

УДК 621.391+004.031

# Модифицированный алгоритм и информационное обеспечение для контроля параметров мультиагентных сетей

Разработана система мониторинга параметров с датчиков группы БПЛА. Проанализированы существующие системы мониторинга, разработаны алгоритм модели управления группой БПЛА, веб-приложение для мониторинга и контроля полета группы БПЛА. Разработанная программная система имеет возможность модернизации или изменения по желанию и требованию заказчика. Для разных целей могут использоваться различные наборы датчиков, механизмов контроля и мониторинга полета.

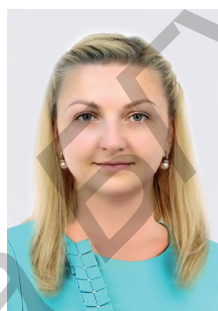
**Введение.** Постоянное развитие человечества требует создания систем мониторинга его деятельности, который необходим для сбора всесторонней информации и ее обработки. Постоянное использование мониторинга обеспечивает своевременное выявление ошибок. С развитием современных беспроводных сетей связи часто возникают ситуации, когда необходимо контролировать состояние системы, к которой отсутствует локальный доступ из-за ее территориальной удаленности или физических ограничений безопасности, что требует создания систем удаленного мониторинга.

Известны системы активного и пассивного мониторинга [1]. Пассивный мониторинг



**С.И. ПОЛОВНЯ,**  
заведующий кафедрой ТКС,  
канд. техн. наук, доцент

УО «Белорусская государственная академия связи»



**Ю.А. ДУЙНОВА,**  
преподаватель  
кафедры ТКС



**Н.М. БОБРИК,**  
студент, УО «Белорусский  
государственный  
университет информатики  
и радиоэлектроники»

предполагает получение данных в режиме чтения, например, системы сбора данных о температуре, о загрузке процессора, о потреблении оперативной памяти. Активный мониторинг включает элементы воздействия на среду (приложения, аппаратное обеспечение). Например, система, которая выполняет корректирующее действие при определенных внешних условиях. Системы мониторинга строятся по классической архитектуре «клиент–сервер». Взаимодействие клиента и сервера выполняется с помощью известных протоколов, а данные передаются через сети передачи данных.

Для выполнения мониторинга сервер использует и модифицирует текущую конфигурацию. Также он проводит сканирование системы, оповещает, если произошли сбои, сохраняет в конфигурационном файле результаты сканирования для последующего вывода их в графическом виде. Сам сервер не способен графически отображать параметры сети. Для получения графического изображения, а также некоторых видов оповещения используется клиент.

Клиент хранит информацию только в оперативной памяти. Основное предназначение клиента – правильно отобразить состояние системы и в случае сбоев или других событий показать это в клиентском приложении для удобного восприятия. Также клиент отвечает за создание графического интерфейса пользователя, предназначенного для конфигурирования сервера.

Тип мониторинга определяется областью его использования.

Системы пассивного мониторинга относятся к классу систем, которые используются для контроля возникновения нештатных ситуаций. После обработки информации с элементов системы возможна последовательность действий, среди которых – отображение полученной информации, принятие определенных решений для устранения возникшей ситуации. Способ представления данных может быть разным: построение графиков, генерация сообщений и др. Примерами таких систем являются MRTG (Multi Router Traffic Grapher) и Nagios. Автор MRTG создал его для контроля загруженности интерфейсов на сетевых устройствах. Однако данный тип системы мониторинга не позволяет в режиме реального времени отслеживать какие-либо показатели, но позволяет хранить статистику и отображать ее в виде графиков. Интерфейс Nagios ориентирован на наличие оператора, который следит за состоянием показателей системы. Время поступления данных регулируется временем опроса элементов. Удаленные системы опрашиваются посредством программных интерфейсов, и при наступлении каких-либо событий в интерфейсе выводятся соответствующие информационные сообщения.

Активный мониторинг характеризуется набором заранее заданных действий, как реакция на определенные события. Таким образом, для него характерно наличие обратной связи. Примерами таких систем являются HP OpenView и IBM Tivoli. Это интеллектуальные системы, в зависимости от возникновения событий генерирующие ответные действия для восстановления требуемых показателей. По всем параметрам к ним можно отнести систему «Умный дом», которая активно производит мониторинг ситуации и может совершать по заданному сценарию необходимые действия.

Также важным критерием оценки качества системы мониторинга следует считать степень ее защищенности от следующих критических факторов:

- отказ отдельных элементов или системы в целом;

- умышленное вредоносное воздействие на систему.

Недостатком таких систем является наличие центрального сервера, который выполняет всю вычислительную нагрузку. В случае выхода его из строя возникает угроза работоспособности всей системы. Способом устранения этого недостатка может являться использование децентрализованной системы, которая самостоятельно следит бы за состоянием своих узлов и в случае необходимости перераспределяла задачи вышедшего из строя узла.

**Постановка задачи.** Развитие мировой коммуникационной инфраструктуры немислимо без использования альтернативных беспроводных сетей связи, которые позволяют обеспечить связь в труднодоступных местах и при чрезвычайных ситуациях, а также улучшают качество. Такие сети могут быть построены на основе беспилотных летательных аппаратов (БПЛА), организованных в группы по принципам построения мультиагентных систем [2].

Основным принципом мультиагентной системы являются самостоятельность каждого ее объекта, возможность получения только локальной информации и децентрализация управления. Каждый объект системы с его уникальным набором функций нуждается в контроле своих параметров.

БПЛА могут выполнять множество практических задач: поиск, транспортировка, мониторинг, поисково-спасательные работы, исследование труднодоступных мест и территорий. Однако такие задачи состоят из подзадач, и одними из важнейших являются координация движения БПЛА и мониторинг их параметров. Поэтому возникает необходимость разработать алгоритм, который позволил бы осуществлять эффективный мониторинг параметров БПЛА и управление основными их функциями [3].

**Модифицированный алгоритм мониторинга мультиагентной сети.** Наиболее простой и удобной платформой для мониторинга является веб-приложение.

Создать эффективную систему мониторинга параметров БПЛА можно на основе JavaScript-фреймворка с открытым исходным кодом AngularJS, предназначенного для разработки одностраничных приложений с целью расширения браузерных приложений на основе MVC-шаблона, а также упрощения тестирования и разработки [4].

Фреймворк работает с HTML, содержащим дополнительные пользовательские атрибуты, которые описываются директивами, и связывает ввод или вывод области страницы с моделью, представляющей собой обычные переменные JavaScript. Значения этих переменных задаются вручную или извлекаются из статических или динамических JSON-данных.

MVC – это не шаблон проекта, это конструктивный шаблон, который описывает способ построения структуры приложения, сферы ответственности и взаимодействие каждой из частей в данной структуре.

Структура MVC-шаблона изображена на рисунке 1.



Рисунок 1 – Структура MVC-шаблона

Приложение разделяется на три основных компонента и три области задач.

**Модель (Model)** состоит из данных и правил, которые используются для работы с ними, а также представляют концепцию управления приложением. В любом приложении вся структура моделируется как данные, которые обрабатываются определенным образом.

Модель дает контроллеру представление данных, которые запросил оператор. Она содержит наиболее важную часть логики приложения. Контроллер содержит в основном организационную логику для самого приложения.

**Вид (View)** обеспечивает различные способы представления данных, которые получены из модели. Он может быть шаблоном, который заполняется данными.

**Контроллер (Controller)** управляет запросами оператора, вызывает и координирует действие необходимых ресурсов и объектов.

Алгоритм получения и отображения данных веб-приложением, основанный на решении, предлагаемом в AngularJS, показан на рисунке 2.

В отличие от классического алгоритма в модифицированный алгоритм добавлены запрос о получении данных на БПЛА, графическая визуализация данных, интервал повторения алгоритма в 10 секунд.

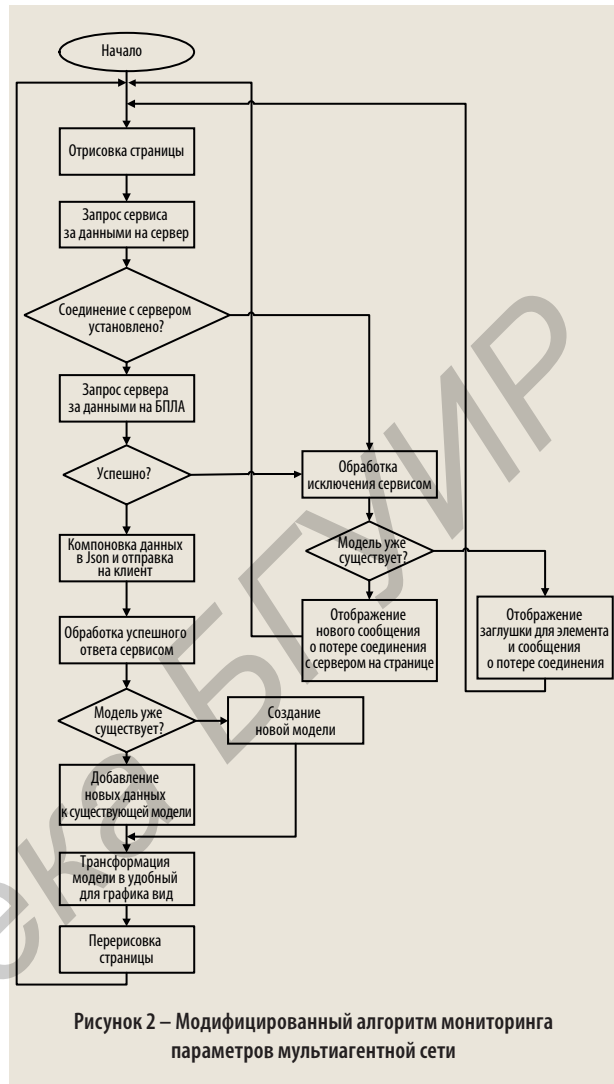


Рисунок 2 – Модифицированный алгоритм мониторинга параметров мультиагентной сети

**Информационное обеспечение.** На основе алгоритма разработана система мониторинга мультиагентной сети в виде одностороннего приложения на основе JavaScript-фреймворка AngularJS.

Для разработки информационного обеспечения использовалась среда IntelliJ IDEA, которая включает в себя все необходимые средства для разработки и отладки приложения, а также расширенный редактор кода с большим набором функций, редактор интерфейса и стандартный встроенный отладчик. Редактор интерфейса позволяет создавать виды и элементы в них с помощью визуального редактора, что сокращает количество строк кода и позволяет наглядно отображать схему графического интерфейса приложения.

Основной модуль (главная страница) приложения – то, что пользователь видит первым, когда заходит на страницу с приложением, – показан на рисунке 3. На этой странице отображается общая информация со всех датчиков.

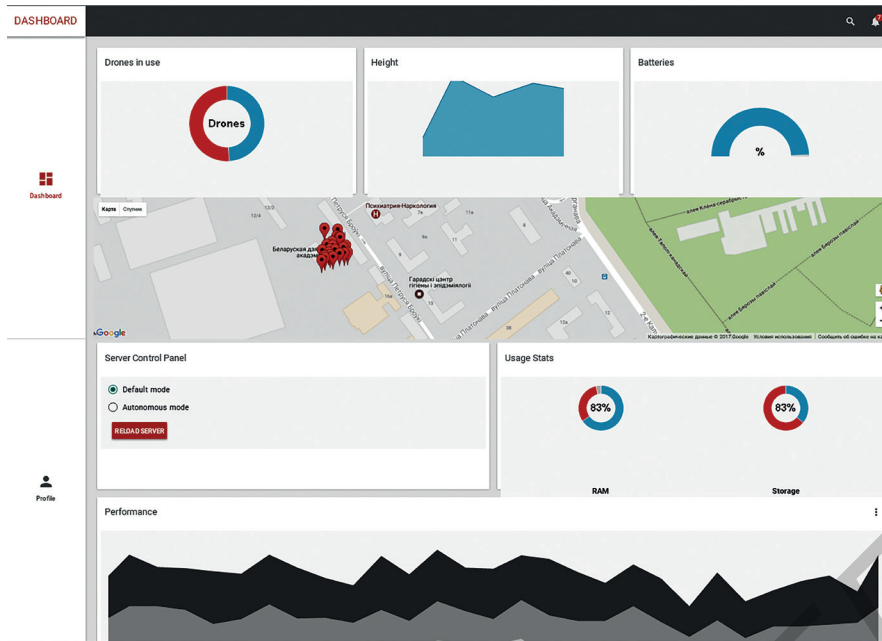


Рисунок 3 – Главная страница системы мониторинга

В данной реализации приложения на главной странице представлены следующие модули мониторинга:

- модуль мониторинга состояния каждого БПЛА;
- модуль мониторинга высоты полета БПЛА;
- расположение БПЛА на карте;
- панель управления режимами работы сервера;
- нагрузка на основные компоненты сервера;
- статистика работы компонентов сервера;
- поиск по приложению;
- количество непросмотренных уведомлений;
- сообщения;
- модуль перехода на страницу профиля оператора.

На рисунке 4 представлен фрагмент, в котором описана структура дополнительных модулей, используемых в проекте: Google material icons, Angular, Google Maps. Также в данном фрагменте иници-

```
<!doctype html>
<html class="no-js" ng-app="angularMaterialAdmin">
  <head>
    <meta charset="utf-8">
    <title>Real-time device dashboard</title>
    <meta name="description" content="Angular real-time device dashboard">
    <meta name="author" content="NikitaBobrik">
    <meta name="viewport" content="width=device-width">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
    <script src="http://maps.google.cn/maps/api/js"></script>
  </head>
  <body>
    <div ui-view layout="row" layout-fill></div>
  </body>
</html>
```

Рисунок 4 – Фрагмент кода, отвечающий за отображение главной страницы

ализируется главный модуль Angular-приложения – angularMaterialAdmin.

Стоит также отметить, что формирование всех зависимостей проекта для JavaScript и CSS используется сборщиком проекта Gulp. Сборщик способен выполнять определенные задачи в том или ином порядке.

#### Задача partials

```
gulp.task('partials', function () {
  return gulp.src ([
```

формирует все html-шаблоны, сохраняет их в нужную директорию и минифицирует для того, чтобы эти файлы занимали как можно меньше места.

#### Задачи images, fonts, misc

```
gulp.task('images', function () {
  return gulp.src(paths.src + '/assets/images/**/*')
    .pipe(gulp.dest(paths.dist + '/assets/images/'));
```

```
gulp.task('fonts', function () {
  return gulp.src($mainBowerFiles())
```

```
gulp.task('misc', function () {
  return gulp.src(paths.src + '/**/*.ico')
```

обрабатывают картинки, шрифты и иконки соответственно и сохраняют их в нужную директорию.

Кроме вышеизложенных задач, в проекте присутствуют и другие не менее важные задачи.

#### Задачи inject и styles

```
gulp.task('html', ['inject', 'partials'], function () {
  var partialsInjectFile = gulp.src (paths.tmp +
    '/partials/templateCacheHtml.js', {read: false});
  var partialsInjectOptions = {
```

отвечают за компилирование SCSS в CSS, обработку CSS, а также всех JavaScript-модулей и добавление этих зависимостей в корневой файл проекта.

Таким образом, использование в приложении Gulp берет на себя часть задач, связанных с компиляцией, сборкой, внедрением зависимостей и даже проверкой синтаксиса кода, что значительно упрощает и ускоряет разработку приложения, а также предупреждает от ошибок, связанных с написанием и последующей интерпретацией кода.

В главном Angular-контроллере приложения происходит подключение дополнительных внешних модулей, необходимых для работы приложения, инициализация модели и описание методов для взаимодействия с представлением.

Также для работы одностраничного приложения необходима соответствующая навигация. Для этой задачи отлично подходит внешний модуль `ui.router`, который используется в конфигурационном файле `index.js`. Здесь описаны все роутинги, используемые в приложении, а также приводится цветовая палитра, используемая для системы мониторинга. При желании конечного пользователя цветовая схема может быть легко изменена благодаря тому, что цвета приложения не заданы статически во всех местах, а описаны в одном месте и используются всем приложением динамически.

Основной информативной частью приложения являются графики, на которых представлена и систематизирована информация с различных датчиков.

Все компоненты, используемые в данном коде, стилизуются библиотекой `Angular Material`, содержащей в себе готовые `Angular`-директивы для красивого отображения данных. Это удобно, т. к. важным фактором в пользовательских приложениях является дизайн. Разработчик же зачастую не разбирается в дизайне либо у него попросту нет на это времени, а эта библиотека предлагает набор готовых дизайнерских решений, придуманных компанией `Google`, – `material design`.

Кроме основного модуля, разработаны следующие модули: профиль, статистика активных БПЛА, высота ведущего БПЛА, заряд батарей ведущего БПЛА, карта с местонахождением БПЛА, панель управления режимами работы сервера, модуль использования ресурсов сервера, мониторинг производительности.

**Выводы.** На основе предложенного алгоритма можно создавать аналогичные алгоритмы для контроля практически любого параметра БПЛА – агента мультиагентной сети и мониторинга комплекса параметров группы.

Предлагаемая система мониторинга является простой, нересурсоемкой и не требует установки дополнительного программного обеспечения.

#### ЛИТЕРАТУРА

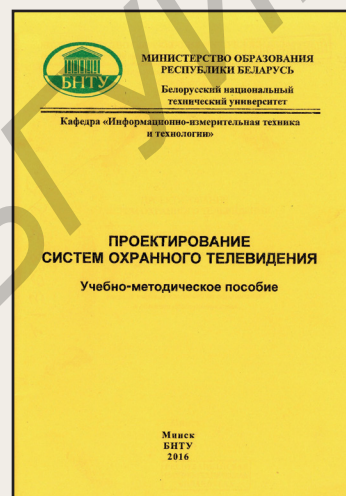
1. Гайфулин, Т.А. Анализ современных систем мониторинга / Т.А. Гайфулин, Д.С. Костомаров // Известия тульского государственного университета. – 2013. – № 9. – С. 51–55.
2. Дуйнова, Ю.А. Альтернативная коммуникационная сеть и правило управления группой ее элементов с голономной моделью связей / Ю.А. Дуйнова, С.И. Половения // Веснік сув'язі. – 2016. – № 6 (140). С. 30–38.
3. Половения, С.И. Алгоритм мониторинга мультиагентной сети / Ю.А. Дуйнова, Н.М. Бобрик, С.И. Половения // Инновационные технологии научного развития: материалы IX Междунар. науч.-практ. конф., г. Казань, РФ, 20 мая 2017 г. – С. 14–15.
4. **Lukas Ruebbelke with Brian Ford.** *AngularJS in Action.* – Manning Publications, 2015.

*A system for monitoring parameters from UAV group sensors has been developed. Analyzed existing monitoring systems, developed an algorithm for the UAV group management model, developed a web application for monitoring and control of UAV flight. The developed software system has the possibility of modernization, or changes according to the customer's desire and requirement. Different sets of sensors, flight monitoring and monitoring mechanisms can be used for different purposes.*

Получено 05.06.2017.



Один из крупнейших информационных центров Беларуси предлагает специалистам ознакомиться с новыми изданиями по теме «Телекоммуникации и связь»



**Тявловский К.Л.**

**Проектирование систем охранного телевидения: учебно-методическое пособие для студентов специальности 1-38.02.03 «Техническое обеспечение безопасности» / К.Л. Тявловский [и др.]; Министерство образования Республики Беларусь, Белорусский национальный технический университет, кафедра «Информационно-измерительная техника и технологии».** – Мн.: БНТУ, 2016. – 68 с.: ил., табл. ISBN 978-985-550-798-8

В издании рассмотрены основные методы и этапы проектирования систем охранного телевидения различных типовых объектов. Показана необходимость учета различных факторов, влияющих на конфигурацию и состав системы охранного телевидения. Пособие может быть использовано при самостоятельной работе, а также при выполнении курсовых и дипломных проектов.