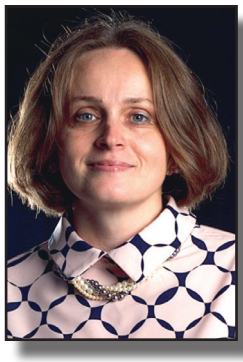
УДК 004.896:629.4.014
DOI: https://doi.org/10.30932/1992-3252-2021-19-1-18-46

Построение архитектуры интеллектуальной системы управления городской рельсовой транспортной системой



Виктор АЛЕКСЕЕВ **Леонид БАРАНОВ** **Максим КУЛАГИН** **Валентина СИДОРЕНКО**

Алексеев Виктор Михайлович *– Российский университет транспорта, Москва, Россия.*
Баранов Леонид Аврамович *– Российский университет транспорта, Москва, Россия.*
Кулагин Максим Алексеевич *– АО «Научно-исследовательский институт железнодорожного транспорта», Москва, Россия.*
Сидоренко Валентина Геннадьевна *– Российский университет транспорта, Москва, Россия*.*

Рост объёма пассажирских перевозок в условиях крупных городских агломераций и мегаполисов эффективно обеспечивается объединением общественного транспорта и городских линий железных дорог. Управление движением в этих условиях требует создания интеллектуальных централизованных многоуровневых систем управления, реализующих заданные показатели качества, комфорта и безопасности перевозок пассажиров. Современные системы управления дополнительно решают задачи экономии энергии на тягу подвижного состава, являются фундаментом и составной частью систем цифровизации городского транспорта и города в целом. Построение систем, решающих задачи планирования и управления движением, реализуется с применением алгоритмов, использующих методы искусственного интеллекта, принципы иерархического построения централизованных систем, возможности технологии Big Data. В этих условиях необходимо учитывать возросшие требования не только к программному обеспечению, но и к теоретическим и практическим решениям организации сети.

В данной статье рассматриваются вопросы формирования архитектуры и требований к разрабатываемым приложениям и их интеграции с базами данных для создания централизованной интеллектуальной системы управления городской рельсовой транспортной системой (ЦИСУ ГРТС). В статье предлагаются оригинальные подходы к архитектуре сети, маршрутизации информационных потоков и программному обеспечению ЦИСУ ГРТС. В основе построения маршрутизации лежит использование полносвязной сети. Это позволяет значительно увеличить пропускную способность сети и выполнить требования по защите информации, так как информационные потоки формируются на базе однотипных протоколов, что препятствует образованию скрытых каналов передачи. Реализация ядра с использованием полносвязности позволяет по меткам информационных потоков заранее сформировать маршруты обмена информации между серверами и приложениями, развёрнутыми в ЦИСУ ГРТС. Использование шифрованных меток информационных потоков значительно усложняет проведение атак и организацию сбора информации о структуре сети.

Платформы для разработки интеллектуальных систем управления (ИСУ), к которым относится ЦИСУ ГРТС, огромные вычислительные мощности, хранилища данных и новые фреймворки становятся всё более доступными для учёных и разработчиков и позволяют быстро развиваться ИСУ. В статье рассматриваются вопросы взаимодействия приложений с базами данных на основе комбинации нескольких подходов, используемых в области Big Data, обосновывается сочетание методологии интернета вещей (IoT) и микросервисной архитектуры. Данная комбинация позволит выделить в системе бизнес-процессы и сформировать потоковую обработку данных, требующих оперативного анализа человеком, что доказывается соответствующими примерами.

Таким образом, целью статьи является формализация принципов организации информационного обмена ЦИСУ ГРТС и автоматизированных систем управления (АСУ) железнодорожных компаний (в нашем случае – на примере ОАО «РЖД»), организаций, предоставляющих услуги ГРТС, и городских органов управления, реализация сформулированных принципов в архитектуре ЦИСУ ГРТС и формализация принципов организации микросервисной архитектуры программного обеспечения ЦИСУ ГРТС. Основными методами исследования являются теория графов, методы Big Data, IoT.

Ключевые слова: городской рельсовый транспорт, интеллектуальное управление, микросервисная архитектура, архитектура сети, большие данные, архитектура программного обеспечения, городской рельсовый транспорт.

---

*Информация об авторах:
Алексеев Виктор Михайлович – доктор технических наук, профессор кафедры управления и защиты информации Российского университета транспорта, Москва, Россия, alekseevvm@rambler.ru.
Баранов Леонид Аврамович – доктор технических наук, профессор, заведующий кафедрой управления и защиты информации Российского университета транспорта, Москва, Россия, Baranov.miit@gmail.com.
Кулагин Максим Алексеевич – заместитель начальника отдела разработки технологических информационных систем АО «Научно-исследовательский институт железнодорожного транспорта» (АО «ВНИИЖТ»), Москва, Россия, kulagin.maxim@vniizht.ru.
Сидоренко Валентина Геннадьевна – доктор технических наук, профессор кафедры управления и защиты информации Российского университета транспорта, Москва, Россия, ✉ Valenfalk@mail.ru.

Статья поступила в редакцию 20.01.2021, принята к публикации 26.02.2021.

For the English text of the article please see p. 33.

ВВЕДЕНИЕ

Рост объёма пассажирских перевозок в условиях крупных городских конгломераций и мегаполисов эффективно обеспечивается объединением общественного транспорта и городских линий железных дорог. Управление движением в этих условиях требует создания интеллектуальных централизованных многоуровневых систем управления, реализующих заданные показатели качества, комфорта и безопасности перевозок пассажиров. Современные системы управления дополнительно решают задачи экономии энергии на тягу подвижного состава, являются фундаментом и составной частью систем цифровизации городского транспорта и города в целом.

Построение систем, решающих задачи планирования и управления движением, реализуется с применением алгоритмов, использующих методы искусственного интеллекта, принципы иерархического построения централизованных систем, возможности технологии Big Data. В этих условиях необходимо учитывать возросшие требования не только к программному обеспечению, но и к теоретическим и практическим решениям организации сети. В первую очередь необходимо отметить тот факт, что ИСУ обрабатывают значительно больший объём информации, поскольку используют методы распознавания для контроля доступа на объекты инфраструктуры: это фоновое распознавание лиц персонала, детектирование фактов вторжения, а также увеличение объёмов технологической видео- и аудиосвязи, данных о параметрах большого числа объектов. В системах безопасного управления фиксируются все переговоры персонала. В этой связи значительно возрастают объёмы передаваемой информации. Если для аудиосвязи для обеспечения качества передаваемой информации достаточно использования технологий сжатия информации в каналах со скоростью передачи информации до 64 кБ/с, то для поддержки видеосвязи требуется как минимум в десять раз лучшее качество. Модели управления движением пригородного железнодорожного транспорта в черте городов (например, на Московских центральных диаметрах (МЦД) и Московском центральном кольце (МЦК)) [1], на линиях метрополитена и скоростного трамвая [2]) используют значительные объёмы информации, поступающие от систем контроля доступа на инфраструктуру, поскольку необходимо оперативно управлять остановкой подвижного состава при возникновении экстренных ситуаций [3]. Эти системы строятся с использованием моделей распознавания образов, и, как следствие, это ведёт к росту объёмов передаваемого трафика как по беспроводным, так и по оптическим каналам передачи информации. В этой связи объёмы хранимой информации исчисляются в петабайтах. Следует отметить, что изменились не только объёмы передаваемой информации, но и сущность внешних и внутренних атак на информационные ресурсы [4–7]. Если ранее атака предполагала внесение некоторого сбоя в работу системы, то сегодня упор делается на тайный сбор информации и передачу информации скрытым способом.

Построение сети стандартными способами [8; 9], которые были рекомендованы и использованы на практике при реализации систем нового класса таких, как ИСУ, не могут быть применены. Как правило, сети строят по стандартной схеме с двумя или тремя центральными коммутаторами, к которым подключают периферийные коммутаторы, связывающие различные объекты. К этим объектам относятся серверы, приложения, базы данных и субъекты (диспетчеры, программисты, администраторы различных подсистем: безопасности, сети LTE, баз данных и др.). Маршрутов для организации потоков информации при трёх коммутаторах всего два. Это не позволяет дифференцировать трафик передачи информации даже при использовании визуализации в локальной сети. В результате реализация информационных потоков, ориентированных на использование прямых каналов передачи от объектов – контроллеров системы управления к объектам – серверам, практически не выполнима. Между тем ИСУ требуют агрегирования большого количества данных и высокой скорости передачи информации, предполагающей использование оптических интерфейсов для связи с базами хранения информации. В этой связи построение ядра сети ИСУ с большим числом центральных коммутаторов откры-

вает новые возможности увеличения скорости, объёма передаваемой информации, организации взаимодействия приложений и баз данных, а также организации защиты информации.

Известно, что мощность сети определяется связями между узлами. Количество связей *B* определяется следующим образом: $B = N(N-1)/2$, где N – количество узлов. Количество маршрутов M в этом случае равно $M = B - 1$. Это позволяет осуществить канальную передачу информации от большого числа объектов (объекты автоматики и телемеханики, микропроцессорная централизация (МПЦ), рельсовые цепи на станциях и перегонах, стрелочные переводы), с которых необходимо собирать информацию, передавать в базу данных и производить их обработку соответствующим приложением [10]. Применение полносвязных узлов в ядре сети позволяет организовать информационные потоки доставки информации по прямым каналам с использованием оптических линий связи, что в десятки раз повышает скорость и объёмы передаваемой информации. Необходимо отметить тот факт, что дифференцирование потоков по признакам позволяет максимально затруднить передачу по скрытым каналам, так как количество признаков, с помощью которых можно передать информацию путём перестановок, ограничено, поскольку в каждом потоке используется практически один протокол. В организации сети соединение информационных потоков на отдельных узлах происходит крайне редко, что является необходимым условием высокой скорости передачи.

Модель сети ИСУ строится на основе дискреционного метода [11], что означает однозначную привязку субъекта к объекту. На практике это означает, что за субъектом закреплены персональные объекты в сети.

Целью исследования является реализация методов интеллектуальной защиты информационных потоков путём организации временных доверенных маршрутов на основе меток и использования полносвязного ядра сети ИСУ, а также взаимодействия приложений системы интеллектуального управления на основе принципов микросервисной архитектуры программного обеспечения.

Исходя из поставленной цели в статье рассматриваются вопросы организации структуры сети, организации временных доверенных маршрутов для интеграции подсистем интеллектуального управления городским транспортом, а также микросервисной архитектуры, направленной на реализацию в интеллектуальной системе управления программного взаимодействия подсистем в корпоративной сети. Для решения поставленных задач используются *методы* системного анализа, теории графов и компьютерных наук.

РЕЗУЛЬТАТЫ

Структура сети ИСУ

В ИСУ осуществляется управление движущимися объектами по беспроводной сети 4*G* (*LTE*) [10; 11]. По сети *LTE* осуществляется передача разнородного трафика: голос, видео и данные. В этой связи использование технологии мультисервисной сети *MPLS* является необходимым и оправданным условием. Дискреционную модель сети *MPLS* опишем через декартово произведение объектов и субъектов:

$$(O_i^{MPLS} \bullet O_j^{MPLS}) \bullet S_k,\ i \neq j,\ k = 1,\ m,$$

где S_k – субъекты, персонал;

m – число субъектов;

$(O_i^{MPLS} \bullet O_j^{MPLS})$ – декартовое произведение объектов в сети *MPLS*.

Сеть *MPLS* функционирует на специализированном оборудовании, в операционной системе которого реализованы специальные функции обработки пакетов. К пакету добавляется метка, содержащая префикс *IP*-адреса, порт входа и порт выхода на маршрутизаторах *PE* (провайдерский маршрутизатор) и *P* (маршрутизатор ядра сети *MPLS*). Маршруты заранее записываются в таблицу маршрутизации. Метка длиной 33 бита быстро обрабатывается, в связи с этим скорость доставки увеличивается. На маршрутизаторах *PE* происходит снятие метки, и далее пакеты поступают в маршрутизатор *CE* (клиентский маршрутизатор) и в локальную сеть.

Рассмотрим работу приложения, развёрнутого на объекте O_i^{MPLS} в сети *MPLS*[1]

[1] Методический документ «Меры защиты информации в государственных информационных системах» (утв. Федеральной службой по техническому и экспортному контролю 11 февраля 2014 г.). [Электронный ресурс]: https://www.garant.ru/products/ipo/prime/doc/70491518/. Доступ 17.02.2021.

[4–6]. Приложение Pr_l на объекте O_i^{MPLS} инициализируется субъектом и отправляет по интерфейсу *Ethernet*-пакет, содержащий *MAC*-адреса отправителя и получателя, а также область *DATA*, где помещаются данные используемого приложением Pr_l протокола. В области *DATA Ethernet*-пакета есть расширение, в которое помещается идентификатор N_{Pr_r} информационного потока, создаваемого приложением Pr_l. Идентификатор представляет шифрованную запись *MAC*-адреса объекта O_i^{MPLS} и номер приложения, работающего в сети, а также параметр t_0 – начало передачи потока и t_{live} – параметр жизни или продолжительности информационного потока. Назначение параметров t_0 и t_{live} будет раскрыто далее. Для формирования шифра кода используем симметричный ключ *kl*, заранее заданный для приложения и сервера:
$(MAC_{O_i^{MPLS}}, N_{\mathrm{Pr}_r}, t_0, t_{live})_{E_{kl}}$,

где E_{kl} – шифрование E с секретным ключом *kl*.

Далее идёт стандартная процедура обработки в сети *MPLS*. Пакет поступает на маршрутизатор *CE*, который отправляет его на маршрутизатор *PE*, где к пакету приклеивается метка, и по префиксу *IP*-адреса он направляется к выходному маршрутизатору *PE*, где метка снимается.

При прохождении сети *MPLS* происходит замена *MAC*-адреса первоначального источника, но он сохраняется в расширенной области *DATA*.

После снятия метки пакет с данными поступает на *Ethernet*-интерфейс коммутаторов узлов ядра сети, где в таблице маршрутизации для $(MAC_{O_i^{MPLS}}, N_{\mathrm{Pr}_r}, t_0, t_{live})_{E_{kl}}$ заранее прописаны возможные варианты движения по маршруту. Организует маршрутизацию специализированный сервер – монитор безопасности объектов (МБО).

Объекты сети *MPLS* O_i^{PE}, подсоединённые к выходам на маршрутизаторы *PE* и связанные с частью объектов ядра сети ИСУ O_j^{ISC}, можно описать через Декартово произведение:
$O_i^{PE} \bullet O_j^{ISC}, i \neq j$.

На любом из портов объектов сети *MPLS* O_i^{PE} может появиться пакет с идентификатором:
$(MAC_{O_i^{MPLS}}, N_{\mathrm{Pr}_r}, t_0, t_{live})_{E_{kl}}$.

Возможна реализация двух вариантов маршрутизации информационных потоков.

Первый вариант. Сервер МБО, зная все МАС-адреса и номера, присвоенные приложениям N_{Pr_r}, формирует маршруты по сети ядра ИСУ. Маршрут описываем через начальную и конечную точки:
$M = M(O_j^{ISC}, O_k^{ISC})$,

где j и k – начальные и конечные точки маршрута.

Для выбора кратчайшего маршрута применим алгоритм Дейкстры с параметром $C = 1$ и матрицей *[Z]* контроля занятости узла в ранее составленном маршруте. В качестве маршрута выбирается минимальный по расстоянию до конечного объекта:

$$d(M) = \sum_{i=1}^{n} C(O_j^{ISC}, O_k^{ISC}),$$

где $M(O_j^{ISC}, O_k^{ISC})$ – произвольный маршрут от начальной j-й до конечной k-й точки. Конечная точка доставки (сервер приложений, база данных или другие объекты) определяется не по *IP*-адресу в области *DATA*, а по его *MAC*-адресу (определен заранее), $C(O_j^{ISC}, O_k^{ISC})$ – расстояние от начальной j-й до конечной k-й. Реализация алгоритма маршрутизации представлена в табл. 1.

В описании модели выбора выходного порта используется параметр контроля занятости узла матрица [Z], строки которой соответствуют портам объекта, а столбцы – объектам. На пересечении столбцов и строк ставится 0, если порт занят, в противном случае – 1. Если порт уже занят в маршруте, то при выборе маршрута для множества объектов, работающих с другим приложением, выбирается другой свободный порт объекта.

Рассмотрим назначение параметров времени начала t_0 и продолжительности существования информационного потока t_{live}. Время начала может быть заранее согласовано в приложении, определены начало и продолжительность передачи информационного потока [7; 12; 13]. Эта информация посылается приложением на сервер МБО и используется им для анализа идентификатора потока, например, при осуществлении опроса датчиков-контроллеров состояния элементов инфраструктуры, подвижных объектов, объектов

Таблица 1

Алгоритм формирования маршрута (составлен авторами)

Идентификатор	Объект ядра	Входной порт	Выходной порт
$(MAC_{O_i^{MPLS}}, N_{\mathrm{Pr}_r}, t_0, t_{live})_{E_{kl}}$	O_j^{ISC}	e_i, i = 1, n	$e_k \times [Z], k = 1, n, k \neq i$
$(MAC_{O_{i+k}^{MPLS}}, N_{\mathrm{Pr}_{r+k}}, t_0, t_{live})_{E_{kl}}$	O_{j+k}^{ISC}	e_i, i = 1, n	$e_k \times [Z], k = 1, n, k \neq i$
Таблица заполняется в соответствии со структурной схемой сети			

систем автоматики и др. Вместе с этим может быть одиночный запрос от сервера приложения к объекту контроллеру, где также прописывается время начала и продолжительность существования потока ответа. Аналогично приложение сервера информирует приложение обеспечения безопасности объектов. Временны**е** параметры информационного потока для системы безопасности важны для того, чтобы определить время существования маршрута в ядре сети ИСУ. Параметры t_0 *и* t_{live} задаются приложением, определяются из необходимых потребностей на основании знания значений параметров контроля, которые используются в приложении. По окончании времени t_{live} происходит отмена маршрута. Это означает, что любая атака с использованием *MAC*-адреса внутри сети будет невозможна, поскольку доверенный маршрут временный и был разобран.

Второй вариант. Модель организации маршрутов основывается на использовании локального назначения *MAC*-адресов коммутатора по его портам администратором. Метка о локальном управлении *MAC*-адресом ставится в конце, и *MAC*-адрес имеет вид: xxxx.xxxx.xxxx.xxxx.xxxx.xx01. Таблица маршрутов при втором подходе строится на основе: *MAC*-адресов, порта, метки m_l. Метка m_l назначается на основе *IP*-адреса получателя в области *DATA*-пакета *Ethernet*. При этом в сети назначается протокол *IEEE* 802.1*q*, где содержится байт с тегом метки m_l, байт с типом протокола и указателем, что в сети используются пакеты протокола *IEEE* 802.1*q*. В таблице маршрутов сравнивается метка, назначенная заранее сервером МБО, и метка в протоколе *IEEE* 802.1*q*, в результате пакет направляется на соответствующий порт коммутатора. При достижении конечной точки доставки информационная метка удаляется, и пакет с данными поступает на сервер. Отметим, что применение данной технологии совпадает с виртуализацией сети и при полносвязности позволяет изолировать информационные потоки. Также стоит заметить, что это является одним из условий реализации сети и информационного взаимодействия для критически важных объектов Российской Федерации.

Рассмотрим модель взаимодействия приложений (различных задач управления ИСУ), которыми управляют диспетчеры ГРТС, администраторы сетей *LTE*, сетей подсистем ГРТС (метрополитена, пригородного железнодорожного транспорта в черте городов, скоростного трамвая):

$$(O_i^{ISC} \cdot O_j^{ISC})_{coreISC} \cdot (U_i(O_i^{pr} \cdot S_j^{ds}) + U_i(O_i^{adm} \cdot S_j^{adm}) +$$
$$+ U_i(O_i^{db} \cdot S_j^{db}) +$$
$$+ \left[U_i(O_i^{MPLS} \cdot O_j^{MPLS}) \cdot S_j^{sp}\right]_{rail} + \left[U_i(O_i^{MPLS} \cdot O_j^{MPLS}) \cdot S_j^{sp}\right]_{metro} +$$
$$+ \left[U_i(O_i^{MPLS} \cdot O_j^{MPLS}) \cdot S_j^{sp}\right]_{tram} + O^{id} + O^{usc}), i \neq j,$$

где $(O_i^{ISC} \cdot O_j^{ISC})_{coreISC}$ – ядро сети ИСУ;

$U_i(O_i^{pr} \cdot S_j^{ds})$ – централизованные системы управления движением по видам ГРТС: метрополитен, пригородный железнодорожный транспорт в черте городов, скоростной трамвай;

$U_i(O_i^{adm} \cdot S_j^{adm})$ – объекты, связанные с администраторами системы ИСУ (системные, сетевые);

$U_i(O_i^{db} \cdot S_j^{db})$ – объекты, связанные с администраторами баз данных;

S_j^{sp} – обслуживающий персонал;

$[U_i(O_i^{MPLS} \cdot O_j^{MPLS}) \cdot S_j^{sp}]_{rail}$ – сеть централизованной системы управления движением электропоездов пригородного железнодорожного транспорта в черте городов (ЦСУ-ДЭП), объединяющая центр ситуационного управления движением электропоездов

пригородного железнодорожного транспорта в черте городов и интегрированные подсистемы верхнего функционального уровня ЦСУДЭП;

$[U_i(O_i^{MPLS} \cdot O_j^{MPLS}) \cdot S_j^{sp}]_{metro}$ – сеть централизованной системы управления движением поездов метрополитена (ЦСУДПМ), объединяющая центр ситуационного управления движением поездов метрополитена и интегрированные подсистемы верхнего функционального уровня ЦСУДПМ;

$[U_i(O_i^{MPLS} \cdot O_j^{MPLS}) \cdot S_j^{sp}]_{tram}$ – сеть централизованной системы управления движением скоростного трамвая (ЦСУДСТ), объединяющая центр ситуационного управления движением скоростного трамвая и интегрированные подсистемы верхнего функционального уровня ЦСУДСТ;

O^{id} – сеть межведомственного электронного документооборота,

O^{usc} – сеть Городского центра управления внеуличным транспортом.

В составе модели также находится изолированная сеть обеспечения безопасности МБО[2, 3]. Она содержит отдельный интерфейс *GE* для управления маршрутами ядра через таблицы маршрутизации. Дискреционная модель сети сервера МБО имеет вид:

$$O_{GE}^{SecO} \cdot [(O_i^{ISC} \cdot O_j^{ISC})_{GEcoreISC} + \cup_i (O_i^{pr})_{GE} + \cup_i (O_i^{db})_{GE} + \\ + \cup_i (O_i^{MPLS})_{GErail} + \cup_i (O_i^{MPLS})_{GEmetro} + \cup_i (O_i^{MPLS})_{GEtram}],$$

где O_{GE}^{SecO} – объекты сервера МБО, приложений, баз данных, сети *MPLS* в изолированной сети обеспечения безопасности МБО, подключаемые по интерфейсу *GE*.

В сети ИСУ должен применяться отдельный физический управляемый (контролируемый) сетевой интерфейс для каждого внешнего телекоммуникационного сервиса.

В модели предусмотрен обмен с государственными органами и организациями (ОАО «РЖД», Департамент транспорта г. Москвы и др.) через систему межведомственного документооборота[4, 5].

Структура централизованной интеллектуальной системы управления городской рельсовой транспортной системой (ЦИСУ ГРТС) представлена на рисунке ниже (рис. 1). Перечень информационных ресурсов-приложений для решения задач управления:

- интеллектуальная система прогнозирования и анализа пассажиропотока ГРТС;
- интеллектуальная система прогнозирования, планирования и анализа работы ГРТС;
- интеллектуальная система прогнозирования, планирования и анализа работы операторов транспортного средства ГРТС;
- интеллектуальная система прогнозирования, планирования и анализа работы диспетчеров ГРТС;
- интеллектуальная система прогнозирования, планирования и анализа работ по техническому содержанию инфраструктуры ГРТС;
- интеллектуальная система прогнозирования, планирования и анализа работы транспортных средств ГРТС;
- интеллектуальная система управления движением и безопасностью транспортных средств ГРТС;
- бортовые устройства беспилотного управления транспортными средствами ГРТС;
- ситуационный центр.

---

[2] Приказ Федеральной службы по техническому и экспортному контролю Российской Федерации от 25 декабря 2017 г. № 239 «Об утверждении требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации (в ред. Приказов ФСТЭК России от 9 августа 2018 г. № 138, от 26 марта 2019 г. № 60, от 20 февраля 2020 г. № 35). [Электронный ресурс]: https://fstec.ru/tekhnicheskaya-zashchita-informatsii/obespechenie-bezopasnosti-kriticheskoj-informatsionnoj-infrastruktury/288-prikazy/1592-prikaz-fstek-rossii-ot-25-dekabrya-2017-g-n-239. Доступ 17.02.2021.

[3] ГОСТ Р ИСО 14813-1-2011. Национальный стандарт Российской Федерации «Интеллектуальные транспортные системы. Схема построения архитектуры интеллектуальных транспортных систем. Часть 1. Сервисные домены в области интеллектуальных транспортных систем, сервисные группы и сервисы». [Электронный ресурс]: https://docs.cntd.ru/document/1200086739. Доступ 17.02.2021.

[4] Методический документ «Меры защиты информации в государственных информационных системах» (утв. Федеральной службой по техническому и экспортному контролю 11 февраля 2014 г.). [Электронный ресурс]: https://www.garant.ru/products/ipo/prime/doc/70491518/. Доступ 17.02.2021.

[5] Приказ Федеральной службы по техническому и экспортному контролю Российской Федерации от 11 февраля 2013 года № 17 «Об утверждении Требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах» (с изменениями на 27 апреля 2020 года, редакция, действующая с 1 января 2021 года). [Электронный ресурс]: https://docs.cntd.ru/document/499002630. Доступ 17.02.2021.

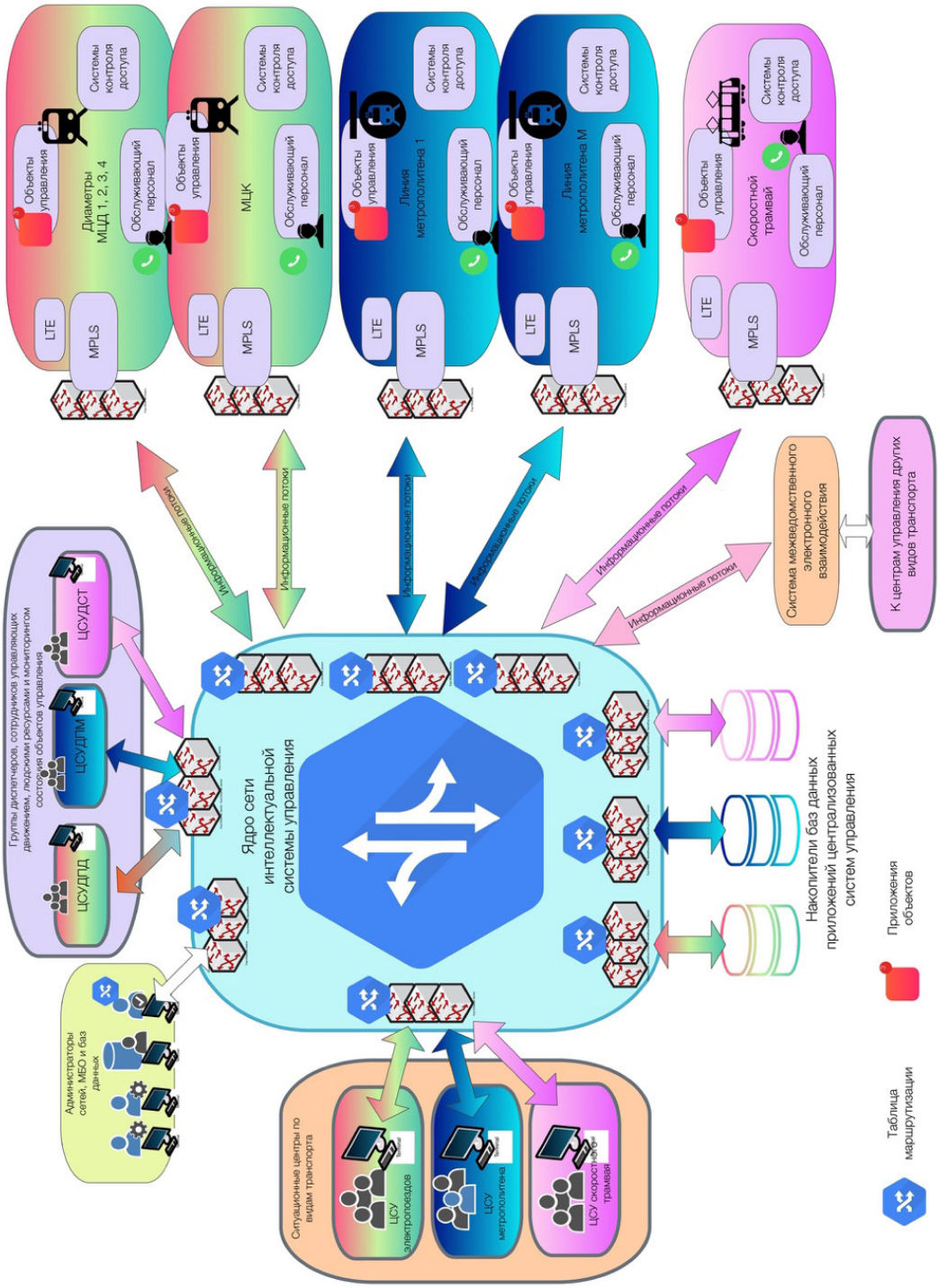


Рис. 1. Архитектура системы сбора и обработки большого объёма данных для внедрения ЦИСУ ГРТС (схема составлена авторами).

Техническая реализация сети ИСУ базируется на использовании оптических коммутаторов со встроенной системой управления посредством порта *GE* сети МБО. Ядро сети ИСУ физически располагается в стойке, где развёртывается полносвязная сеть. Связь между сетями *MPLS* централизованных систем управления движением по видам ГРТС осуществляется на основе установления связей между маршрутизаторами *PE* (снятие метки) сети *MPLS* и пересылки на маршрутизатор CE, соединённый с коммутаторами ядра по оптическим кабелям со скоростью обмена до 8 Гб/с.

Скорость обмена в ядре 16 Гб/с с периферией (персональные компьютеры диспетчеров, администраторов сетей и приложений, а также компьютеров ЦСУ) до 8–16 Гб/с. Хранилища баз данных сформированы по видам транспорта и типам задач управления, решаемых для конкретного вида транспорта. Скорость обмена между объектами ядра сети и накопителями – 16 Гб/с. Объём хранилища составляет: пригородный железнодорожный транспорт в черте городов – до 20 петабайт (Пб), метрополитен – до 20 Пб, скоростной трамвай – до 3 Пб.

По требованиям безопасности выход в сеть Интернет запрещён для критически важных объектов, к которым относится транспорт города[6, 7, 8].

Ядро ЦИСУ ГРТС объединяет в единый хаб следующие центры: Городской центр управления внеуличным транспортом, связанный с центрами ситуационного управления по видам городского рельсового транспорта: пригородного железнодорожного транспорта в черте городов, метрополитена, скоростного трамвая, а также интегрированные подсистемы верхнего функционального уровня централизованного управления движением транспортных средств пригородного железнодорожного транспорта в черте городов, метрополитенов и скоростного трамвая.

Ядро ИСУ предоставляет для каждого центра необходимые ресурсы: диспетчерские пулы по видам городского транспорта отдельно для каждой линии метрополитена, участка пригородного железнодорожного транспорта в черте городов (диаметра или кольца) и участка движения скоростного трамвая; базы данных; серверы; выход в корпоративные сети *MPLS*, развёрнутые на линиях метрополитена, участках пригородного железнодорожного транспорта в черте городов (диаметра или кольца) и участках движения скоростного трамвая.

Диспетчерские пулы оснащены персональными компьютерами, видеостенами, технологической связью с обслуживающим персоналом на линиях и участках.

К ядру ЦИСУ ГРТС подключены администраторы баз данных, приложений систем управления верхнего уровня по видам транспорта и администраторы сети *MPLS* и сотовой связи 4G (*LTE*), а также администраторы безопасности. Ядро ЦИСУ ГРТС обеспечивает организацию изолированных информационных потоков большого объёма между серверами приложений систем управления за счёт применения оптических интерфейсов и реализации полносвязной сети, направленных к объектам нижнего уровня систем управления.

Рассмотрим работу сети при управлении движением электропоездов пригородного железнодорожного транспорта в черте городов. Базовыми устройствами, обеспечивающими безопасное движение, являются системы МПЦ, управляющие стрелками, сигналами, рельсовыми цепями, переездами и другими устройствами в пределах станции, системы автоблокировки (АБ), обеспечивающие контроль

---

[6] Методический документ «Меры защиты информации в государственных информационных системах» (утв. Федеральной службой по техническому и экспортному контролю 11 февраля 2014 г.). [Электронный ресурс]: https://www.garant.ru/products/ipo/prime/doc/70491518/. Доступ 17.02.2021.

[7] Приказ Федеральной службы по техническому и экспортному контролю Российской Федерации от 25 декабря 2017 г. № 239 «Об утверждении требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации (в ред. Приказов ФСТЭК России от 9 августа 2018 г. № 138, от 26 марта 2019 г. № 60, от 20 февраля 2020 г. № 35). [Электронный ресурс]: https://fstec.ru/tekhnicheskaya-zashchita-informatsii/obespechenie-bezopasnosti-kriticheskoj-informatsionnoj-infrastruktury/288-prikazy/1592-prikaz-fstek-rossii-ot-25-dekabrya-2017-g-n-239. Доступ 17.02.2021.

[8] ГОСТ Р ИСО 14813-1-2011. Национальный стандарт Российской Федерации «Интеллектуальные транспортные системы. Схема построения архитектуры интеллектуальных транспортных систем. Часть 1. Сервисные домены в области интеллектуальных транспортных систем, сервисные группы и сервисы». [Электронный ресурс]: https://docs.cntd.ru/document/1200086739. Доступ 17.02.2021.

состояния рельсовой цепи в контрольном режиме, системы энергоснабжения, а также системы контроля инфраструктуры (путь, полотно, рельсы). Система управления высокоскоростным составом агрегирует информацию с этих систем на серверах радиоблокцентров с целью передачи на бортовые системы управления на подвижном составе (с использованием защищённого протокола) по беспроводной системе связи 4G (*LTE*), где происходит расчёт предельных значений скорости для обеспечения тормозного пути.

С целью обеспечения требований безопасности системы МПЦ, диспетчерской централизации (ДЦ), АБ осуществляют управление объектами с использованием собственных серверов по специальным протоколам. Для обеспечения режима безопасного обмена информацией между серверами радиоблокцентра в ЦСУДЭП создана буферная база данных *MySQL*. Информация из базы данных поступает по сети *MPLS* в ядро сети ЦИСУ ГРТС на сервера верхнего функционального уровня ЦСУДЭП.

Подсистемы верхнего функционального уровня ЦСУДЭП используют данные, поступающие из системы контроля инфраструктуры через сеть *MPLS*. Системы контроля инфраструктуры осуществляют сбор информации с элементов пути для определения ограничений скорости, обусловленных состоянием верхнего строения пути, ремонтными работами, отсутствием незаконного проникновения предметов, животных, людей. С целью обеспечения скоростной доставки информации управления от подсистемы верхнего функционального уровня ЦСУДЭП до бортовых систем управления на подвижном составе в ядре ЦИСУ ГРТС формируется маршрут между сервером управления движением подвижного состава и контроллером на подвижном составе. Для пакетов управления устанавливается приоритет не ниже 5 (*QoS*). Время задержки пакетов не более 20 мкс.

Верхний функциональный уровень ЦСУДПМ строится для каждой линии метрополитена. Сеть ЦИСУ ГРТС управления выполняет следующие функции: связывает серверы верхнего функционального уровня ЦСУДПМ каждой линии, с одной стороны, и бортовые системы управления на подвижном составе (нижний уровень ЦСУДПМ), системы контроля доступа на инфраструктуру (контроль появления посторонних предметов, людей и других препятствий: оползней, грунтовых вод и т.д.), системы МЧС и пожарной тревоги, системы диагностики технического состояния устройств автоматики и телемеханики (системы автоматической локомотивной сигнализации с автоматическим регулированием скорости (АЛС-АРС) и ДЦ, предотвращающих превышение скоростей и обеспечивающих остановку состава в экстренных случаях), подвижного состава, систем энергоснабжения с другой стороны.

Для реализации верхнего функционального уровня ЦСУДПМ на линиях метрополитена развёрнута мультисервисная сеть *MPLS*, задачей которой является интеграция беспроводной сети передачи информации 4G (*LTE*) для связи бортовых систем управления на подвижном составе и технологических систем связи для обслуживающего персонала, интеграция по оптическим каналам с системами контроля доступа, системами обеспечения безопасности движения АЛС-АРС, ДЦ, системами технической диагностики устройств энергоснабжения, автоматики и телемеханики, расположенными на линии метрополитена.

Для обеспечения режима приоритетной доставки информации от приложений серверов верхнего функционального уровня ЦСУДПМ реализован режим изолированного маршрута (за счёт полносвязности объектов сети) в ядре сети ИСУ со скоростью 16 Мб/с (по оптическому кабелю) и режим *QoS* в *MPLS* с приоритетом не ниже 5, что обеспечивает доставку информации от серверов управления до бортовых систем управления на подвижном составе с задержкой не более 20 мкс.

Алгоритмы ведения подвижного состава метрополитена и их параметры определяются на верхнем функциональном уровне ЦСУДПМ с учётом текущего технического состояния инфраструктуры и подвижного состава, а также ограничений, формируемых системами обеспечения безопасности движения.

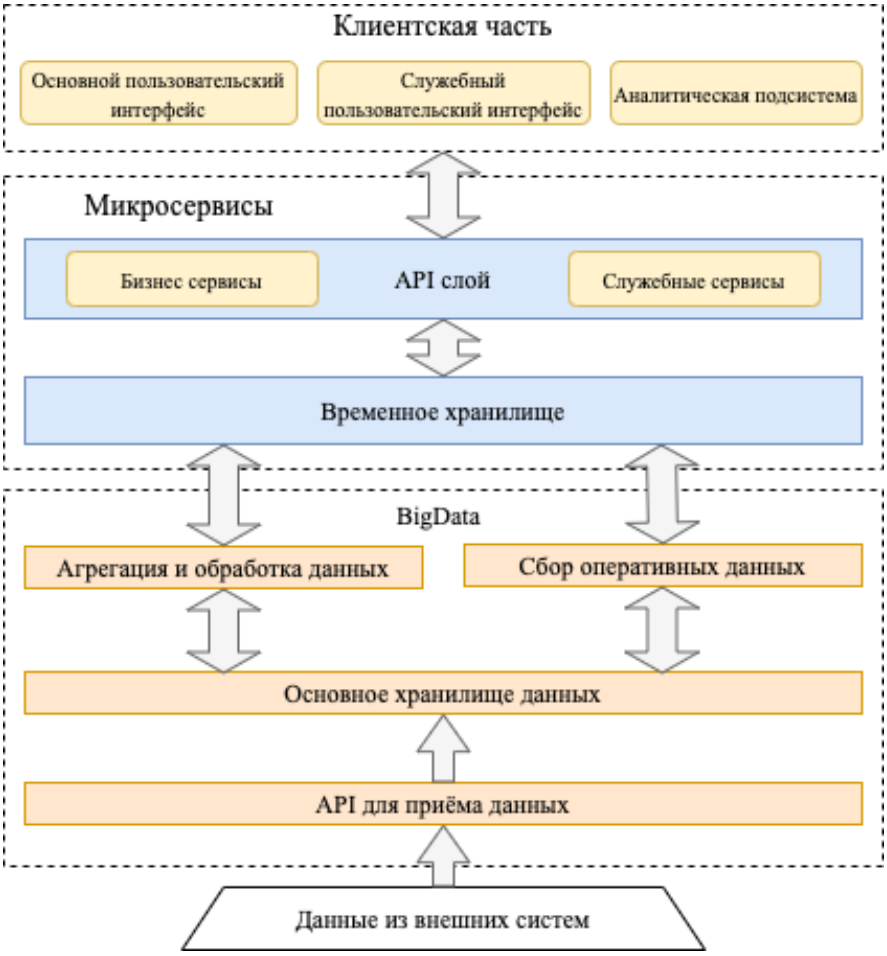


Рис. 2. Архитектура системы сбора и обработки большого объёма данных для внедрения ЦИСУ ГРТС (схема составлена авторами).

Рассмотрим работу сети при управлении движением скоростного трамвая. Связь между бортовыми системами управления на скоростном трамвае и верхним функциональным уровнем ЦСУДСТ осуществляется по сети *LTE* через корпоративную сеть *MPLS*. Местоположение трамвая определяется на основе высокоточной координатной сети и цифровой модели пути, обеспечивающих высокую точность позиционирования скоростного трамвая. Оценка состояния пути проводится с использованием видеокамер, установленных вдоль линий движения. На основании полученной информации формируются ограничения скорости движения или сигнал остановки состава. Требуемая скорость доставки информации от верхнего функционального уровня ЦСУДСТ обеспечивается установкой приоритетов пакетам управления и изолированным маршрутам в ядре сети ЦИСУ ГРТС.

Структура уровня приложений ИСУ

Рассмотрим особенности реализации приложений, управления ими и их взаимодействия с базами данных.

При создании ЦИСУ ГРТС требуется учитывать ряд узких мест с точки зрения хранения автоматизации и взаимодействия с внешними системами:

• хранение и обработка информации, которая необходима для функционирования ЦИСУ ГРТС. Эта информация собирается в большом числе сторонних автоматизированных систем. Имеющиеся автоматизированные системы используют различные технологии, интерфейсы, архитектуру;

• индивидуальная реализация информационного обмена между автоматизированными системами управления (АСУ) сторонних разработчиков;

• достоверность данных, хранящихся в различных АСУ, не всегда обеспечена соответствующими технологиями и процедурами логического контроля данных;

• имеющиеся АСУ обеспечивают только оперативное хранение данных, длительное накопление информации в течение всего жизненного цикла объектов управления не выполняется, что затрудняет или делает невозможным эффективное применение инструментов предиктивной аналитики, машинного обучения и т.д.

Для устранения этих ограничений и создания ЦИСУ ГРТС должны быть решены следующие задачи:

• определение спектра цифровых технологий, позволяющих аккумулировать и анализировать значительные массивы информации от транспортных средств и объектов инфраструктуры в целях обеспечения гибкого подхода к технической эксплуатации ГРТС, технологического оборудования и инженерных сооружений;

• разработка схем данных, обеспечение их хранения в структуре распределённого хранилища данных ЦИСУ ГРТС;

• создание распределённого хранилища данных ЦИСУ ГРТС, использующего технологии больших данных и распределённого реестра, предназначенного для совместного использования ЦИСУ ГРТС и внешними поставщиками данных;

• создание механизма ввода данных в распределённое хранилище на основе унифицированного *API*-интерфейса по инициативе смежных АСУ по факту появления в них данных, необходимых ЦИСУ ГРТС;

• создание возможности ввода данных, поступающих от бортовых систем в режиме реального времени;

• создание функциональности:

– логического контроля данных, поступающих в ЦИСУ ГРТС;

– управления конфликтами и ошибками, выявляемыми при поступлении данных;

– визуализации данных, хранящихся в распределённом хранилище;

– обеспечения возможности раннего выявления рисков в управлении поддержанием технической готовности человеческих и технических ресурсов ГРТС за счёт внедрения современных аналитических методов статистической обработки и машинного обучения;

– реализации информационных сервисов в интересах потребителей информации о ГРТС.

Архитектура по ЦИСУ ГРТС

Авторами разработана архитектура и выставлены требования к построению гетерогенной системы сбора и обработки большого массива информации, поступающей из различных источников с учётом определённых задач.

В качестве базиса при построении системы для работы с большими данными было взято две архитектуры: *lambda* и микросервисная (МС).

Архитектура стоит из трёх основных блоков (рис. 2):

• клиентская часть – сервисы, используемые внешними клиентами;

• микросервисы – сервисы, которые обрабатывают запросы от клиента и взаимодействуют с временным хранилищем;

• *BigData* – основное хранилище с данными большого объёма, требующими значительных ресурсов на использование и обслуживание [14–18].

В клиентской части системы реализованы способы взаимодействия через основной и служебный пользовательский интерфейс. Основной пользовательский интерфейс включает отчётные формы для различных бизнес-направлений системы. Служебный пользовательский интерфейс включает в себя подсистемы мониторинга общего состояния системы и осуществляет прямой доступ через *web*-интерфейс к данным для решения интеллектуальных задач.

Микросервисная часть [19] состоит из слоёв, каждый из которых имеет свой логический уровень:

• *API* слой работает напрямую с запросами, поступающими из клиентской части, и запрашивает необходимые данные из временного хранилища. *API* слой включает сервисные и бизнес модули.

• временное хранилище – хранилище данных с результатами расчётов, оперативными данными и данными о пользователях.

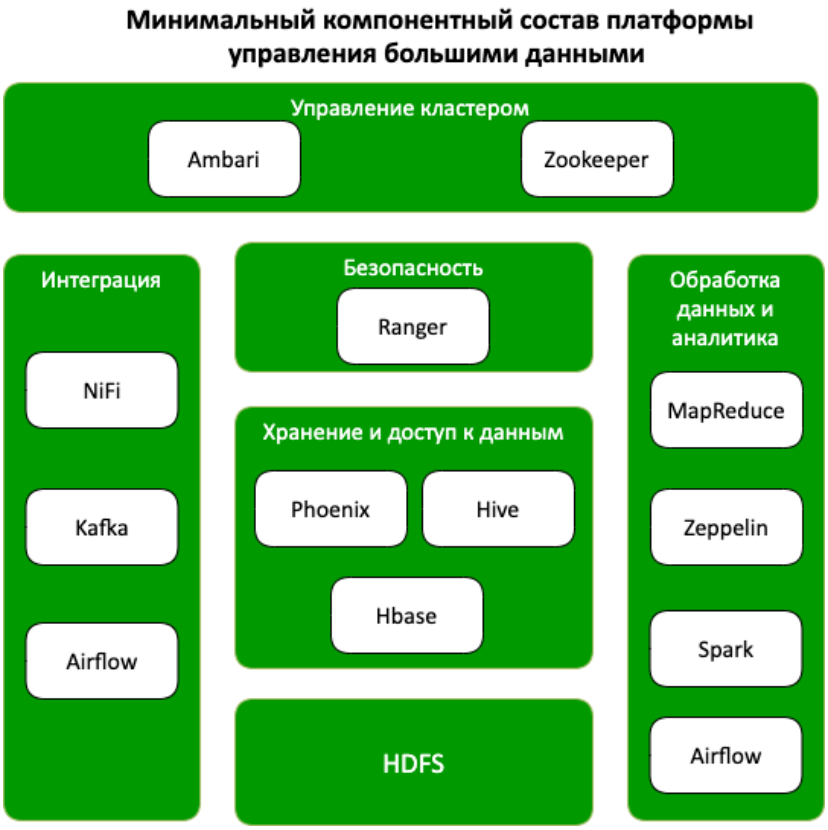


Рис. 3. Программное содержание ЦИСУ ГРТС (схема составлена авторами).

Используемые технологии: *MongoDB* [20] и *CouchDB* [21].

BigData часть состоит из слоёв, каждый из которых имеет свой логический уровень:

• агрегация и обработка данных – сервис выполняет расчёт признаков и прогнозов на основе данных, поступивших в основное хранилище;

• сбор оперативных данных – *API*-сервис для передачи оперативных данных во временное хранилище;

• основное хранилище данных – распределённое файловое хранилище, построенное с использованием технологий *HBase* [22] и *HDFS* [23];

• *API* для приёма данных – сервис приёма данных из внешних систем, построенный с использованием технологии *NiFi* [24].

Программная среда может включать в себя (рис. 3):

• *Unix* подобную операционную систему;

• *Ambari* – развёртывание и администрирование кластера *Hadoop* [25];

• *Zookeeper* – сервис, обеспечивающий распределённую синхронизацию конфигурационной информации пространства имён для группы используемых приложений [26];

• *Ranger* – модуль, обеспечивающий центральное управление контролем доступа к файлам, папкам, базам данных, таблицам [27];

• *NiFi* – унифицированный интерфейс сбора информации из внешних систем [24];

• *Kafka* – менеджер очередей [28];

• *Airflow* – библиотека для планирования и мониторинга рабочих процессов [29];

• *Phoenix* – декларативный *SQL* синтаксис, применяемый к базе данных *HBase* [30];

• *HBase*, *Hive* – распределённое хранение и синхронизация больших данных [25];

• *MapReduce* – технология распределённых вычислений, используемая для параллельных вычислений над большими (до нескольких петабайт) наборами данных в компьютерных кластерах [31];

• *Zeppelin* – визуализация и аналитика больших данных [32];

• *Spark* – фреймворк для распределённой пакетной и потоковой обработки неструктурированных и слабоструктурированных данных [33];

• *HDFS* – распределённое хранение файлов больших данных [25].

Система должна обеспечивать возможность хранения данных ГРТС в течение всего жизненного цикла как самого объекта, так и связанных объектов. Данные не удаляются и накапливаются, принудительная чистка не предусмотрена. Допускается возможность архивирования данных.

Скорость отклика ЦИСУ ГРТС должна составлять:

• для операций навигации по экранным формам системы – не более 5 с;

• для операций формирования отчётов – не более 5 минут.

Система должна обеспечивать отражение следующих изменений:

• добавление новых предприятий – пользователей (функциональных заказчиков) ЦИСУ ГРТС;

• изменение наборов данных, поступающих от ГРТС.

Система должна иметь открытые интерфейсы для развития (модернизации) и интеграции.

Система должна обеспечить возможность поэтапного наращивания своих функциональных возможностей, в том числе:

• замену аппаратного и системного программного обеспечения при условии соответствия требованиям, определённым в документации на типовые и поставляемые программные средства;

• расширение функциональности системы в части разработки новых модулей и подсистем, а также интеграции с разрабатываемыми автоматизированными и информационными системами.

Требования к режимам функционирования системы определены для основного (штатного) режима функционирования, в котором исправно работает оборудование, составляющее комплекс технических средств, и исправно функционирует системное, базовое и прикладное ПО; режим ограниченной функциональности (аварийный режим); режим проведения регламентных работ.

Режим ограниченной функциональности характеризуется отказом одного или нескольких компонентов программного и (или) аппаратного обеспечения. При этом ограниченная работоспособность системы по выполнению функционального назначения сохраняется. Это может быть вызвано нарушением связи или программно-аппаратными сбоями, компенсация которых средствами резервирования невозможна.

Режим проведения регламентных работ предназначен для проведения восстановительных и регламентных работ и модернизации ПО. В этом режиме допускается временная (плановая) недоступность продуктивной системы для пользователей. Регламент перехода из одного режима функционирования в другой, а также инструкции для персонала системы по работе в каждом из указанных выше режимов и по действиям в процессе перехода от одного режима к другому должны быть описаны в эксплуатационной документации.

Требования по диагностированию системы заключаются в том, что должен быть инструментарий диагностирования работоспособности основных процессов системы и выявления событий, важных с точки зрения функционирования информационной системы. Данный инструментарий должен обеспечить необходимый мониторинг функционирования информационной системы путём ведения специализированных журналов (лог-файлов).

Комплексный мониторинг функционирования информационной системы осуществляется комбинацией следующих способов:

• стандартный мониторинг информационной системы – мониторинг аппаратных ресурсов, мониторинг работоспособности процессов и сервисов;

• мониторинг доступности *URL*-ссылок;

• мониторинг записей в лог-файлах информационной системы, имеющих стандартизированную и документированную структуру;

• мониторинг содержимого служебных таблиц БД информационных систем;

• мониторинг сообщений, получаемых от информационной системы по протоко-

лу *SNMP* на серверы мониторинга, с *IP*-адресом, задаваемым параметрически;

• другие способы диагностики, согласованные с сопровождающим персоналом и администраторами системы мониторинга.

При возникновении аварийных ситуаций или ошибок в ПО диагностические инструменты должны позволять сохранять полный набор информации, необходимой для идентификации проблемы.

Должна быть предусмотрена функция выдачи отчётов о работе системы.

ВЫВОДЫ

В результате проведённого исследования предложено использовать организацию защищённых информационных потоков для ЦИСУ ГРТС на основе меток и реализации полносвязного ядра сети системы. Полносвязность ядра позволяет увеличить скорость информационного обмена, разделять потоки между приложениями ИСУ по типам протоколов и создавать временные доверенные информационные потоки с минимальным набором признаков, затрудняя организацию тайных каналов передачи информации.

Реализация сети ЦИСУ ГРТС с использованием полносвязного ядра и оптических каналов позволяет связать сеть *MPLS* подсистем управления верхнего уровня с ядром ЦИСУ ГРТС локальным полносвязным сегментом, что повышает безопасность и объёмы передаваемой информации от периферийного оборудования нижнего уровня.

Предложенная авторами архитектура ПО позволит организовать эффективную работу ЦИСУ ГРТС с точки зрения процессов хранения, получения и обработки данных. В том числе, учитывая тот факт, что система на программном уровне включает в себя микросервисную архитектуру, позволяет обеспечить простоту сопровождения, разработки и развёртывания. Микросервисная архитектура доказала свою состоятельность при разработке комплексной модели линии метрополитена и создании на её базе широкого спектра средств управления, планирования, оптимизации и обучения [34–36], а также при создании системы «Доверенная среда локомотивного комплекса» [37].

ЛИТЕРАТУРА

1. Вакуленко С. П., Роменский Д. Ю., Мнацаканов В. А., Дорохов А. В., Власов Д. Н. Разработка вариантов модернизации Московской монорельсовой транспортной системы // Метро и тоннели. – 2020. – № 4. – С. 28–36. [Электронный ресурс]: https://hutor.info/wp-content/uploads/2020/12/2_533227896687716185 3.pdf. Доступ 14.02.2021.

2. Shevlyugin, M.V., Korolev, A. A., Golitsyna, A. E., Pletnev, D. S. Electric stock digital twin in a subway traction power system. Russian Electrical Engineering, 2019, Vol. 90, Iss. 9, pp. 647–652. DOI: 10.3103/S1068371219090098. Доступ 14.02.2021.

3. Павловский А. А., Охотников А. Л. Информационная транспортная ситуация // Наука и технологии железных дорог. – 2018. – Т. 2. – № 6. – С. 16–24. [Электронный ресурс]: http://www.vniias.ru/images/img/online_journal/pdf/02_2018/2_2018.pdf. Доступ 14.02.2021.

4. Атрощенко В. А., Руденко М. В., Дьяченко Р. А., Багдасарян Р. Х. К вопросу оценки достоверности информации для предотвращения mitm-атаки при передаче закрытой информации по открытым каналам связи // Современные проблемы науки и образования. – 2013. – № 3. – С. 82–82. [Электронный ресурс]: https://science-education.ru/pdf/2013/3/375.pdf. Доступ 14.02.2021.

5. Sudhir, Udipi. The event data management problem: getting the most from network detection and response. Network Security, January 2021, Vol. 2021, Iss. 1, pp. 12–14. DOI: https://doi.org/10.1016/S1353-4858(21)00008-8.

6. Halpin T., Morgan T. Information modelling and relational databases (Second Edition). The Morgan Kaufmann Series in Data Management Systems 2008, 976 p. DOI: 10.1016/B978-0-12-373568-3.X5001-2.

7. Yue, Zeng; Ye, Baoliu; Tang, Bin; Guo, Songtao; Qu, Zhihao. Scheduling coflows of multi-stage jobs under network resource constraints. Computer Networks, January 12, 2021, Vol. 184, pp. 107686. DOI: 10.1016/j.comnet.2020.107686.

8. Уэнстром М. Организация защиты сетей Cisco. – М.: Издательский дом «Вильямс», 2005. – 768 с. ISBN 5-8459-0387-4.

9. Харитонова Е. В. Графы и сети. – Ульяновск: УлГТУ, 2006. – 92 с. [Электронный ресурс]: https://www.studmed.ru/haritonova-ev-grafy-i-seti_9d47b8a399b.html. Доступ 14.02.2021.

10. Кузюков В. А., Новиков В. Г., Сафронов А. И. Микропроцессорные системы управления движением поездов в Московском метрополитене // Автоматика на транспорте. – 2020. – Т. 6. – № 3. – С. 268–293. [Электронный ресурс]: https://cyberleninka.ru/article/n/mikroprotsessornye-sistemy-upravleniya-dvizheniem-poezdov-v-moskovskom-metropolitene/pdf. Доступ 14.02.2021.

11. Девянин П. Н. Модели безопасности компьютерных систем. – М.: Горячая линия-Телеком. – 2018. – 338 с. ISBN 5-7695-2053-1.

12. Zhang, Shunliang; Zhu, Dali. Towards artificial intelligence enabled 6G: State of the art, challenges, and opportunities. Computer Networks, December 24, Vol. 183, pp. 107556. DOI: 10.1016/j.comnet.2020.107556.

13. Mei, Lifan; Hu, Runchen; Cao, Houwei; Liu, Yong; Han, Zifa; Li, Feng; Li, Jin. Realtime Mobile Bandwidth Prediction Using LSTM Neural Network. In: Choffnes D., Barcellos M. (eds) Passive and Active Measurement. PAM 2019. Lecture Notes in Computer Science, vol 11419. Springer, Cham. https://doi.org/10.1007/978-3-030-15986-3_3.

14. Kalipe, G. K., Behera, R. K. Big Data Architectures: A detailed and application oriented review. Int. Journal Innov. Technol. Explor. Eng., 2019, Vol. 8, pp. 2182–2190. [Электронный ресурс]: https://www.researchgate.net/profile/Rajat-Behera/publication/336915402_Big_Data_Architectures_A_detailed_and_application_oriented_review/links/5dba7a2e4585151435d62a79/Big-Data-Architectures-

A-detailed-and-application-oriented-review.pdf. Доступ 14.02.2021.

15. Thurner, T. Big Data Europe for Smart, Green and Integrated Transport. [Электронный ресурс]: https://www.w3.org/community/bde-transport/files/2015/11/Big-Data-for-Smart-Green-and-Integrated-Transport-Workshop-Final-Report.pdf. Доступ 14.02.2021.

16. Авдеева И. Л. Анализ зарубежного опыта использования глобальных технологий «Big Data» // Интернет-журнал «Науковедение». – 2016. – Т. 8. – № 6. – С. 1–11 [Электронный ресурс]: http://naukovedenie.ru/PDF/13EVN616.pdf. Доступ 14.02.2021.

17. Бик Р. Применение Big Data в транспортном планировании. [Электронный ресурс]: https://transport.mos.ru/common/upload/docs/1500293313_Moovit_Moscow_International_Transport_Expert_Council_R.pdf. Доступ 14.02.2021.

18. Технология Big Data на транспорте. Как на транспорте большие данные превратились в ценный актив. Отечественная СУБД Tarantool в проекте аналитики больших данных. [Электронный ресурс]: https://tygeza.ru/tehnologiya-big-data-na-transporte-kak-na-transporte-bolshie-dannye.html. Доступ 14.02.2021.

19. O'Connor, R. V., Elger, P., Clarke, P. M. Continuous software engineering – A microservices architecture perspective. Journal of Software: Evolution and Process, 2017, Vol. 29, Iss. 11, pp. e1866. [Электронный ресурс]: https://www.researchgate.net/profile/Rory-Oconnor-4/publication/316009873_Continuous_software_engineering-A_microservices_architecture_perspective/links/5a26bc4e4585155dd423eecc/Continuous-software-engineering-A-microservices-architecture-perspective.pdf. Доступ 14.02.2021. DOI: 10.1002/smr.1866.

20. Boicea, A., Radulescu, F., Agapin, L. I. MongoDB vs Oracle-database comparison. 2012 3rd International Conference on Emerging Intelligent Data and Web Technologies. IEEE, 2012, pp. 330–335. [Электронный ресурс]: https://www.researchgate.net/profile/Alexandru-Boicea/publication/261040647_MongoDB_vs_Oracle_—_Database_Comparison/links/55c2132b08aebc967defd053/MongoDB-vs-Oracle-Database-Comparison.pdf. Доступ 14.02.2021. DOI: 10.1109/EIDWT.2012.32.

21. CouchDB. Apache CouchDB. [Электронный ресурс]: https://couchdb. apache.org. Доступ 14.02.2021.

22. Vora, M. N. Hadoop-HBase for large-scale data. Proceedings of 2011 International Conference on Computer Science and Network Technology. IEEE, 2011, Vol. 1, pp. 601–605. DOI: 10.1109/ICCSNT.2011.6182030.

23. Shvachko, K., Kuang, H., Radia, S., Chansler, R. The Hadoop Distributed File System. 2010 IEEE 26th symposium on mass storage systems and technologies (MSST). IEEE, 2010, pp. 1–10. DOI: https://doi.org/10.1109/MSST.2010.5496972.

24. Kim, S.-S., Lee, W.-R., Go, J.-H. A Study on Utilization of Spatial Information in Heterogeneous System Based on Apache NiFi. 2019 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2019, pp. 1117–1119. DOI: 10.1109/ICTC46691.2019.8939734.

25. Venner, J., Wadkar, Sameer; Siddalingaiah, Madhu. Pro Apache Hadoop. Apress, Berkeley, CA, 2014, pp. 399–401. DOI: 10.1007/978-1-4302-4864-4_9.

26. Hunt, P., Konar, M., Junqueira, F. P., Reed, B. ZooKeeper: Wait-free Coordination for Internet-scale Systems. USENIX annual technical conference, 2010, Vol. 8, Iss. 9, pp. 1–14. [Электронный ресурс]: https://www.usenix.org/legacy/events/atc10/tech/full_papers/Hunt.pdf. Доступ 14.02.2021.

27. Gupta, M., Patwa, F., Sandhu, R. An Attribute-Based Access Control Model for Secure Big Data Processing in Hadoop Ecosystem. Proceedings of the Third ACM Workshop on Attribute-Based Access Control, 2018, pp. 13–24. [Электронный ресурс]: https://www.researchgate.net/profile/Maanak-Gupta/publication/323785048_An_Attribute-Based_Access_Control_Model_for_Secure_Big_Data_Processing_in_Hadoop_Ecosystem/links/5ab0253b0f7e9b4897c1d52b/An-Attribute-Based-Access-Control-Model-for-Secure-Big-Data-Processing-in-Hadoop-Ecosystem.pdf. Доступ 14.02.2021. DOI: 10.1145/3180457.3180463.

28. Wang, Guozhang; Koshy, Joel; Subramanian, Sriram; Paramasivam, Kartik; Zadeh, Mammad; Narkhede, Neha; Rao, Jun; Kreps, Jay; Stein, Joe. Building a replicated logging system with Apache Kafka. Proceedings of the VLDB Endowment, 2015, Vol. 8, Iss 12, pp. 1654–1655. DOI: 10.14778/2824032.2824063.

29. Singh, P. Airflow. Learn PySpark. Apress, Berkeley, CA, 2019, pp. 67–84. [Электронный ресурс]: https://link.springer.com/content/pdf/10.1007%2F978-1-4842-4961-1.pdf. Доступ 14.02.2021.

30. Akhtar, S., Magham, R. Using Phoenix. Pro Apache Phoenix. Apress, Berkeley, CA, 2017, pp. 15–35. [Электронный ресурс]: https://link.springer.com/content/pdf/10.1007%2F978-1-4842-2370-3.pdf. Доступ 14.02.2021.

31. Condie, T., Conway, N., Alvaro, P., Hellerstein, J., Elmeleegy, K., Sears, R. MapReduce Online. Conference: Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2010, San Jose, CA, USA, April 28–30, 2010, Vol. 10, Iss. 4, pp. 313–328. [Электронный ресурс]: https://www.usenix.org/legacy/events/nsdi10/tech/full_papers/condie.pdf. Доступ 14.02.2021.

32. Cheng, Yanzhe; Liu, Fang; Jing, Shan; Xu, Weijia; Chau, Duen Horng. Building Big Data Processing and Visualization Pipeline through Apache Zeppelin. PEARC'18: Proceedings of the Practice and Experience on Advanced Research Computing, 2018, pp. 1–7. DOI: 10.1145/3219104.3229288.

33. Zaharia, M. [et al.] «Apache spark: a unified engine for big data processing.» Communications of the ACM 59.11 (2016): 56–65. DOI: 10.1145/2934664.

34. Баранов Л. А. Балакина Е. П., Ерофеев Е. В., Сидоренко В. Г. Многофункциональные модели систем управления // Известия высших учебных заведений. Проблемы полиграфии и издательского дела. – 2012. – № 2. – С. 79–82. [Электронный ресурс]: https://publications.hse.ru/mirror/pubs/share/folder/54otctuizr/direct/118351222.pdf. Доступ 14.02.2021.

35. Сидоренко В. Г., Чжо М. А. Исследование возможности применения генетических алгоритмов к решению задач планирования работы электроподвижного состава метрополитена // Электроника и электрооборудование транспорта. – 2017. – № 6. – С. 37–40. [Электронный ресурс]: https://publications.hse.ru/mirror/pubs/share/direct/213900236.pdf. Доступ 14.02.2021.

36. Кульба В. В., Ковалевский С. С., Косяченко С. А. Кузнецов Н. А. Методы анализа и синтеза модульных информационно-управляющих систем. – М.: Физматлит, 2002. – 800 с. [Электронный ресурс]: http://bookfi.net/book/1471957. Доступ 14.02.2021.

37. Харин О. В., Якимов С. М., Кулагин М. А., Гоник М. М., Хлудеев М. А., Ярощук Д. И. Автоматизированная система доверенная среда локомотивного комплекса (2019). Свидетельство о регистрации программы для ЭВМ RU 2020613754, 23.03.2020. [Электронный ресурс]: https://www.elibrary.ru/item.asp?id=42709956. Доступ 14.02.2021.

Благодарности. Исследование выполнено при финансовой поддержке РФФИ, НТУ «Сириус», ОАО «РЖД» и Образовательного Фонда «Талант и успех» в рамках научного проекта № 20-37-51001.