УДК 004.5
DOI: 10.26102/2310-6018/2025.48.1.014

Мониторинг состояния коммуникационных сетей на основе облачных вычислений в режиме реального времени

Амоа Куадио-кан Армел Жеафруа✉, Е.В. Сидоренко, Н.А. Рындин

Воронежский государственный технический университет, Воронеж, Российская Федерация

Резюме. При создании коммуникационной сети неизбежно возникают различные помехи, негативно сказывающиеся на ее эффективности. Отсутствие мер по устранению таких помех затрудняет оптимизацию сети. Среди проблем, вызванных помехами, проблема их блокировки является одной из наиболее существенных. Эта неразрешенная проблема может сделать невозможным успешное проектирование сети. Для решения проблем, связанных с тем, что традиционный метод имеет длительное время отклика на мониторинг перегрузки сети связи, а эффект обнаружения не идеален, предлагается метод мониторинга в реальном времени, основанный на облачных вычислениях для блокировки сети связи. Во-первых, устанавливается точка мониторинга сети связи, и приемник завершает процесс сбора коммуникационных данных. На основе собранных данных выполняется постоянный расчет трафика для определения наличия аварийного состояния блокировки в канале сети связи и определения точного местоположения точки блокировки. Таким образом, информация генерирует тревожное сообщение для получения результатов мониторинга. Экспериментально проанализированы время работы в режиме реального времени и точность метода мониторинга. Установлено, что метод мониторинга позволяет контролировать время задержки в пределах 0,2 с, а частота ошибок мониторинга является низкой.

Ключевые слова: облачные вычисления, телекоммуникации, перегрузка сети, мониторинг в режиме реального времени, точка мониторинга, управление системой, блокировка.

Для цитирования: Амоа Куадио-кан Армел Жеафруа, Сидоренко Е.В., Рындин Н.А. Мониторинг состояния коммуникационных сетей на основе облачных вычислений в режиме реального времени. *Моделирование, оптимизация и информационные технологии.* 2025;13(1). URL: https://moitvivt.ru/ru/journal/pdf?id=1809 DOI: 10.26102/2310-6018/2025.48.1.014

Real-time monitoring of communication networks based on cloud computing

Amoa Kouadio-kan Armel Geoffroy✉, E.V. Sidorenko, N.A. Ryndin

Voronezh State Technical University, Voronezh, the Russian Federation

Abstract. When creating a communication network, various obstacles inevitably arise that negatively affect its effectiveness. The lack of measures to eliminate such interference makes it difficult to optimize the network. Among the problems caused by interference, the problem of blocking them is one of the most significant. This unresolved issue may make successful network design impossible. In order to solve the problems that the traditional method has a long response time to monitor the congestion of the communication network and the detection effect is not ideal, a real-time monitoring method based on cloud computing for blocking the communication network is proposed. Firstly, a communication network monitoring point is established, and the receiver completes the communication data collection process. Based on the collected data, continuous traffic calculation is performed to determine whether there is an emergency blocking state in the communication network channel and determine the exact location of the blocking point. In this way, the information generates an alarm message to obtain the monitoring results. The real-time running time and the accuracy of the monitoring method are experimentally analyzed. It is found that the monitoring method can control the delay time within 0.2 s,

and the monitoring error rate is low.

Keywords: cloud computing, telecommunications, network congestion, real-time monitoring, monitoring point, system management, blocking.

For citation: Amoa Kouadio-kan Armel Geoffroy, Sidorenko E.V., Ryndin N.A. Real-time monitoring of communication networks based on cloud computing. *Modeling, Optimization and Information Technology*. 2025;13(1). (In Russ.). URL: https://moitvivt.ru/ru/journal/pdf?id=1809 DOI: 10.26102/2310-6018/2025.48.1.014

Введение

В процессе построения коммуникационной сети (КС) часто возникают различные помехи, которые влияют на производительность сети; если проблема помех не устранена, то оптимизацию сети будет трудно выполнить. Среди проблем, связанных с помехами, проблема блокировки помех является серьезной. Если она не будет решена, построение сети станет невозможным. Блокирование помех заключается в том, что при одновременном поступлении в приемник сигнала сильной помехи и полезного сигнала нелинейные компоненты канала связи приемника будут насыщены, что приведет к нелинейным искажениям и блокировке приемника, который выходит за пределы рабочего диапазона усилителя и микшера, что делает приемник неспособным к демодуляции. Как правило, это создает помехи в работе приемника, что приводит к невозможности нормально сообщать о нижнем уровне шума в сети связи. При слишком сильном сигнале использование полного сигнала также приведет к амплитудному сжатию и блокировке. Основной причиной блокировки является нелинейность устройства, особенно многоступенчатые эффекты интермодуляции. В то же время ограничение динамического диапазона приемника также приведет к возникновению помех. Блокировка приведет к сбою в работе приемника, а длительная блокировка может также привести к необратимому снижению производительности приемника.

Для подтверждения необходимо выполнить следующие действия:

Сдвиг частоты. В соответствии с принципом блокирования помех, для радиочастотного терминала, тип фильтра которого IF, сильный сигнал помех может быть исключен из приема радиочастотных сигналов путем сдвига частоты (изменения центральной частотной точки приема радиочастотных сигналов), так что уровень сигнала, попадающего на прием радиочастотных сигналов, составляет менее 40 дБм. Если RTWP сети связи снижен, можно определить, что сеть помех заблокирована.

Ослабление сигнала VGA. Для одномодовых станций, которые, как предполагается, подвержены помехам, если степень помех не очень велика, можно использовать ослабление сигнала VGA для определения того, подавлена ли сеть связи из-за возможности подавления сигнала помех. Если в процессе ослабления сигнала VGA RTWP сети связи резко меняется, это означает, что ячейка заблокирована.

Волновая ловушка. Режекторный фильтр (полосовой ограничивающий фильтр) может быть настроен таким образом, чтобы в разумной степени ослаблять сильный сигнал помех в соответствии с реальной ситуацией на объекте, так что общий принимаемый сигнал при радиочастотном приеме будет ниже порогового значения блокировки помех, что позволит определить, заблокирована ли сеть с нарушенным сигналом.

Отключение источника помех. Этот метод является самым простым. Если RTWP сети возвращается в нормальное состояние после обнаружения предполагаемого источника помех (сигнал помех не попадает в принимающую беспроводную сеть), это может свидетельствовать о том, что сеть связи заблокирована.

В статье представлена технология облачных вычислений, позволяющая осуществлять постоянный мониторинг перегрузки коммуникационной сети.

Сеть использует физические каналы связи для соединения изолированных рабочих станций или хостов с целью формирования канала передачи данных для совместного использования ресурсов и коммуникации [1, 2]. Однако с усложнением информации в сети связи и увеличением объема данных в сети связи возникают некоторые сетевые проблемы, такие как перегрузка сети [3, 4]. Производительность передачи данных по сети снижается из-за ограниченных ресурсов узлов хранения и пересылки.

Что касается архитектуры Интернета, то возникновение перегрузки является неотъемлемым атрибутом. Однако если условие блокировки имеет определенную длительность, то при исчерпании места в кэше маршрутизатор отбрасывает пакет только для того, чтобы сеть могла избежать состояния блокировки. В большой среде облачных вычислений, чтобы поддерживать нормальную работу сети и избегать негативного воздействия перегрузки на сеть, необходимо принять некоторые контрмеры для поддержания нормального режима работы сети связи.

Разработка метода мониторинга блокировки сети в режиме реального времени

Установка точки мониторинга сети

На определенном узле КС необходимо установить устройство сетевого мониторинга, чтобы получить данные о параметрах производительности всех каналов, связанных с этим узлом. Необходимо учитывать, на каких узлах установлены устройства сетевого мониторинга, и можно получить данные о производительности всех каналов и включить мониторинг.

Несколько устройств сетевого мониторинга могут быть разделены на несколько областей сетевого мониторинга. Одно устройство сетевого мониторинга может логически принадлежать нескольким областям сетевого мониторинга, то есть нескольким службам мониторинга производительности сети для нескольких пользователей, что позволяет сократить количество устройств и снизить затраты на проектирование [5, 6]. Это требует разумного расположения точек сетевого мониторинга и рационального разделения зон сетевого мониторинга и управления ими. Чтобы полностью реализовать функцию точки мониторинга, ее структура разделена на часть локальной вычислительной сети, серверную систему, различные рабочие станции, сервер терминалов, преобразователь протоколов и удаленное сетевое устройство. Структура точки мониторинга показана на Рисунке 1.

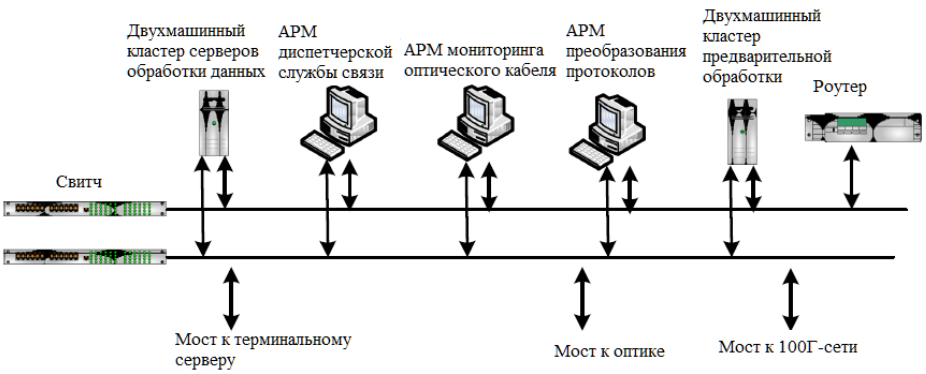

Рисунок 1 – Принципиальная схема структуры точки мониторинга
Figure 1 – Schematic diagram of the monitoring point structure

Точка мониторинга оснащена двумя сетевыми коммутаторами. Каждый сервер и рабочая станция оснащены двумя сетевыми интерфейсами, что гарантирует работу в случае отказа любого сетевого коммутатора. Настроены два сетевых сегмента,

внутренней сети и внешней сети, с помощью функции моста в программе Network Switch, чтобы гарантировать, что точки доступа внутреннего и внешнего сетевых сегментов не будут создавать помех друг другу. Функция подключения к удаленной локальной сети реализуется через маршрутизатор. Кластер серверов с двумя главными серверами устанавливает связь между двумя серверами и переводит оба сервера в режим работы кластера. Эти два сервера являются резервными копиями данных друг друга, и система гарантирует, что они не будут обнаружены ни на одном из серверов. После сбоя работа системы не нарушается.

Добавлены высокопроизводительные периферийные устройства ввода-вывода, такие как лазерные принтеры, для совместного использования с онлайн-пользователями. Данные с каждой подстанции передаются на центральную станцию через мост. Каждый мост оснащен двойным сетевым портом для обеспечения соединения с RC [7, 8]. В то же время предусмотрен двойной порт WAN для реализации двойного канала E1, основного и резервного, и обеспечивается автоматическое переключение двух каналов. Кроме того, требуется процессор преобразования протоколов для ввода данных из различных подсистем мониторинга связи по различным протоколам.

Радиочастотный приемник сбора данных о КС

Сбор и хранение данных являются основой для мониторинга производительности, позволяя получать и сохранять необработанную информацию. Радиочастотный канал оснащен регулируемым цифровым аттенюатором и VGA-интерфейсом, обеспечивающими достаточную динамику для выполнения требований по блокировке. Однако, если блокирующий сигнал находится далеко от рабочей частоты, он может находиться в непосредственной близости от АЦП или другой полосы дискретизации Найквиста и быть отобран АЦП [9]. Если частота помех выбрана и попадает в диапазон «использовать полный сигнал», что приводит к искажению сигнала, радиочастотный и цифровой фильтры не подавляют сигнал. В этом случае необходим фильтр промежуточной частоты, чтобы предотвратить цифровую выборку нежелательных сигналов. Структура таблицы данных мониторинга производительности показана на Рисунке 2.

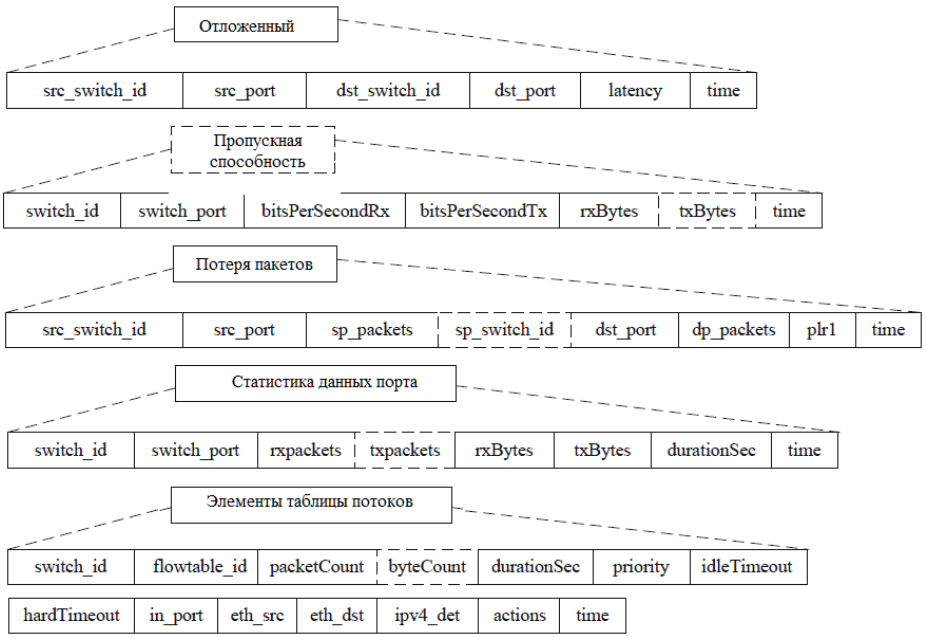

Рисунок 2 – Схема цепочки сбора коммуникационных данных
Figure 2 – Communication data collection chain diagram

Создание вышеуказанной таблицы и подключение к базе данных в контроллере осуществляется с помощью интерфейса JDBC. В сочетании со скоростью, дешевизной и открытым исходным кодом базы данных MySQL, использование JDBC в качестве интерфейса MySQL стало распространенным явлением. JDBC – это стандартный API для языка Java для взаимодействия с базами данных. Приложения Java могут напрямую обращаться к базе данных через этот стандартный интерфейс. JDBC также поддерживает возможность единообразного доступа к различным типам баз данных. Создайте поток сбора данных о производительности в режиме реального времени и сохраните собранные данные. Модуль взаимосвязи между реализацией сбора и хранения данных в основном разделен на три пакета реализации:

– пакет подключения к базе данных CLOS.sql, основная функция которого заключается в реализации функций подключения и управления базой данных;

– пакет Java Bean colSto.bean, который в основном определяет структуру данных различных таблиц данных о производительности;

– пакет бизнес-операций cloSto.op, который в основном выполняет функции приема и хранения данных.

Когда модуль мониторинга производительности получает параметры производительности, он вызывает соответствующий класс в пакете бизнес-операций. Операции по хранению данных для обеспечения распределенного хранения собранных данных в режиме реального времени.

Вычисление потока данных в сети связи в режиме реального времени

Трафик данных, собранный в сети связи, рассчитывается в режиме реального времени, и вычислительная структура в режиме реального времени сравнивается с максимальным объемом памяти и максимальной нагрузкой на пропускную способность, и проверяется, может ли трафик данных плавно достигать указанного объема памяти через полосу пропускания для хранения данных [10]. Расчетная формула для расчета потока данных в сети связи в режиме реального времени выглядит следующим образом:

$$Q = min\left\{\sum_{k\in I_l}\frac{F_l}{\sum_{k\in I_l}P_{lk}y_{lk}-F_l} + G\sum_{\substack{l\in L\\k\in I_l}}(S_{lk}y_{lk}+d_i m_{lk}) + V\sum_{\substack{l\in L\\k\in I_l}}\left(C_{lk}F_{ly_{lk}}\right)\right\}, \quad (1)$$

где F_l – локальный трафик данных по каналу КС; Q – общий трафик данных; I_l – набор индикаторов модели потенциального соединения для первой связи; P_{lk} – индекс модели первого звена, выбранного в качестве пропускной способности связи k; S_{lk} – возможный набор маршрутов для узла первой связи; C_{lk} – коэффициент линейной переменной при выборе индекса модели первого звена; d_i – длина первой связи; m_{lk} – соединитель; I_l – скорость поступления пакетов; G, V – фиксированный весовой коэффициент; L – множество всех связей КС.

Ограничения:

$$\sum_{k\in S_p} x_p = 1 \ \forall p \in \Pi, \quad (2)$$

$$\sum_{k\in I_l} y_{lk} = 1 \ \forall l \in L, \quad (3)$$

где p – набор всех пар узлов связи в сети; x_p – переменная оптимизации, равна 1, когда маршрут выбран, иначе 0; y_{lk} – переменная оптимизации. Когда выбран индекс модели первого канала «ask», значение равно 1, в противном случае оно равно 0; может быть получен коммуникационный трафик в реальном времени по определенному каналу, а также могут быть получены максимальный предел и пропускная способность дискового пространства. Для определения расхода потока в нормальных условиях сравнивается максимальный предел производительности.

Распознавание аномальных блокировок и анализ характеристик

Помимо того, что трафик данных в КС, превышающий максимальную пропускную способность КС, может привести к перегрузке сети, аномалии передачи данных также являются другой причиной перегрузки. Чтобы осуществлять мониторинг аномальных узлов передачи данных в режиме реального времени, необходимо автоматическое инспектирование узлов. Программа мониторинга узла считывает IP-адрес устройства из базы данных и циклически отправляет команду проверки связи. Полученный результат фиксируется в рабочем состоянии устройства узла. Если время ожидания результата истекло, это означает, что устройство узла работает неправильно. Как только сетевой трафик станет ненормальным, IP-адрес и распределение портов изменятся. Если произойдет ошибка конфигурации сети, исходный IP-адрес и IP-адрес назначения увеличатся, что приведет к резкому увеличению количества пакетов, отправляемых хостом. В соответствии с этой особенностью для анализа дисперсии характеристик распределения трафика используется метод матрицы сетевого трафика. Предположим, что характеристика трафика есть A, общее количество выборок есть B, количество выбранных выборок есть C, а количество повторений конкретной характеристики трафика i есть n_i. Таким образом, выборка характеристик трафика может быть осуществлена следующим образом:

$$F(x) = -\sum_{i=1}^{C} \left(\frac{n_i}{B}\right) log_2 \left(\frac{n_i}{B}\right). \quad (4)$$

Если все выборки дают одинаковый результат, то $F(x)=0$; если все выбранные выборки имеют большую степень разброса, то $F(x)=log_2 C$ может описывать аномальное поведение различных характеристик потока, а затем выполнять обработку захвата пакетов.

Когда система мониторинга трафика фиксирует передаваемый кадр Ethernet, ей необходимо сначала проанализировать пакет данных, а затем извлечь соответствующие данные и сохранить результат извлечения в БД, что удобно для анализа аномального сетевого трафика в режиме реального времени. Чтобы обеспечить точность системного перехвата, необходимо сначала прочитать конфигурационный файл; затем установить соединение с БД, зарегистрировать источник данных ODBC, использовать функцию openDatabase() в файле скрипта Python для подключения к БД; настроить драйвер перехвата для создания различных типов, создать таймеры и потоки, мониторинг аномального трафика в режиме реального времени с использованием таймеров; наконец, создать сервер мониторинга в режиме реального времени и вызвать функцию Bind(), чтобы получить IP-адрес сервера мониторинга из файла конфигурации, тем самым завершив процесс перехвата пакетов. Пакет данных о ненормальном трафике может быть перехвачен в режиме реального времени, и может быть добавлена функция отображения мониторинга в режиме реального времени, чтобы пользователь мог просматривать результат захвата пакета в режиме реального времени, а затем находить аномальный узел.

Результаты эксперимента

Чтобы проверить целесообразность исследования метода мониторинга блокировки КС в режиме реального времени, были проведены эксперименты. В качестве экспериментальных объектов выбираются 50 наборов данных за определенный период времени, и данные о нормальном состоянии получаются в соответствии с историческими записями за предыдущие периоды, после чего выполняется стандартизированная обработка. Использовался симулятор Mininet для создания топологии сети связи на

виртуальной машине и подключения к удаленному контроллеру для реализации построения сетевой среды связи. В качестве экспериментальных показателей были взяты ошибка мониторинга в режиме реального времени и время задержки связи в КС. Традиционный метод мониторинга сравнивается с мониторингом трафика в сети связи в режиме реального времени. В ходе эксперимента сравниваются традиционный метод мониторинга и разработанный метод мониторинга в режиме реального времени, и результаты сравнения показаны на Рисунке 3.

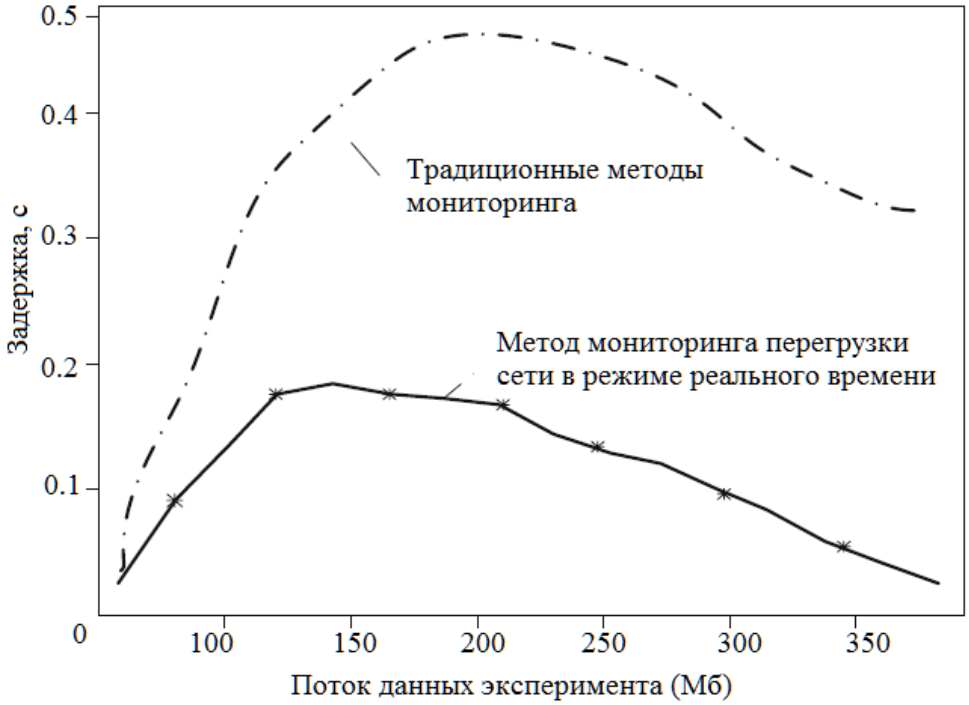

Рисунок 3 – Результаты сравнения времени задержки связи
Figure 3 – Results of comparison of communication delay time

Время задержки при использовании метода блокировки КС для мониторинга в режиме реального времени напрямую отражает эффективность метода. Из сравнения экспериментальных результатов видно, что время задержки связи при использовании традиционного метода мониторинга будет увеличиваться с увеличением объема данных. Несмотря на явную тенденцию к сокращению задержки при достижении количества 280 М, время задержки всегда превышает 0,3 с. Напротив, в методе мониторинга в режиме реального времени, блокирующем сеть связи, используется метод сбора данных и расчета в режиме реального времени, поэтому экспериментальные результаты показывают идеальный результат задержки. Время задержки всегда контролируется в пределах 0,2 с, а когда поток данных превышает 250 м, наблюдается явная тенденция к снижению, что полностью отражает характер метода мониторинга в режиме реального времени. Кроме того, точность метода мониторинга результатов мониторинга также очень важна. Результаты информации, полученной с помощью разработанного метода мониторинга в режиме реального времени, могут быть точно определены в том месте, где происходит блокировка, и рассчитано значение блокировки, что удобно для своевременной блокировки аварийных ситуаций. Метод имеет низкую погрешность мониторинга и высокую точность.

Для дальнейшей проверки точности мониторинга с помощью этого метода используются традиционный метод и разработанный. Результаты приведены в Таблице 1.

Таблица 1 – Точность мониторинга КС
Table 1 – Accuracy of CN monitoring

Время мониторинга (мин.)	Точность мониторинга КС (%)	
	Традиционный метод	Предложенный метод
10	67	89
20	69	96
30	71	93
40	70	92
50	65	95
Среднее значение	68,4	93

Согласно Таблице 1, точность мониторинга данных зависит от времени мониторинга. Когда время мониторинга составляет 10 минут, показатель точности мониторинга данных сети связи традиционным методом составляет 67 %, а предложенным методом – 89 %. При продолжительности мониторинга 50 минут точность мониторинга данных в КС традиционным методом составляет 65 %, а в предложенном методе – 95 %. Средний показатель точности традиционного метода составляет 68,4 %, в то время как предложенного – 93 %.

Заключение

Таким образом, в среде облачных вычислений для обеспечения стабильности работы компьютерной КС необходимо обеспечить, чтобы сетевой трафик всегда находился в нормальном состоянии. С помощью метода мониторинга КС в режиме реального времени можно отслеживать текущее состояние всей КС и своевременно обнаруживать проблему блокировки сети, что облегчает ежедневное управление и техническое обслуживание. В процессе разработки метода мониторинга было обнаружено, что, хотя метод обладает высокой точностью мониторинга, долгосрочную стабильную работу метода еще предстоит изучить, и в будущем разработать высокопроизводительный метод мониторинга для поддержки мониторинга сети связи в режиме реального времени.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Jayakumari D.S., Mathusoothana S Kumar R, Venkadesh P., Divya S.V. *Computer Networks.* San International; 2024. https://doi.org/10.59646/cn/283
2. Yan K., Ma W., Sun S. Communications and Networks Resources Sharing in 6G: Challenges, Architecture, and Opportunities. *IEEE Wireless Communications.* 2024;31(6):102–109. https://doi.org/10.1109/MWC.003.2400038
3. Liu H. Research on control method of blocking jamming in HF communication system. *Digit. Technol. Appl.* 2019;37(1):29–30.
4. Chen Z., Dai Y., Liu Y. Crack propagation simulation and overload fatigue life prediction via enhanced physics-informed neural networks. *International Journal of Fatigue.* 2024;186. https://doi.org/10.1016/j.ijfatigue.2024.108382
5. Edwards J. Network Monitoring and Defense. In: *Critical Security Controls for Effective Cyber Defense: A Comprehensive Guide to CIS 18 Controls.* Berkeley: Apress; 2024. pp. 371–404. https://doi.org/10.1007/979-8-8688-0506-6_13
6. Rychlicki M., Kasprzyk Z., Pełka M., Rosiński A. Use of Wireless Sensor Networks for Area-Based Speed Control and Traffic Monitoring. *Applied Sciences.* 2024;14(20). https://doi.org/10.3390/app14209243

7. Liu P., Cai Y., Lu G. Space Environment Data Transfer System Based on BBR Congestion Control Algorithm. *Chinese Journal of Space Science.* 2019;39(1):111–117. https://doi.org/10.11728/cjss2019.01.111
8. Paul J.B.J., Rekh A.S., Prabakaran E.P.G. A novel semi elliptical slotted dual port rectenna for RF energy harvesting. *Analog Integrated Circuits and Signal Processing.* 2025;122. https://doi.org/10.1007/s10470-025-02323-1
9. Cardinale C., Brunton S.L., Colonius T. Spectral proper orthogonal decomposition using sub-Nyquist rate data. arXiv. URL: https://doi.org/10.48550/arXiv.2501.02142 [Accessed 12th December 2024].
10. Wei B., Xiao L., Wei W., Song Ya., Yan B., Huo Zh. A high-bandwidth and low-cost data processing approach with heterogeneous storage architectures. *Personal and Ubiquitous Computing.* 2023;27(2):159–176. https://doi.org/10.1007/s00779-020-01383-6

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Амоа Куадио-кан Армел Жеафруа, аспирант, Вронежский государственный технический университет, Воронеж, Российская Федерация.
e-mail: csit@bk.ru

Amoa Kouadio-kan Armel Geoffroy, postgraduate student, Vronezh State Technical University, Voronezh, the Russian Federation.

Сидоренко Евгений Васильевич, аспирант, Вронежский государственный технический университет, Воронеж, Российская Федерация.
e-mail: csit@bk.ru

Evgeny V. Sidorenko, postgraduate student, Vronezh State Technical University, Voronezh, the Russian Federation.

Рындин Никита Александрович, доктор технических наук, доцент, профессор кафедры искусственного интеллекта и цифровых технологий, Воронежский государственный технический университет, Воронеж, Российская Федерация.
e-mail: csit@bk.ru
ORCID: 0000-0002-0774-2352

Nikita A. Ryndin, Doctor of Engineering Sciences, Docent, Professor, Department of Artificial Intelligence and Digital Technologies, Voronezh State Technical University, Voronezh, the Russian Federation.

Статья поступила в редакцию 26.01.2025; одобрена после рецензирования 03.02.2025; принята к публикации 05.02.2025.

The article was submitted 26.01.2025; approved after reviewing 03.02.2025; accepted for publication 05.02.2025.