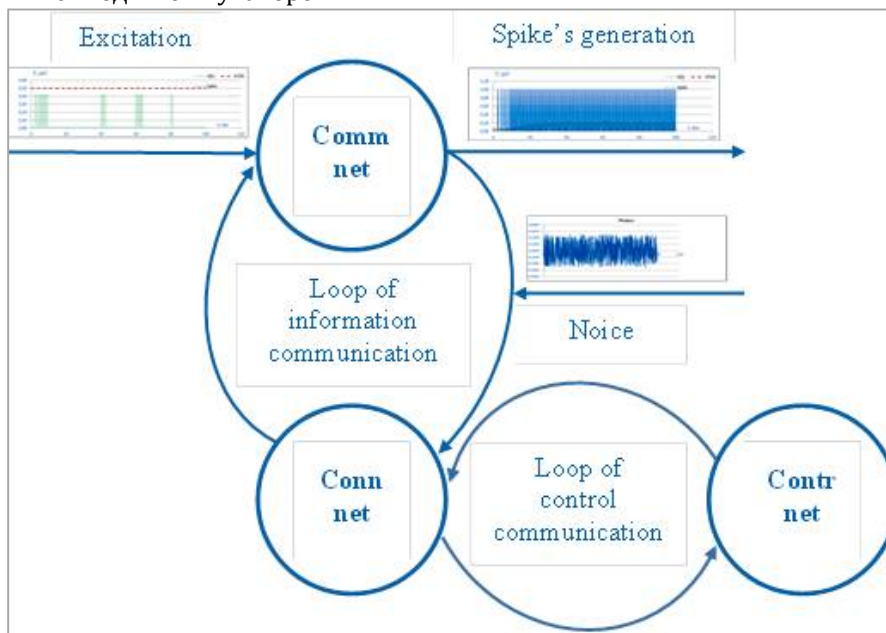


Рисунок 8 – Схема эксперимента по влиянию сети контроллеров на работу сети коммутаторов

В отсутствие петли управляющей связи и при превышении возбуждающими импульсами порогового значения коммутаторы генерировали устойчивые спайки на всем временном интервале возбуждающих импульсов (0 ...100 мс), независимо от изменения амплитуды помехи в широком диапазоне значений, как показано на рисунке 8.



На рисунке 9 приведена диаграмма спайков коммутаторов.

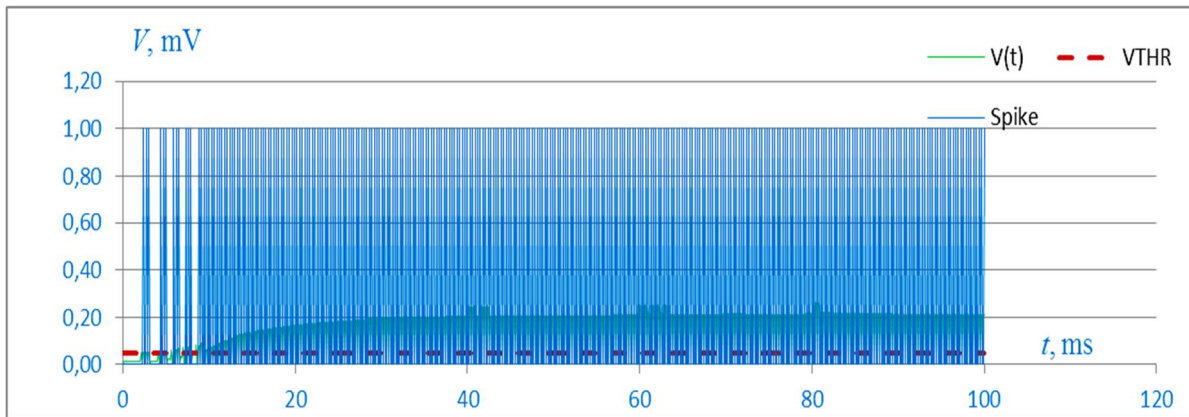


Рисунок 9 – Диаграмма спайков, генерируемых коммутаторами (модель нейрона «Накопление-отжиг») в отсутствии петли управляющей связи

Установлено, что при включении петли управляющей связи, функционирующей по регуляторным моделям обмена нейро- и глиапередатчиками [15-17], для некоторых реализаций помехи происходят срывы генерации. На рисунках 10 и 11 приведены примеры воздействия различных реализаций помехи ( $0,5 \text{ мкА/см}^2$ ) при различных уровнях связи коннекторов и контроллеров.

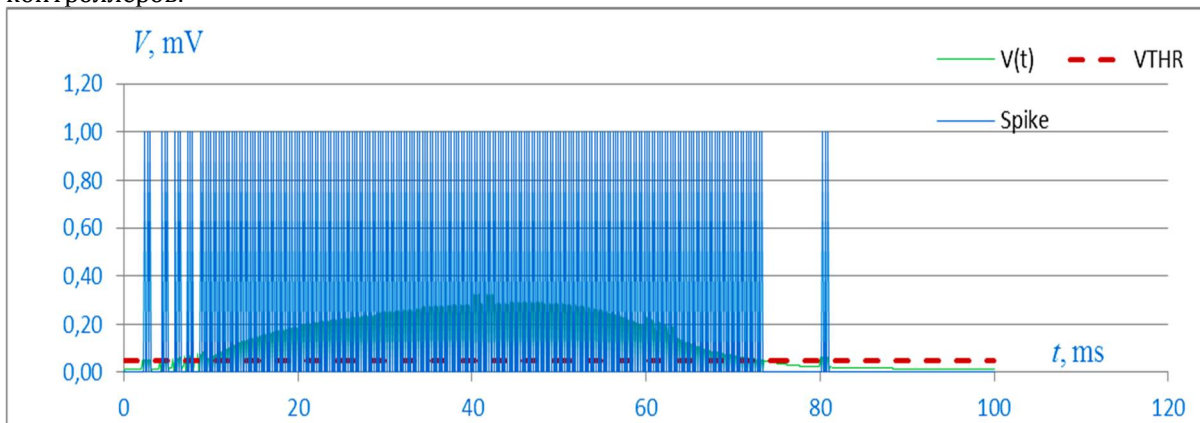


Рисунок 10 – Диаграмма спайков, генерируемых коммутаторами при заданной «жесткости» петли управляющей связи, функционирующей по моделям [15-17]

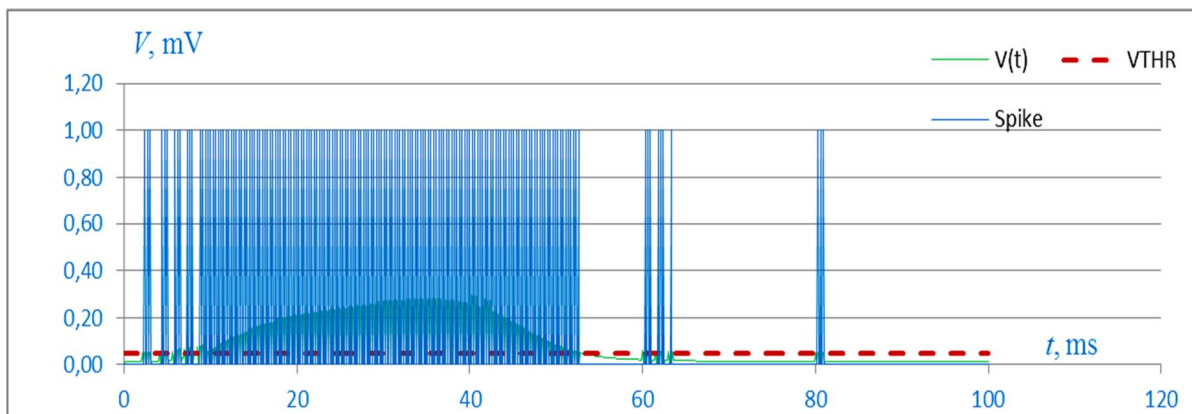


Рисунок 11 – Диаграмма спайков, генерируемых коммутаторами при повышении «жесткости» петли управляющей связи, функционирующей по моделям [15-17]

### Заключение

Проведенные имитационные эксперименты показали, что разработанная функционально-структурная модель гетерогенной многосвязной сети активных элементов и её объектно-ориентированная программная реализация составляют основу достаточно гибкого и

универсального инструмента для симуляции задач нейродинамики. Достоверность результатов исследования подтверждается их совпадением с результатами, полученными другими учёными в процессе решения аналогичных задач. Например, приведенные в статье авторские результаты симуляции нейробиологической сети полностью совпали с результатами известных работ Е.М. Ижикевича. Аналогичным образом в результате серии модельных экспериментов была подтверждена принципиальная возможность обеспечения управляемости сети со стороны контроллеров.

По данным вычислительного эксперимента, переход к использованию графических процессоров и технологии параллельного программирования CUDA (от компании NVIDIA), позволит обеспечить более высокую производительность на больших объёмах данных. Последнее обусловлено высокой эффективностью графических процессоров (по сравнению с универсальными процессорами) в условиях осуществления принципа компьютерных вычислений SIMD (Single Instruction Multiple Data). Детальное изучение влияния этих механизмов на производительность рассматриваемой гетерогенной сетевой модели и её многовариантное тестирование на современной линейке графических процессоров (NVIDIA Pascal) планируется в продолжение настоящей работы.

В дальнейшем разработанная сетевая модель и её усовершенствованная программная реализация могут быть использованы для экспериментального анализа и оценки устойчивости динамического процесса управления сетью в области критических значений параметров связей, а также для конструирования гетерогенных сетей на базе оптоэлектронных технологий [22].

### Литература

1. Kock G., Becher T. Simulation kunstlicher neuronaler Netze auf der Basis abstrakter Netzwerkbeschreibungen // GMD-Spiegel. 1997. V. 1. P. 29–31.
2. Nicol J. R., Wilkes C. T., Manola F. A. Object orientation in heterogeneous distributed computing systems // Computer. 2002. V. 26, I. 6. P. 29–31.
3. Binkert N.L., Hallnor E.G., Reinhardt S.K. Network-Oriented Full-System Simulation using M5 // Sixth Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW), New York. ACM SIGARCH Computer Architecture News. 2003.
4. Hall J.D., Hart J.C. GPU acceleration of iterative clustering // Proc. SIGGRAPH. 2004. – 13p.
5. Ohmer J., Maire F., Brown R. Implementation of kernel methods on the GPU // Proc. DICTA. 2005, Cairns, Australia, 2005. – 8 p.
6. Bahrampour S., et all. Comparative study of Caffe, Neon, Theano, and torch for deep learning // Workshop track – ICLR. 2016. P. 1–11.
7. Kalman R. E. Mathematical description of linear dynamical systems // J. Soc. Indus. Appl. Math. 1963. Ser. A 1. P. 152–192.
8. Gao X.-D. et al. Control efficacy of complex networks // Sci. Rep. 2016. V 6, Article number: 28037.
9. Yang-Yu L., Jean-Jacques S., Barabasi A.-L. Controllability of complex networks // Nature. 2011. Vol 473. P. 167–173.
10. Gao J., Barzel B., Barabasi A.-L. Universal resilience patterns in complex networks // Nature. 2016. Vol 530. P. 307–312.
11. Pu C.-L., Pei W.-J., Michaelson A. Robustness analysis of network controllability // Physica. 2012. A 391. P. 4420–4425.
12. Nepusz T., Vicsek T. Controlling edge dynamics in complex networks // Nature Physics. 2012. V 8. P. 568–573.
13. Slotine J.-J., Liu Y.-Y. The missing link // Nature Physics. 2012. V 9. P. 1–2.
14. Poznyak A.S., Sanchez E.N., Yu W. Differential Neural Networks for Robust Nonlinear Control. Identification, State Estimation and Trajectory Tracking. – Singapore. World Scientific publishing. 2001. – 422 p.
15. Ghanim U., et all. Anti-phase calcium oscillations in astrocytes via inositol (1, 4, 5)-trisphosphate regeneration // Cell Calcium. 2006. № 39. P. 197–208.
16. Nazari S., et all. A digital implementation of neuron–astrocyte interaction for neuromorphic applications // Neural Networks. 2015. V. 66, P. 79–90.
17. Agulhon C., et all. What Is the Role of Astrocyte Calcium in Neurophysiology? // Neuron. 2008. V. 59. I. 6. P. 932–946.
18. Izhikevich E.M. Hybrid spiking models. // Philosophical transactions. 2010. Series A, Mathematical, physical, and engineering sciences. V. 368, № 1930. P. 5061–5070.
19. Izhikevich E.M. Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. Cambridge, Massachusetts London, England. The MIT Press. 2007. – 497 p.
20. Izhikevich E.M. Simple model of spiking neurons // IEEE Transactions on neural networks. 2003. V. 14, № 6. P. 1569–1572.
21. Pu C.-L., Pei W.-J., Michaelson A. Robustness analysis of network controllability // Physica. 2012. A 391. P. 4420–4425.
22. Kompanets I.N., Neevina T.A., Kompanets S.I. Method of parallel switching of optical channels // QUANTUM ELECTRON. 2012, V. 42 (12). С. 1093–1096.

### References

1. Kock G. Becher T. Simulation kunstlicher neuronaler Netze auf der Basis abstrakter Netzwerkbeschreibungen // GMD-Spiegel. 1997. V. 1. P. 29–31.
2. Nicol J. R. Wilkes C. T., Manola F. A. Object orientation in heterogeneous distributed computing systems // Computer. 2002. V. 26. I. 6. P. 29–31.
3. Binkert N.L. Hallnor E.G., Reinhardt S.K. Network-Oriented Full-System Simulation using M5 // Sixth Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW). New York. ACM SIGARCH Computer Architecture News. 2003.
4. Hall J.D., Hart J.C. GPU acceleration of iterative clustering // Proc. SIGGRAPH. 2004. – 13p.
5. Ohmer J., Maire F., Brown R. Implementation of kernel methods on the GPU // Proc. DICTA. 2005. Cairns. Australia. 2005. – 8 p.

6. Bahrampour S., et al. Comparative study of Caffe. Neon. Theano. and torch for deep learning // Workshop track – ICLR. 2016. P. 1-11.
7. Kalman R. E. Mathematical description of linear dynamical systems // J. Soc. Indus. Appl. Math. 1963. Ser. A 1. P. 152–192.
8. Gao X.-D., et al. Control efficacy of complex networks // Sci. Rep. 2016. V 6. Article number: 28037.
9. Yang-Yu L., Jean-Jacques S., Barabasi A.-L. Controllability of complex networks // Nature. 2011. Vol 473. P. 167–173.
10. Gao J., Barzel B., Barabasi A.-L. Universal resilience patterns in complex networks // Nature. 2016. Vol 530. P. 307–312.
11. Pu C.-L., Pei W.-J., Michaelson A. Robustness analysis of network controllability // Physica. 2012. A 391. P. 4420–4425.
12. Nepusz T., Vicsek T. Controlling edge dynamics in complex networks // Nature Physics. 2012. V 8. P. 568–573.
13. Slotine J.-J., Liu Y.-Y. The missing link // Nature Physics. 2012. V 9. P. 1–2.
14. Poznyak A.S., Sanchez E.N., Yu W. Differential Neural Networks for Robust Nonlinear Control. Identification. State Estimation and Trajectory Tracking. – Singapore. World Scientific publishing. 2001. – 422 p.
15. Ghanim U., et al. Anti-phase calcium oscillations in astrocytes via inositol (1, 4, 5)-trisphosphate regeneration // Cell Calcium. 2006. № 39. P. 197–208.
16. Nazari S., et al. A digital implementation of neuron–astrocyte interaction for neuromorphic applications // Neural Networks. 2015. V. 66. P. 79–90.
17. Agulhon C., et al. What Is the Role of Astrocyte Calcium in Neurophysiology? // Neuron. 2008. V. 59. I. 6. P. 932–946.
18. Izhikevich E.M. Hybrid spiking models. // Philosophical transactions. 2010. Series A. Mathematical, physical, and engineering sciences. V. 368. № 1930. P. 5061–5070.
19. Izhikevich E.M. Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. Cambridge. Massachusetts London. England. The MIT Press. 2007. – 497 p.
20. Izhikevich E.M. Simple model of spiking neurons // IEEE Transactions on neural networks. 2003. V. 14. № 6. P. 1569–1572.
21. Pu C.-L., Pei W.-J., Michaelson A. Robustness analysis of network controllability // Physica. 2012. A 391. P. 4420–4425.
22. Kompanets I.N., Neevina T.A., Kompanets S.I. Method of parallel switching of optical channels // QUANTUM ELECTRON. 2012. V. 42 (12). С. 1093–1096.

Поступила 21.10.2016

#### Об авторах

**Калачев Алексей Александрович**, ведущий программист Федерального государственного автономного учреждения «Государственный научно-исследовательский институт информационных технологий и телекоммуникаций», a.kalachev@informika.ru;

**Краснов Андрей Евгеньевич**, доктор физико-математических наук, профессор, главный научный сотрудник Федерального государственного автономного учреждения «Государственный научно-исследовательский институт информационных технологий и телекоммуникаций», a.krasnov@informika.ru;

**Надеждин Евгений Николаевич**, доктор технических наук, профессор, главный научный сотрудник Федерального государственного автономного учреждения «Государственный научно-исследовательский институт информационных технологий и телекоммуникаций», e.nadezhdin@informika.ru;

**Никольский Дмитрий Николаевич**, кандидат физико-математических наук, доцент, ведущий научный сотрудник Федерального государственного автономного учреждения «Государственный научно-исследовательский институт информационных технологий и телекоммуникаций», d.nikolsky@informika.ru;

**Репин Дмитрий Сергеевич**, кандидат технических наук, начальник управления телекоммуникаций Федерального государственного автономного учреждения «Государственный научно-исследовательский институт информационных технологий и телекоммуникаций», r\_d\_s@informika.ru.