IOT Vega Server

Руководство

Введение

В данном документе описаны возможности сервера IOT Vega Server, способы взаимодействия с ним, а также настройки при первом запуске.

Оглавление

Описание IOT Vega Server

Сетевой сервер IOT Vega Server это инструмент для организации сетей стандарта LoRaWAN любого масштаба.

Предназначен для управления опорной сетью базовых станций, работающих под управлением ПО Packet forwarder от компании Semtech, приема данных с оконечных устройств и передачи их внешним приложениям, а также передачи данных от внешних приложений на LoRaWAN устройства.

Сервер работает по спецификации LoRaWAN 1.02 и поддерживает любые оконечные устройства, работающие согласно данной версии спецификации.

Все принятые от оконечных устройств данные сохраняются во встроенной в IOT Vega Server базе данных и всегда доступны для внешних приложений.

Открытый API, основанный на технологии Web Socket позволяет подключать к IOT Vega Server внешние приложения и использовать возможности LoRaWAN сетей в ваших проектах.

IOT Vega Server выпускается в виде консольного приложения для операционных систем Windows, Linux и Linux-ARM (сборка для запуска на платформе базовых станций).

Для управления сервером используется приложение IOT Vega **AdminTool** с простым дружественным интерфейсом. AdminTool открывает перед администратором сервера широкие возможности по управлению сетью LoRaWAN. С AdminTool вы можете добавлять в сеть новые оконечные устройства LoRaWAN, просматривать карту сети, контролировать базовые станции, а также управлять правами пользователей. IOT Vega AdminTool предоставляется в виде Web-приложения.

Для работы с устройствами используется клиентское приложение IOT Vega **Pulse** с широким спектром возможностей извлечения данных, обработки и представления в различных форматах (таблица, график, отчет, диаграмма).

Возможности

- Поддержка любых оконечных устройств LoRaWAN 1.0.1
- Поддержка оконечных устройств класса A и C
- Встроенная база данных
- Поддержка работы с внешней базой данных
- Удобное приложение администратора
- Построение карты сети
- Управление пользователями сети
- Гибкая настройка подключенных к серверу устройств
- Поддержка произвольных частотных планов
- Онлайн просмотр пакетов с каждого устройства (нисходящие и восходящие пакеты)
- Графики связи для каждого устройства в сети

Установка

Перед обновлением (установкой новой версии) сервера, рекомендуется выполнить резервное сохранение базы данных (БД) и файла настроек по следующим причинам:

- новая версия сервера модифицирует структуру БД, поэтому, в случае необходимости возврата к предыдущей версии, сервер не сможет работать с изменённой БД;
- файл с настройками заменяется файлом из установочного пакета, тем самым возвращает старые настройки к значениям по умолчанию. Обновление файла настроек необходимо, поскольку в новой версии сервера добавлены новые настройки.

Версия для Windows не требует установки. Необходимо распаковать архив и запустить исполняемый файл.

Для корректной работы сервера необходимо обязательно установить обе версии библиотек, которые находятся в директории **IOT Vega Server/msvc c++ 2013**. После этого можно начинать работать с сервером.

Версия для Linux поставляется в виде установочного DEB пакета для 32-х и 64-х разрядных систем.

Процесс установки приложения на Linux:

- скачать файл на ПК;
- установить, выполнив команду: **sudo dpkg –i /путь/к/файлу/iot-vega-server-1.1.5.deb**;
- для конфигурирования необходимо изменить содержимое файла с настройками **/opt/iot-vega-server/settings.conf**;
- для запуска сервера в работу необходимо выполнить скрипт **sudo ./iot-vega-server.sh** в директории **/opt/iot-vega-server** либо написать новый скрипт с указанием пути к библиотекам (директория установки).

Для корректной работы плагина внешней базы данных MySQL на операционной системе **Linux** необходимо наличие библиотек libssl (версии младше 1.1.0) в системе, а именно определенных символьных ссылок. С пакетом поставляются ссылки, но зачастую либо версии библиотек разнятся, либо места установок. Из-за чего оказывается невозможным запуск сервера. Для решения проблемы восстановления ссылок необходимо выполнить следующее:

- перейти в директорию с файлами сервера **cd /opt/iot-vega-server**;
- выполнить команду **ls –al**. В результате будет отображено содержимое директории. Обращаем внимание на две символьные ссылки:
  - libcrypto.so.10 -> /lib/x86_64-linux-gnu/libcrypto.so.1.0.0
  - libssl.so.10 -> /lib/x86_64-linux-gnu/libssl.so.1.0.0

  Если в системе имеются файлы, на которые указывают эти ссылки, то после стрелки будет указано их реальное расположение. В таком случае можно запускать сервер.
- Если соответствующие файлы не обнаружены в системе, необходимо самостоятельно выполнить поиск (команда **find / -name "libcrypto*"**) и создать символьные ссылки (команда **ln –s "путь_к_библиотеке" libcrypto.so.10**) для

обоих файлов. После чего необходимо снова выполнить команду **ls –al** и убедиться что обе ссылки стали корректными.

- Если сервер не запускается после выполненных операций, необходимо перейти в папку sqldrivers (команда **cd sqldrivers**). Здесь необходимо выполнить следующую команду: **LD_LIBRARY_PATH=/opt/iot-vega-server/ ldd libqsqlmysql.so**.

Примерный результат выполнения:

```
linux-vdso.so.1 =>  (0x00007ffd035fe000)
libQt5Sql.so.5 => /opt/iot-vega-server/libQt5Sql.so.5 (0x00007f9a76730000)
libQt5Core.so.5 => /opt/iot-vega-server/libQt5Core.so.5 (0x00007f9a76010000)
libmysqlclient.so.18 => /opt/iot-vega-server/libmysqlclient.so.18 (0x00007f9a75a40000)
libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f9a75822000)
libz.so.1 => /lib/x86_64-linux-gnu/libz.so.1 (0x00007f9a75609000)
libssl.so.10 => /opt/iot-vega-server/libssl.so.10 (0x00007f9a753aa000)
libcrypto.so.10 => /opt/iot-vega-server/libcrypto.so.10 (0x00007f9a74fce000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f9a74dca000)
libstdc++.so.6 => /usr/lib/x86_64-linux-gnu/libstdc++.so.6 (0x00007f9a74ac6000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f9a747c0000)
libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007f9a745aa000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f9a741e1000)
libicui18n.so.56 => /opt/iot-vega-server/libicui18n.so.56 (0x00007f9a73d48000)
libicuuc.so.56 => /opt/iot-vega-server/libicuuc.so.56 (0x00007f9a73990000)
libicudata.so.56 => /opt/iot-vega-server/libicudata.so.56 (0x00007f9a71fad000)
librt.so.1 => /lib/x86_64-linux-gnu/librt.so.1 (0x00007f9a71da5000)
libgthread-2.0.so.0 => /usr/lib/x86_64-linux-gnu/libgthread-2.0.so.0 (0x00007f9a71ba3000)
libglib-2.0.so.0 => /lib/x86_64-linux-gnu/libglib-2.0.so.0 (0x00007f9a7189b000)
/lib64/ld-linux-x86-64.so.2 (0x00007f9a76b8a000)
libpcre.so.3 => /lib/x86_64-linux-gnu/libpcre.so.3 (0x00007f9a7165d000)
```

Данная команда отображает зависимости для плагина базы данных MySQL. Если обнаружены пустые строки – необходимо обратиться в техническую поддержку компании.

Версия для Linux-ARM поставляется в виде архива и не требует установки. Архив **iot-vega-server (armhf) v1.1.5.tar.gz** необходимо скопировать в папку **/opt/** и разархивировать командой **tar –xvf iot-vega-server (armhf) v1.1.5.tar.gz**.

Далее необходимо изменить файл конфигурации под ваши требования. В работу сервер запускается через скрипт **iot-vega-server.sh**.

Работа с внешними базами данных не поддерживается.

Настройки

Сервер работает в автономном режиме, требуется лишь первоначальное конфигурирование перед запуском сервера. Чтобы осуществить первоначальную настройку сервера необходимо зайти в папку с файлами сервера и открыть settings.conf с помощью любого текстового редактора (например, «Блокнот»).

При редактировании файла settings.conf нельзя использовать символ «!» - восклицательный знак

Содержимое файла выглядит следующим образом:

```
# Host connection settings
[host]
        # IP-address for UDP connection (gateway connection)
ip=127.0.0.1
        # Port for UDP connection (gateway connection)
udpPort=8001
        # Port for TCP (WebSocket) connection
tcpPort=8002
        # "path" part of webSocket address
webSocketPath=/
        # Flag of using SSL encryption for WebSocket
useSSL=0
        # SSL certificate filename (certificate must be in server's directory)
certFileName=cert.crt
        # SSL key filename (key must be in server's directory)
keyFileName=key.key

# LoRaWAN network settings
[lora]
        # LoRaWAN network identifier (should be random between 1 and 127)
networkID=1
        # Flag for using Plug-and-Play gateways function.
        # If this value is 1, server would automatically append all gateways which connected to one
usePnPGateway=1

# Super user options
[root]
        # Login for super user
root=root
        # Password for super user (recommendation: change this password to your own)
password=123

# Console settings (volume of debug information)
[console]
        # Maximum level of console messages that will be shown (levels of messages represented below)
maxMsgLevel=20
        # Maximum level of console messages that will be saved into LOG file (levels of messages represented below)
maxLogMsgLevel=0
        #       Console message levels:
        # errors        = 0
```

```
        # uplink        = 1
        # downlink      = 2
        # warning       = 3
        # info          = 4
        # debug         = 20

# External DataBase settings
[external_db]
        # Flag of using external DB
useExternalDb=0
        # Type of external DB. Supported only next types:
        #       MYSQL
        #       SQLITE
typeExternalDb=MYSQL
        # Name of external DB (schema's name for MYSQL)
nameExternalDb=server
        # IP and port of DB's server ("localhost" is supported)
ipExternalDb=127.0.0.1
portExternalDb=5505
        # User login and password (user should have maximum level of privileges)
userExternalDb=admin
passwordExternalDb=admin

# self-test settings
[selftest]
        # Flag for enabling anonymous reports with possible server errors
enableReports=1
```

Секция [host] содержит настройки портов и сетевых соединений:
ip – IP адрес через который будут работать базовые станции;

Данный параметр необходимо изменить на корректный IP адрес, через который будут работать базовые станции. В противном случае связи с базовыми станциями не будет

udpPort – порт для базовых станций (БС);
tcpPort – порт для внешних приложений (в том числе для AdminTool);
webSocketPath – содержимое поля «путь» в адресе WebSocket. Если задать в данном поле /ws, то при подключении из AdminTool нужно будет указывать ws://address:port/ws;
useSSL – флаг разрешения использования SSL шифрования для WebSocket;
certFileName – имя файла, в котором содержится SSL сертификат (с расширением);
keyFileName – имя файла, в котором содержится SSL ключ (с расширением).

Для налаживания корректной работы SSL шифрования необходимо выполнить следующие действия:

- получить SSL сертификат (и ключ), подписанный доверенным центром сертификации (это может быть и бесплатный доверенный центр, например sslforfree или letsencrypt);
- установить OpenSSL версии 1.0.2 (сервер с версией OpenSSL 1.1.0 не работает);
- скопировать файлы сертификата и ключа (для linux достаточно создать ссылки) в директорию с сервером;
- разрешить использовать SSL шифрование, прописать соответствующие имена файлов (с сертификатом и ключом) в конфигурационном файле;

- запустить сервер, проследить за возможными ошибками.

Версия OpenSSL для Windows доступна на сайте iotvega.ru по ссылке http://iotvega.com/content/ru/soft/server/Win32OpenSSL-1_0_2n.zip.

Секция [lora] содержит данные для идентификации в LoRa-сети:
networkID – определяет идентификатор LoRaWAN сети. Перед запуском сервера необходимо установить случайное значение в диапазоне от 1 до 127 включительно;

Если в непосредственной близости будут работать несколько сетей LoRaWAN, которые используют одинаковые NetworkID – каждая из сетей продолжит работу в нормальном режиме, но периодически будут появляться ошибки о некорректности NwkSKey того или иного устройства:

"INVALID_DEVICE_NETWORK_SESSION_KEY"

usePnPGateway – флаг разрешения автоматического добавления БС на сервер. Если эта опция разрешена, сервер любую неизвестную ранее БС будет добавлять в список зарегистрированных БС с параметрами по умолчанию.

Секция [root] определяет идентификационные данные суперпользователя, где root – логин, а password – пароль.

Секция [console] задает настройки объема выводимой информации в консоль и сохранения ее в LOG файл (параметры данной секции обновляются каждую минуту, поэтому для изменения уровня выводимых сообщений перезапуск сервера не нужен):
maxMsgLevel – максимальный уровень сообщений (включительно), которые будут отображаться в консоли сервера. Расшифровка уровней сообщений приведена ниже;
maxLogMsgLevel – максимальный уровень сообщений, которые будут сохраняться в LOG файл.

Уровни сообщений консоли:

- 0 = сообщения с критическими ошибками;
- 1 = исходящие (uplink) сообщения;
- 2 = восходящие (downlink) сообщения;
- 3 = сообщения с предупреждениями;
- 4 = информационные сообщения (зачастую несут информацию отладочного характера);
- 20 = сообщения с отладочной информацией.

Секция [external_db] содержит настройки по работе с внешней базой данных (БД) – после изменения настроек необходимо перезапустить сервер:
useExternalDb – флаг, разрешающий работу с внешней БД. Может принимать значения 1 или 0;
typeExternalDb – параметр, задающий тип внешней БД в виде строки. В настоящий момент поддерживаются два типа БД:

- MYSQL;

- SQLITE – создание и работа с файлом БД с наименованием, отличающимся от наименования по умолчанию («server.db»).

nameExternalDb – наименование базы данных. Для SQLITE соответсвует наименованию файла, для MYSQL – наименование схемы;
ipExternalDb – IP адрес сервера внешней СУБД (не используется для SQLITE) в виде строки (поддерживается значение «localhost»);
portExternalDb – порт сервера внешней СУБД (не используется для SQLITE);
userExternalDb – имя пользователя для авторизации на сервере внешней СУБД (не используется для SQLITE);
passwordExternalDb – пароль для авторизации пользователя на сервере внешней СУБД (не используется для SQLITE).

Пользователь должен обладать максимальными правами на внесение изменений в состав БД

При переходе с локальной БД на внешнюю, при первом запуске сервера данные из локальной БД будут перемещены во внешнюю БД

Секция [selftest] содержит флаг разрешения отправки анонимных сообщений с возможными обнаруженными неисправностями.

Сервер в процессе своей работы накапливает информацию о возможных сбоях и может раз в сутки отправлять данные отчеты разработчикам компании Вега-Абсолют. Эта информация будет крайне полезной при поиске и решении возможных сбоев в работе сервера. Отправляемая информация не содержит никаких конфиденциальных данных.

Команды для сервера приведены в файле API IOT Vega Server Rev21.pdf.

IOT Vega AdminTool

ПО IOT Vega AdminTool является удобным Web-приложением для администрирования сервера и позволяет добавлять в сеть новые оконечные устройства LoRaWAN, просматривать карту сети, контролировать базовые станции, а также управлять правами пользователей.

Разберем **пример подключения новой или редактирования параметров существующей БС** на сервере (Рис. 1).

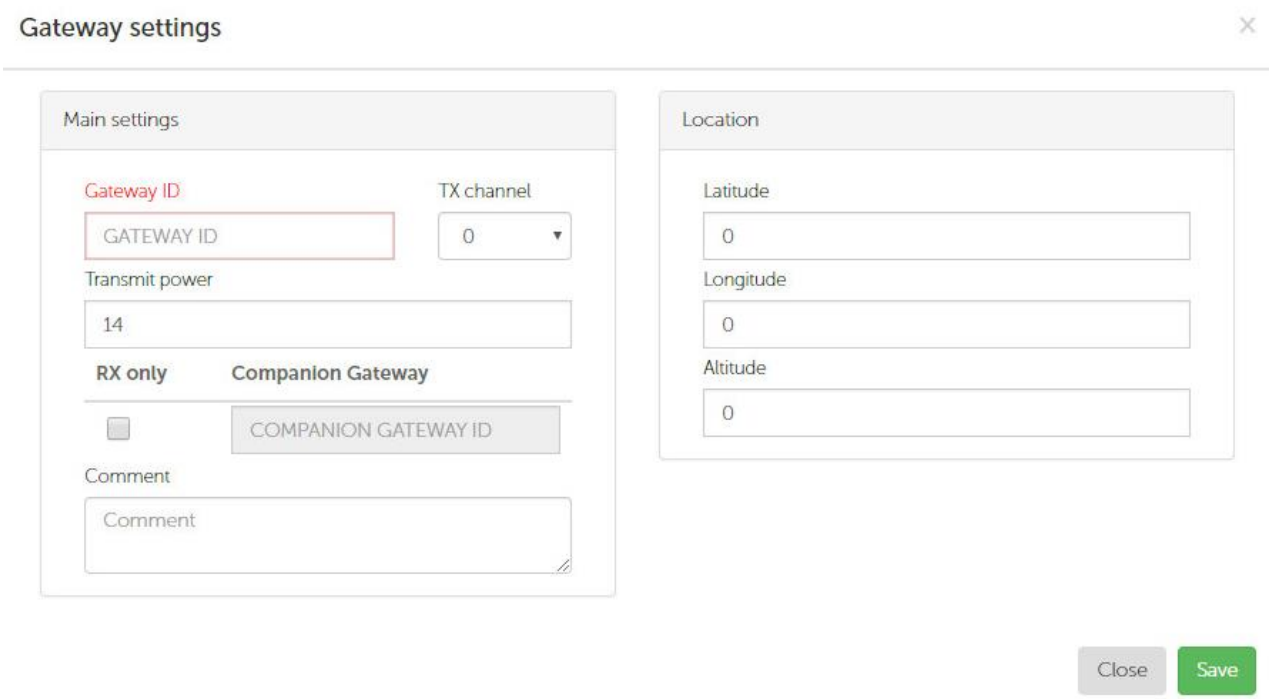

Рис. 1. Окно подключения БС на сервер.

Обязательные настройки:

Gateway ID – идентификатор БС (16 шестнадцатеричных символа – 8 Байт);

TX channel – канал БС, используемый для передачи сообщений на оконечные устройства (downlink). Данный параметр указывается в настройках ПО “packet_forwarder” на БС (обычно в файле “global_conf.json”). По умолчанию используется 0 канал;

Transmit power – мощность вещания БС. Максимальная мощность вещания обычно обусловлена схемотехникой БС и ограничена в ПО “packet_forwarder”; при превышении данного параметра БС будет возвращать ошибку с соответствующим кодом.

Здесь необходимо оговорить критерии, по которым выбирается необходимое значение мощности вещания.

Обратимся к файлу «global_conf.json» из настроек ПО «packet_forwarder» на БС.

В списке допустимых наборов параметров вещания «tx_lut_..» содержится набор мощностей, на которых БС может осуществлять передачу. Для стандартного списка это диапазон от -6 до 27 дБм с разным шагом (всего 16 значений).

Так же имеется параметр «antenna_gain», который определяет усиление антенны (в дБм).

В итоге, разница величин «Transmit power» и «antenna_gain» должна соответствовать одному из значений из списка «tx_lut_..». Например, «antenna_gain» = 3 дБм и планируется передавать данные на мощности 10 дБм («Transmit power»). Разница между «Transmit power» и «antenna_gain» составляет 7дБм, но такого значения нет в списке «tx_lut_..» (и если использовать такой набор параметров, то при попытке отправки данных, БС будет присылать сообщение с ошибкой о некорректной мощности). Ближайшие разрешенные значения из «tx_lut_..» это 6 и 10 дБм, соответственно, для такой конфигурации разрешенные значения «Transmit power» будут 9 или 13 дБм.

RX only – флаг, указывающий, что БС работает только на прием. При установке данного флага, передача данных на оконечные устройства через соответствующую БС запрещена. Данная опция введена для организации «полно-дуплексной БС»: это вариант когда в непосредственной близости установлено две БС и одна из них работает только на прием, а вторая работает в обычном режиме. При этом достигается непрерывность прослушивания радиоэфира;
Companion Gateway – идентификатор БС компаньона, работающей в нормальном режиме при организации «полно-дуплексной БС»;
Comment – поле для комментария (например, наименования БС);

Область Location – содержит поля для ввода координат размещения БС: широта, долгота, высота. Если в БС имеется встроенный GPS приемник, введенные координаты будут обновлены актуальными значениями автоматически.

Разберем пример регистрации нового устройства или изменения параметров существующего (Рис. 2 и Рис. 3).

На Рис. 2 представлен снимок формы для регистрации устройства с указанием основных параметров.

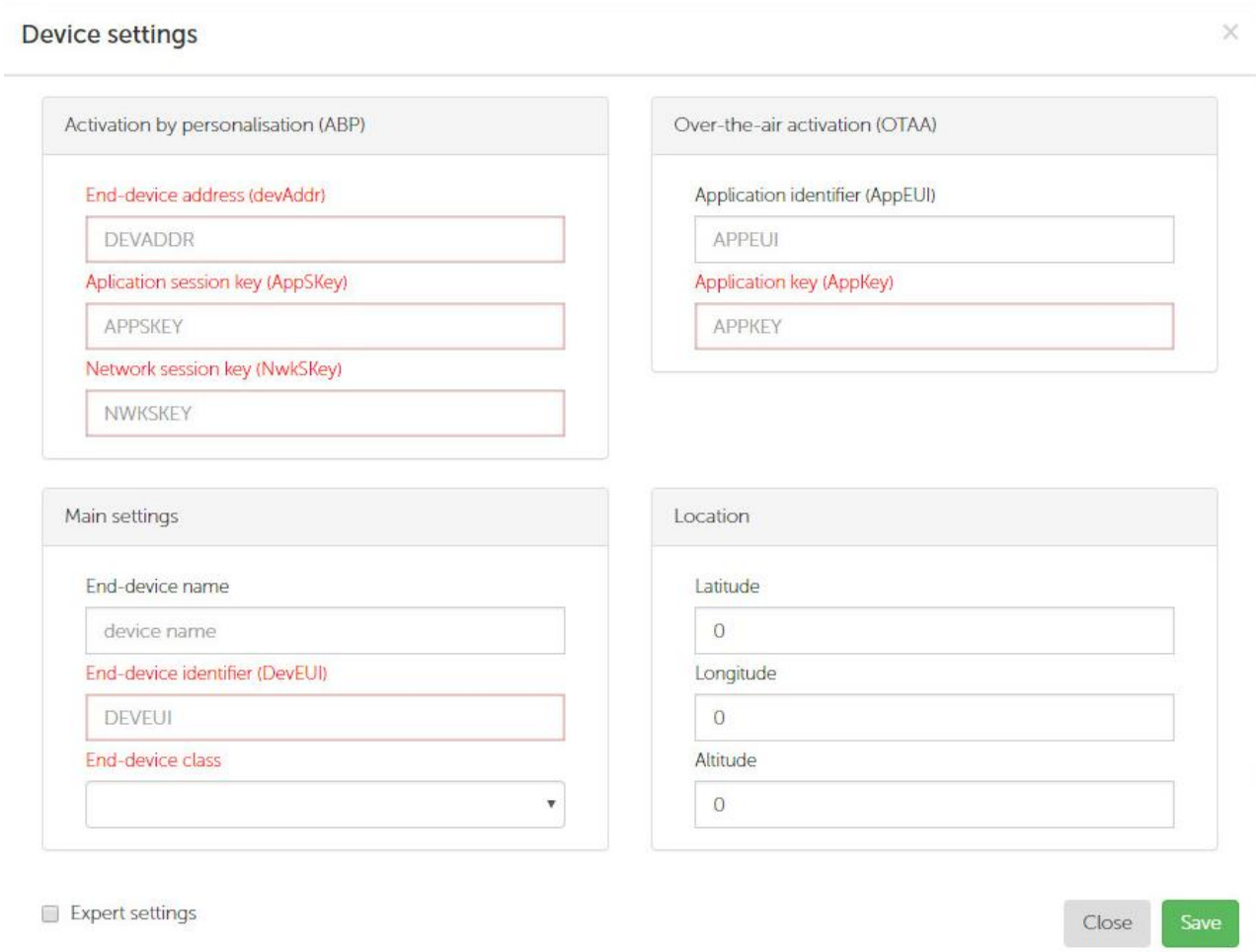

Рис. 2. Основные параметры для регистрации устройства на сервере.

Область **Activation by personalization (ABP)** содержит параметры необходимые для регистрации устройства на сервере через ABP:
devAddr – адрес устройства в LoRaWAN сети. Это 32-х разрядное число в шестнадцатеричном виде, например 012345AB;
AppSKey – сессионный ключ приложения – это строка из 32-х шестнадцатеричных символов;
NwkSKey – сетевой сессионный ключ – это строка из 32-х шестнадцатеричных символов.

Область **Over-the-air activation (OTAA)** содержит параметры необходимые для регистрации устройства на сервере через OTAA:
AppEUI – EUI идентификатор приложения устройства – строка из 16-ти шестнадцатеричных символов;
AppKey – ключ приложения устройства – это строка из 32-х шестнадцатеричных символов.

Для регистрации устройства достаточно указать хотя бы один из типов активации или оба сразу.

Область **Main settings** содержит следующие параметры:
End-device name – пользовательское наименование устройства;
DevEui – EUI идентификатор устройства (уникальный номер устройства) - строка из 16-ти шестнадцатеричных символов;
End-device class – класс устройства. Этот параметр может принимать два значения: CLASS_A либо CLASS_C. Поддержка устройств CLASS_B находится в разработке.

Область **Location** содержит поля для указания координат размещения устройства: широта, долгота, высота.

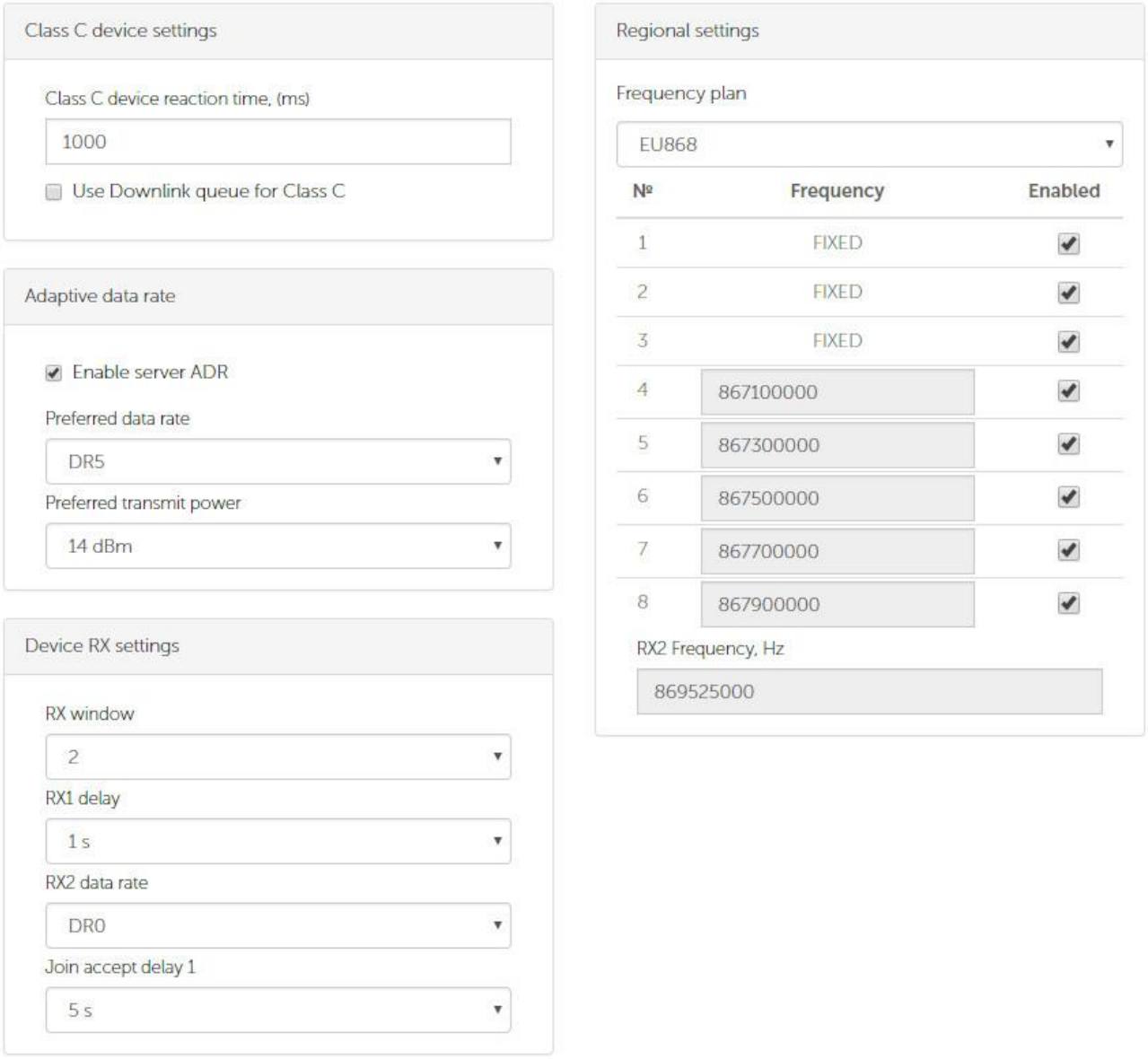

Рис. 3. Экспертные параметры для регистрации устройства на сервере.

Нажав на галочку **Expert settings**, можно получить доступ к экспертным настройкам устройства (см Рис. 3).

Область **Class C device settings** доступна только устройствам CLASS_C. Доступны следующие параметры:

Class C reaction time – время реакции устройства CLASS_C. Это время между окончанием приема сообщения сервера устройством и началом отправки возможного ответа от устройства (т.е. по сути время на подготовку возможного ответа). Параметр введен для улучшения качества работы с сообщениями от сервера к устройствам класса C;

Use downlink queue for Class C – флаг разрешающий помещать пакеты для отправки в очередь отправки. Зачастую устройства CLASS_C работают в “online” режиме и если пакет не доставлен до устройства в настоящий момент, то его доставка уже не требуется – в таком случае очередь пакетов не нужна. Даже наоборот, очередь сыграет негативную роль, как накопитель уже устаревшей информации, которая

будет передана в обязательном порядке, занимая эфирное время. По умолчанию опция выключена.

Область Adaptive data rate содержит настройки алгоритма ADR, предназначенного для автоматического изменения скорости вещания устройств в зависимости от качества связи (в условиях хорошего приема скорость будет увеличиваться, тем самым снижая время передачи пакетов и увеличивая срок службы устройства от батареи):
Enable server ADR – флаг, разрешающий работу алгоритма ADR на сервере для конкретного устройства (если галочка снята, сервер не будет регулировать скорость передачи устройства, даже если алгоритм ADR на самом устройстве активирован);
Preferred data rate – значение DR (скорости), к которому будет стремиться алгоритм ADR сервера;
Preferred transmit power – мощность передатчика устройства, которую сервер задаст ему при очередном сеансе связи.

Параметр Preferred transmit power стоит изменять (уменьшать относительно значения по умолчанию) в том случае если качество связи для устройства на максимальной скорости удовлетворительно

Область Device RX settings содержит настройки приемных окон устройства:
RX window – номер приемного окна (1 или 2), через которое сервер по умолчанию будет передавать данные. Если установить 1-ое окно, то в случае если сервер не может отправить данные в 1-ое приемное окно (например, нет свободной БС), то будет выполнена попытка отправить данные во 2-ое приемное окно;
RX1 delay – задержка открытия устройством первого приемного окна (по умолчанию 1 секунда). 2-ое приемное окно открывается всегда (если в первое приемное окно данные не были приняты) через 1 секунду после 1-го;
RX2 data rate – скорость передачи данных для 2-го приемного окна;
Join accept delay 1 – задержка открытия устройством первого приемного окна для получения регистрационной информации при активации в сети способом OTAA (по умолчанию 5 секунд).

Если базовая станция работает через мобильный интернет, то величина задержек доставки пакетов до такой станции может достигать существенных величин. Поэтому для избегания проблем с работой устройств (отправка подтверждений и MAC-команд) необходимо время открытия первого приемного окна сдвинуть более чем на 1с. Обычно достаточно 3с интервала. В некоторых случаях потребуется сдвиг на больший интервал – необходимо следить за стабильностью работы сети.

Область Regional settings содержит настройки частотного плана для соответствующего устройства. Здесь предлагается выбор из двух существующих наборов (официальный европейский частотный план и один из вариантов российского частотного плана), а также имеется возможность настроить уникальный набор частот.

Каждый из наборов частот состоит из:

- трех каналов с фиксированными частотами (данные каналы жестко заданы в устройстве без возможности их изменения через протокол LoRaWAN);
- пяти каналов для приема и передачи сообщений. Если канал не используется, необходимо задать нулевую частоту и отключить соответствующую галочку в настройках **Enabled**;
- одного канала, определяющего частоту приема данных для второго приемного окна.

Маска активных каналов (список флагов **Enabled**) расположена слева от полей ввода частоты и содержит флаги отключения/включения соответствующих каналов (частот) для передачи данных на устройстве. Данный параметр передается устройству при каждой JOIN процедуре и в каждом сообщении алгоритма ADR (если ADR разрешен).

Приложение А. Описание структуры базы данных

ПО IOT Vega Server в процессе своей работы постоянно использует базу данных (встроенную – SQLite или внешнюю). Для обеспечения начала работы требуются минимальные настройки по доступу к внешней базе данных, а для встроенной базы данных настройки не требуются вообще. Структура базы данных в обоих случаях создается автоматически. Следует исключить влияние пользователя непосредственно на структуру базы данных, чтобы не нарушить работу сервера.

Тем не менее, существует вариант работы с базой данных в режиме только чтения. Например, при считывании накопленных данных от устройства. Такой подход призван снять нагрузку с сервера на вычитывание данных от устройств через API и в целом ускорить работу системы с запросами пользователя.

База данных состоит из следующих таблиц:

- «bs» - перечень подключенных базовых станций;
- «devices» - перечень подключенных устройств;
- «rawdata» - данные принятые от устройств и переданные на устройства;
- «queuetransmit» - очередь отправки пакетов на устройство;
- «coveragemap» - список базовых станций, через которые были приняты данные от соответствующего устройства. По данной таблице можно построить карту покрытия сети;
- «deviceattributes» - список свойств соответствующего устройства;
- «users» - список пользователей;
- «userdevices» - список устройств, доступных соответствующему пользователю.

Описание таблиц «queuetransmit», «deviceattributes», «users» и «userdevices» не будут приведены, поскольку в этих таблицах либо хранится буферная информация, либо информация в специфическом формате. Предполагается, что данная информация не потребует прямого доступа для чтения и всегда доступна через быстро выполняемые команды API

Таблица 1 - Структура полей таблицы «bs» базы данных

Поле	Тип данных	Описание
id	INTEGER	Идентификатор (не используется)
mac	TEXT	ID базовой станции
comment	TEXT	Комментарий
latency	INTEGER	Текущая задержка по сети БС-сервер
downport	INTEGER	Порт для передачи данных на БС
hostaddress	TEXT	IP адрес БС для передачи данных
longitude	REAL / DOUBLE	Координата БС – долгота
latitude	REAL / DOUBLE	Координата БС – широта
altitude	INTEGER	Координата БС – высота на местности
active	INTEGER	Флаг активности БС (на связи)
rxonly	INTEGER	Флаг работы БС только на прием
complimentarybsmac	TEXT	ID БС-компаньона, через которую разрешается передавать данные, если текущая БС работает только на прием
downlinkchannel	INTEGER	Канал для передачи данных
maxpower	INTEGER	Мощность передачи через БС
lastonline	INTEGER	Время последней активности БС

Таблица 2 - Структура полей таблицы «rawdata» базы данных

Поле	Тип данных	Описание
data	BLOB	Данные в HEX массиве
macbs	TEXT	ID базовых станций, через которые был передан пакет, объединены через «+»
deveui	TEXT	ID устройства
rssi	INTEGER	RSSI
snr	REAL / DOUBLE	SNR
freq	INTEGER	Частота в Гц
sf	INTEGER	SF
time	INTEGER	время в формате мс, прошедших с начала UNIX эпохи
fcntup	INTEGER	номер пакета
port	INTEGER	используемый порт
ack	INTEGER	наличие флага ACK в пакете
macdata	BLOB	MAC команды
type	INTEGER	Тип пакета (см описание команды «get_data» в API)
direction	INTEGER	Направление передачи: «UPLINK» или «DOWNLINK»
status	TEXT	Статус (см описание команды «get_data» в API)
activationside	INTEGER	Флаг, показывает используемый тип активации при передачи текущего пакета

Таблица 3 - Структура полей таблицы «coveragemap» базы данных

Поле	Тип данных	Описание
id	INTEGER	Идентификатор (не используется)
deveui	TEXT	ID устройства
macbs	TEXT	ID базовой станции
snr	REAL / DOUBLE	SNR
rssi	INTEGER	RSSI
time	INTEGER	время в формате мс, прошедших с начала UNIX эпохи

Таблица 4 - Структура полей таблицы «devices» базы данных

Поле	Тип данных	Описание
deveui	TEXT	ID устройства
devname	TEXT	Наименование устройства
appeui	TEXT	AppEUI устройства
appkey	BLOB	AppKEY
devaddrabp	INTEGER	DevAddr для ABP типа регистрации
nwkskeyabp	BLOB	NwkSKEY для ABP типа регистрации
appskeyabp	BLOB	AppSKEY для ABP типа регистрации
devaddrotaa	INTEGER	DevAddr, сформированный сервером при OTAA типе регистрации
nwkskeyotaa	BLOB	NwkSKEY, сформированный сервером при OTAA типе регистрации
appskeyotaa	BLOB	AppSKEY, сформированный сервером при OTAA типе регистрации
devnonce	INTEGER	Последний DevNonce в JOIN запросе
lastjoints	INTEGER	время последнего JOIN в формате мс, прошедших с начала UNIX эпохи
class	INTEGER	Класс устройства: 1 = «A», 3 = «C»
rxwinnum	INTEGER	Номер используемого приемного окна на устройстве
timingprof	INTEGER	Не используется, может отсутствовать
lastrssi	INTEGER	Последний RSSI в пакете от устройства
lastsnr	INTEGER	Последний SNR в пакете от устройства
lastpower	INTEGER	Мощность передачи устройства
lastsf	INTEGER	Последний SF в пакете от устройства
lastch	INTEGER	Не используется
preferdr	INTEGER	Максимальный DR при работе ADR
preferpower	INTEGER	Не используется
fcntup	INTEGER	Номер последнего принятого пакета
fcntdown	INTEGER	Номер последнего переданного пакета
dutycycle	INTEGER	Не используется

chanmask	INTEGER	16-битная маска каналов
battery	INTEGER	Не используется
adr	INTEGER	Флаг разрешения использовать ADR (устройство)
macbs	TEXT	Не используется
lastdatats	INTEGER	Время последней активности устройства в мс
longitude	REAL / DOUBLE	Координата устройства – долгота
latitude	REAL / DOUBLE	Координата устройства – широта
altitude	INTEGER	Координата устройства – высота на местности
freqplan	INTEGER	Не используется, может отсутствовать
reactiontime	INTEGER	Время реакции устройства
lastactivatside	INTEGER	Тип активации, используемый в последнем принятом пакете
rx1delay	INTEGER	Задержка открытия первого приемного окна, с
rx2delay	INTEGER	Задержка открытия второго приемного окна, с
join1delay	INTEGER	Задержка открытия первого приемного при JOIN, с
join2delay	INTEGER	Задержка открытия второго приемного при JOIN, с
rx2dr	INTEGER	DR второго приемного окна на устройстве
rx2freq	INTEGER	Частота второго приемного окна, Гц
freq4	INTEGER	Частота канала №4, Гц
freq5	INTEGER	Частота канала №5, Гц
freq6	INTEGER	Частота канала №6, Гц
freq7	INTEGER	Частота канала №7, Гц
freq8	INTEGER	Частота канала №8, Гц
usedownqueue	INTEGER	Флаг использования очереди пакетов на отправку
adrthrescounter	INTEGER	Буферный параметр
serveradrenable	INTEGER	Флаг разрешения использовать ADR (сервер)

Приложение Б. Структура и состав основных сообщений в консоли

ПО IOT Vega Server в процессе работы выводит в консоль некоторую оперативную информацию. Объем выводимой информации зависит от соответствующих настроек сервера (см. описание конфигурационного файла).

В консоли могут появляться сообщения с отладочной информацией, состав которых может меняться. Но основные типы сообщений неизменны. Ниже приведены описания их структуры и состава.

1. **Сообщение о неподдерживаемой версии протокола, используемого в ПО PacketForwarder на БС.**

Сообщение выделено красным цветом и начинается со строки «**WARNING! Skip gateway msg, invalid protocol version:**». Далее идет информация о IP адресе БС и используемой версии протокола;

2. **Сообщение об ошибке, возникшей при передаче пакета на устройство через БС.**

Сообщение выделено сиреневым цветом и имеет формат: «**<< %1 | TX_ACK %2 | %3**», где

%1 – ID базовой станции;
%2 – время приема ошибки;
%3 – код ошибки в виде строки.

Возможные коды ошибок:

- «**TOO_LARGE_GW_PING_ERR**» - слишком большой пинг до БС;
- «**COLLISION_ERR**» - указанное в пакете время отправки уже занято другим сообщением;
- «**POWER_ERR**» - в последнем пакете для отправки указано некорректное значение мощности передачи;
- «**FREQ_ERR**» - в последнем пакете для отправки указано некорректное значение частоты передачи;
- «**NO_VACANT_GW**» - нет свободной БС;
- «**PAYLOAD_SIZE_ERR**» - невозможно отправить пакет указанной длины.

3. **Сообщение о принятом JOIN запросе.**

Цвет сообщения жёлтый, формат сообщения: «**>> GW-%1: JOIN_REQ | %2 | %3 | %4 | SF%5 | RSSI:%6 | %7 | %8**», где

%1 – ID базовой станции;
%2 – ID (DevEUI) устройства;
%3 – дата и время приема сообщения, с точностью до мс;
%4 – частота в МГц, с точностью до 100 кГц;
%5 – значение SF;
%6 – значение RSSI;
%7 – значение SNR;
%8 – результат обработки сообщения. Может принимать одно из следующих значений:

- «**VALIDATED**» - запрос обработан корректно, будет отправлен ответ;

- «**UNKNOWN DEVICE | DEVNONCE=0x%1**» - неизвестное устройство, где **%1** – это используемый в пакете DevNonce в HEX;
- «**REPETITION DEVNONCE**» - прием повторного запроса;
- «**INVALID APPEUI**» - значение AppEUI из запроса не совпадает с тем, что указано при регистрации устройства.

4. Сообщение об отправленном ответе на JOIN запрос.

Голубой цвет текста. Формат сообщения: «**>> GW-%1: JOIN_REQ | %2 | %3 | %4 | SF%5 | VALIDATED**», где

%1 – ID базовой станции;
%2 – ID (DevEUI) устройства;
%3 – дата и время приема сообщения, с точностью до мс;
%4 – частота в МГц, с точностью до 100 кГц;
%5 – значение SF.

5. Сообщение о принятом пакете от устройства.

Зеленый цвет текста. Формат сообщения: «**>> GW-%1: %2 | %3 | %4 | %5 | SF%6 | RSSI:%7 | %8 | CNT:%9 | PORT:%10 | %11**», где

%1 – ID базовой станции;
%2 – тип пакета. Используется «**CONF_UP**» или «**UNCOF_UP**». Если в пакете присутствуют MAC данные, к типу добавляется символы «**+M**»;
%3 – ID (DevEUI) устройства;
%4 – дата и время приема сообщения, с точностью до мс;
%5 – частота в МГц, с точностью до 100 кГц;
%6 – значение SF;
%7 – значение RSSI;
%8 – значение SNR;
%9 – номер пакета;
%10 – номер порта;
%11 – дополнительная информация. Может содержать следующую информацию:

- если пакет повторный от устройства, данное поле содержит код «**REPETITION PACKET**» и цвет текста сообщения становится синим;
- иначе содержит информацию о размере принятого сообщения в виде «**[%1]**», где **%1** – размер сообщения в байтах. Если дополнительно содержатся MAC данные, то добавляется и их размер в виде « **| M [%1]**», где **%1** – размер MAC данных в байтах.

6. Сообщение об отправленном пакете для устройства.

Фиолетовый цвет текста. Формат сообщения: **«>> GW-%1: %2 | %3 | %4 | %5 | SF%6 | CNT:%7 | PORT:%8 | %9**», где

%1 – ID базовой станции;
%2 – тип пакета. Используется «**CONF_DOWN**» или «**UNCOF_DOWN**». Если в пакете присутствуют MAC данные, к типу добавляется символы **«+M**»;
%3 – ID (DevEUI) устройства;
%4 – дата и время прихода подтверждения об отправке сообщения (не путать с реальным временем отправки);
%5 – частота в МГц, с точностью до 100 кГц;
%6 – значение SF;
%7 – номер пакета;

%8 – номер порта;
%9 – содержит информацию о размере передаваемого сообщения в виде «**[%1]**», где **%1** – размер сообщения в байтах. Если дополнительно содержатся MAC данные, то добавляется и их размер в виде « **| M [%1]**», где **%1** – размер MAC данных в байтах.

7. Сообщение о возникших ошибках при отправке данных.

Красный цвет текста. Формат сообщения: «**>> GW-%1: %2 | %3 | %4 | %5 |**», где

%1 – ID базовой станции;
%2 – тип пакета. Используется «**UNCONF**», «**CONFRM**» или «**UNCKNOWN**». Если в пакете присутствуют MAC данные, к типу добавляется символы «**+M**»;
%3 – ID (DevEUI) устройства;
%4 – дата и время попытки неуспешной отправки;
%5 – код ошибки в виде строки.
Возможные коды ошибок:

- «**TOO_LARGE_GW_PING_ERR**» - слишком большой пинг до БС;
- «**COLLISION_ERR**» - указанное в пакете время отправки уже занято другим сообщением;
- «**POWER_ERR**» - в последнем пакете для отправки указано некорректное значение мощности передачи;
- «**FREQ_ERR**» - в последнем пакете для отправки указано некорректное значение частоты передачи;
- «**NO_VACANT_GW**» - нет свободной БС;
- «**PAYLOAD_SIZE_ERR**» - невозможно отправить пакет указанной длины.

Информация о документе	
Заголовок	IOT Vega Server
Тип документа	Руководство
Номер документа	B02-server-01
Номер и дата последней ревизии	11 от 12.02.2018

Этот документ применим к следующим продуктам:

Тип продукта	Название продукта
Программное обеспечение	IOT Vega Server
	IOT Vega Admin Tool

История ревизий

Ревизия	Дата	Имя	Комментарии
01	05.06.2017	КЕВ	Дата создания документа
02	06.06.2017	МАИ	Мелкие правки
03	16.06.2017	КЕВ	Мелкие правки
04	10.07.2017	МАИ	Добавлен раздел «Установка», исправлен раздел «Настройка»
05	14.08.2017	КЕВ	Мелкие правки
06	31.08.2017	МАИ	Добавлен раздел «IOT Vega AdminTool», скорректирован раздел «Настройка»
07	27.09.2017	МАИ	Мелкие правки в разделе «IOT Vega AdminTool»
08	20.12.2017	МАИ	Изменения в разделе «Настройки»
09	15.01.2018	МАИ	Изменения в разделе «Установка» и «Настройки» в связи с добавлением версии для Linux ARM
10	22.01.2018	БИЮ	Добавлена ссылка на openssl и рекомендации по допустимым символам в settings.conf в разделе «Настройки»
11	12.02.2018	МАИ	Добавлены Приложения А и Б

vega-absolute.ru