

Транзитный GPS-сервер (под Windows и Linux)

Document Title:	Техническое описание GPS-сервера V1.3
Revision:	1.03_RU
Date:	2012-11-06
Status:	Draft
Document Control ID:	Server_V1.3

Назначение:

Сервер слушает заданный TCP+UDP-порты и является посредником между бриджами и клиентами. Сервер работает максимум с 254 бриджами и 16 клиентскими программами.

Принимает данные от бриджей, ведет общий суточный архив в бинарный файл, пересылает информационные пакеты активным клиентам в рабочем режиме, обеспечивает обмен “команда-ответ” между клиентами в командном режиме и бриджами, предоставляет доступ клиентам к суточным файлам архива. Разграничения между клиентами можно производить путем назначения каждой группе клиентов и их бриджей уникального пароля шифрования.

Дополнительно реализован Web-сервис, использующий TCP-порт с тем же номером, что и UDP. Сервис самостоятельно реализует http-протокол и позволяет получить текущую информацию об одном заданном объекте с помощью любого браузера (например, с мобильного устройства) в виде участка карты заданного масштаба с последним полученным расположением трекера и его данными.

Сервер исполнен в вариантах под Win32 и под Linux. На одном VDS (машине) можно запустить большое количество серверов (каждый слушает свой порт и архивирует в свою папку).

Сервер работает в одном потоке, использует около 1К RAM + буфер одного TCP и одного UDP-сокета (unblocked) + буфер одного открытого файла архивации (ab) + буфер открываемого по требованию клиента (чтение архива) файла (rb). Системные вызовы в рабочем цикле: `gettimeofday`, `accept`, `recvfrom`, `recv`.

Ограничения:

- один сервер работает с 254 бриджами максимум;
- один сервер работает с 16 клиентами максимум;
- с одним и тем же бриджем одновременно может работать только один клиент в командном режиме;
- считывать файл архива одновременно может только один клиент;
- получить информацию о трекера с помощью web-сервиса одновременно может только один клиент (обмен происходит однократно в виде GET-запроса и ответа, затем соединение закрывается и сервер готов к обслуживанию следующего Web-клиента, а текущий клиент далее работает только с сервисом GoogleStaticMap).

Настройка:

В ini-файле задается номер TCP+UDP-порта, время деактивации клиента в рабочем режиме (сек), время деактивации клиента в командном режиме (сек), время деактивации клиента в режиме чтения архива (сек)

Конфигурационный файл `gps.ini`:

- номер TCP+UDP-порта
- таймаут клиента в рабочем режиме

- таймаут клиента в командном режиме (Jobber-a)
- таймаут клиента в режиме чтения архива

Логика работы:

Клиентская программа в **рабочем режиме** периодически (обычно 1 раз в 10 сек) отправляет серверу UDP KeepAlive-пакет, после чего сервер регистрирует клиента и в дальнейшем пересылает такому клиенту все информационные пакеты от бриджей. Если в течение времени деактивации (обычно 20 сек) от клиента не поступит очередной KeepAlive, сервер убирает данного клиента из списка активных, прекращая ему рассылку.

Клиент в **Web-режиме** обращается к серверу с помощью любого браузера по ссылке вида:

<http://yourserver.org:4105/B5M2Z15> , где

yourserver.org – доменное имя или IP-адрес хостинга (VDS, компьютера);

4105 – tcp-порт данного демона;

B5 – номер трекера (5);

M2 - тип карты (2);

Z15 – масштаб карты (15);

Параметры B,M и Z могут быть расположены в любом порядке и могут быть опущены (в этом случае, а также при неверном значении параметра используется значение по умолчанию).

B – номер трекера (1-254), по умолчанию последний активный трекер;

M – тип карты (1-дорожная, 2-фото, 3-гибрид, 4-рельеф), по умолчанию 1.

Z – масштаб (5-24), большее значение соответствует более детальной карте, по умолчанию 11 (отображается участок приблизительно 25*25 км).

Сервер возвращает клиенту последнюю информацию о трекаре в виде небольшой html-страницы с Java-скриптами и тот час сессия закрывается, далее браузер автоматически обращается на Google Static Map сервис, отдающий заданную карту заданного масштаба размером 512*512 пикселей.

Заголовок страницы содержит информацию о номере трекара и дате-времени его последней активности; в левом верхнем углу карты расположена панель управления, а также текстовая информация: координаты, скорость, направление движения и топливо. Значок трекара расположен строго в центре карты, точка соответствует острию “капли”. Цвет (желтый/зеленый) определяется состоянием трекара (занят/свободен), метка (точка/кружок) – способом определения координат (GPS/GSM-локация).

При необходимости смены типа карты или масштаба клиент производит соответствующий выбор в панели управления в окне браузера и нажатием кнопки “>>” обновляет изображение.

ВНИМАНИЕ: при обновлении изображений клиент работает только с сервисом Google Static Map, используя данные о трекаре, полученные от сервера при первичном запросе. Для обновления данных о трекаре необходимо повторно запросить сервер, использовав функцию браузера “Обновить”.

Клиентская программа в **командном режиме** использует сервер как транзитный узел для обмена данными с выбранным бриджем. Вначале такой клиент должен отправить на сервер команду “установка командного режима для бриджа ВВ”, сервер отправляет подтверждение “ОК” и после этого ожидает появления указанного бриджа в сети и информирует об этом данного клиента сообщением “бридж в сети”. Время ожидания обычно 200 сек. Клиент может периодически повторять установку командного режима, тем самым перезапуская интервал ожидания.

При получении ответа “бридж в сети” клиент отправляет на сервер команду для бриджа, которую сервер пересылает указанному ранее бриджу. Ответ бриджа сервер пересылает данному клиенту.

ВНИМАНИЕ: только один клиент может включить командный режим для одного и того же бриджа.

Клиентская программа в режиме чтения суточного архива инициирует обмен с сервером отсылкой запроса с указанием даты нужного суточного архива. Получив такой пакет, сервер регистрирует данного клиента в режиме чтения (с таймаутом для режима чтения) или отправляет код ошибки, если в это время другой клиент уже работает с архивом. Затем сервер открывает необходимый файл архива, в случае ошибки отправляет код ошибки клиенту и снимает его с регистрации. При успешном открытии отправляется сообщение о длине файла в байтах. Клиент отправляет пакет-запрос на выдачу данных, указывая позицию файла архива. Сервер обновляет таймаут и в ответ сервер выдает пакет с указанием позиции и блоком данных фиксированной длины (256 байт), или код ошибки, если архив не инициализирован или запрашиваемая позиция находится за пределами файла. Клиент повторяет процедуру до считывания всех данных. При достижении конца файла сервер отправляет клиенту сообщение об окончании. Клиент может прервать чтение, отправив команду отмены задачи, при этом сервер закроет читаемый файл архива, снимет данного клиента с регистрации и будет готов поставить другого клиента в режим чтения.

Файл архива:

суточный файл для архивирования всех поступивших данных от всех бриджей. Имя файла формируется в виде **sYYMMDD.bin** (YYMMDD - год, месяц, дата) в папке **ARH** в рабочей папке сервера. В начале первой секунды новых суток (по системному времени сервера, приведенному к UTC) старый файл закрывается и открывается новый.

Данные в архиве хранятся в виде записей трех типов:

BB CS DH NM XG XH XL YG YH YL VS UU

0<BB<FF

длина 12 байт на запись

последние 8 байт могут шифроваться как блок DES

CS считается как хог всех 12-и байт записи до 0 ПЕРЕД шифрованием

DH NM (5 старш.бит дата, 5 бит часы, 6 мл. бит минуты) формируется из системного времени получения пакета, приведенного к UTH

BB CS NN [DH NM XG XH XL YG YH YL VS UU] [...] –от 1 до 8 точек

0<BB<FF

длина 13+10*NN байт, где NN – количество точек минус 1 (0<NN<7)

последние 8 байт каждой точки могут шифроваться как блок DES

CS считается как хог всех байт пакета до 0 ПЕРЕД шифрованием

DH NM (5 старш.бит дата, 5 бит часы, 6 мл. бит минуты): GPS UTH время фиксации данной точки бриджем в флеш-архиве.

00 BB 00 00 00 CS AA DD DD DD DD XG 00 XH XL YG YH YL VS U1 U2 TT FH FL

длина 24 байта на запись

0<BB<FF, затем три нулевых байта. 0-й и 12-й байт записи - нулевые.

При просмотре архива для определения типа записи клиент ориентируется сначала на байт 0: в записи третьего типа он всегда нулевой (при этом дополнительно проверяются на ноль байты 2, 3, 4 и 12), а затем (если первый байт записи не 0) - на третий байт записи (**NN/DH**): в записи первого типа он всегда больше 7 (т.к. DH не может быть меньше 07: старшие 5 бит (дата) всегда >0), в записи второго типа данный байт определяет к-во точек и не может быть больше 7.

Для каждого бриджа записи одного типа располагаются в файле архива последовательно (в порядке их времени поступления от бриджа), но данные 2-го типа обычно содержат точки, зафиксированные перед предыдущей последней записью 1-го типа (т.к. извлекаются из флеш-архива бриджа при потерях связи и т.п.). Поэтому клиенту рекомендуется упорядочить архив после считывания следующим образом:

С начала архива найти первую запись второго типа для данного бриджа, запомнить позицию и считать ее. Затем с начала архива последовательно сохранять записи первого типа до тех пор, пока их даты меньше первой точки в ранее найденной записи второго типа. Если нет, то сохранять точки из записи второго типа, пока их даты меньше, или пока не исчерпается ее длина. В последнем случае найти следующую запись второго типа, и т.д.

Кроме того, записи третьего типа могут быть информационными (старший бит байта AA установлен, такие записи идут последовательно по времени) и кеш (старший бит байта AA сброшен), такие записи могут идти в любой последовательности и клиент при чтении архива сервера должен отсортировать их по метке времени DD DD DD DD вместе с информационными пакетами.

Т.о. при минимуме использования ресурсов бриджа и сервера, без SQL и т.п. клиент имеет возможность работать с серверным архивом.

ВНИМАНИЕ: поступившие данные заносятся в файл архива без предварительной проверки контрольной суммы в зашифрованном виде (от каждой группы бриджей – своим ключом, неизвестном серверу), поэтому должны проверяться клиентом перед их дальнейшим использованием после считывания архива, выбора своих бриджей и их дешифровки.

Обрабатываемые пакеты:

1. Информационные пакеты от бриджей в формате BIN_S:

BB CS AA XG XH XL YG YH YL VS UU

0<BB<FF

длина 11 байт

последние 8 байт могут шифроваться как блок DES

CS считается как хог всех 11-и байт пакета до 0 ПЕРЕД шифрованием.

Запоминается IP и порт данного бриджа **BB**

Если старший бит **AA=1**, то бриджу отсылается ответ 1 байт **BB**

В неизменном виде пересылаются всем активным клиентам, находящемуся в рабочем режиме.

Кроме того, если один из активных клиентов находится в командном режиме для бриджа **BB**, то ему отсылается **00 BB** (“бридж **BB** в сети”)

Кроме того, данные архивируются в общий суточный файл с именем **sYYMMDD.bin** (YYMMDD - год, месяц, дата) в формате:

BB CS DH NM XG XH XL YG YH YL VS UU

(12 байт на запись), где DH NM (5 старш.бит дата, 5 бит часы, 6 мл. бит минуты) формируется из системного времени, приведенного к UTH

2. Информационные пакеты дочитки архива от бриджей в формате BIN_DS:

BB CS NN [DH NM XG XH XL YG YH YL VS UU] [...] –от 1 до 8 точек

0<BB<FF

длина 13+10*NN байт, где NN – количество точек минус 1 (0<NN<7)

последние 8 байт каждой точки могут шифроваться как блок DES

CS считается как хог всех байт пакета до 0 ПЕРЕД шифрованием

DH NM (5 старш.бит дата, 5 бит часы, 6 мл. бит минуты): GPS UTH время фиксации данной точки бриджем в флеш-архиве.

Запоминается IP и порт данного бриджа **BB**

Бриджу всегда отсылается ответ 1 байт **00**

В неизменном виде пересылаются всем активным клиентам, находящемуся в рабочем режиме.

Кроме того, данные архивируются в общий суточный файл с именем **sYYMMDD.bin** (YYMMDD - год, месяц, дата) в формате поступившего пакета без изменения.

3. Информационные пакеты от бриджей в формате BIN_EE:

BB 00 00 00 CS AA DD DD DD DD XG XH XL YG YH YL VS U1 U2 TT FH FL

Длина 22 байта. Сервер **всегда** подтверждает получение байтом **AA**

BB - идентификатор бриджа (1-254) и затем обязательно три нулевых байта.

CS = хог всех байт до 0.

AA-старший бит-тип пакета (1-очередной, 0-из кеша)

AA - инкрементируемый идентификатор пакета.

DDDDDDDD - 32-битное поле UNIX TimeStamp в секундах (старший байт вначале)

XG – мл. 7 бит -северная широта в градусах (0-89)

XG – – старший бит – метод определения координат (0-GPS, 1-GSM-локация).

YG – восточная долгота в градусах (0-179)

XH XL, YH YL – минуты и доли минут широты и долготы (например 1F 2B = 07979=7.979 минут)

VS-младшие 4 бита определяют скорость движения из расчета $3+11*(x-1)$ км/ч (при $x=0$ скорость =0).

VS-средние 3 бита определяют вектор направления движения из расчета $45*x$ градусов (если скорость 0, то значение вектора должно игнорироваться)

VS-старший бит определяет состояние объекта (0-свободен, 1-занят)

U1 – младшие 7 бит - значение от АЦП1 (ADC) в % (0-127). Значение 127 передается в первом пакете после перезагрузки трекера.

U1 – старший бит – состояние цифрового входа зажигания (Motor).

U2 – младшие 7 бит - значение от АЦП2 (BAT) в % (0-127). Значение 127 передается в первом пакете после перезапуска GPRS-соединения.

U2 – старший бит – информация о состоянии цифрового выхода (блокировка двигателя).
TT - значение температуры (-127 +127 в формате char) от датчика топлива
FH, FL - старший и младший биты уровня топлива (short) от цифрового датчика: (0-1000) или отрицательный код ошибки. Датчик имеет возможность хардверного тарирования в литры, но на сервисах обычно предусматривают возможность индивидуального софтового тарирования по каждому клиенту.

0<BB<FE

длина 22 байта

два блока по 8 байт (начиная с байта 6) могут шифроваться как блоки DES
CS считается как хог всех 22-х байт пакета до 0 ПЕРЕД шифрованием.

Бриджу всегда отсылается ответ 1 байт **AA**

Если старший бит байта **AA** установлен, то полученный пакет в неизменном виде пересылаются всем активным клиентам, находящемуся в рабочем режиме.

Кроме того, данные архивируются в общий суточный файл с именем **sYYMMDD.bin** (YYMMDD - год, месяц, дата) в формате:

00 BB 00 00 00 CS AA DD DD DD DD XG 00 XH XL YG YH YL VS U1 U2 TT FH FL
(24 байт на запись, байты 0 и 12 обязательно нулевые).

Если **0<BB<FE** и далее следуют три нулевых байта, то данный бридж считается как бридж **BB**: запоминается IP и порт данного бриджа **BB** для командного режима, а также данные точки для отображения бриджа **BB** браузерными клиентами.

Кроме того, в этом случае, если один из активных клиентов находится в командном режиме для бриджа **BB**, то ему отсылается **00 BB** (“бридж **BB** в сети”)

4. Мультикеш пакеты от бриджей в формате BIN_EE:

BB 00 00 00 CS AA [DD DD DD DD XG XH XL YG YH YL VS U1 U2 TT FH FL] ...

Длина $6+n*16$ байт, где $n=2-24$ (количество точек в пакете). Сервер **всегда** подтверждает получение байтом **AA**. Данный пакет клиентам не пересылается, но разбивается на n записей и архивируется в формат архива BIN_EE (по 24 байта на запись), байт CS подсчитывается отдельно для каждой записи.

5. Пакеты от клиентов для управления сервером:

первый байт **FF**, длина 3 байта

FF FF FF - KeepAlive. Если IP-порт клиента, от которого поступил пакет, уже имеется в записях (максимум 16 клиентов), то перезапускается интервал деактивации и устанавливается рабочий режим для данного клиента, если нет - добавляется клиент.
В ответ посылается такой же пакет: **FF FF FF**

FF 00 FF - flush.

Записывает буфер текущего файла архива на диск, клиенту отсылается системная дата-время сервера (**FF DH NM**).

Данная команда применяется перед загрузкой файла архива клиентом и позволяет клиенту ориентироваться, какой файл он грузит: текущий (будет дополняться на сервере в дальнейшем) или старый (может быть сохранен на локальном диске для повторного использования).

FF FF 00 - сброс командного режима для данного клиента. Переводит клиента в НЕАКТИВНЫЙ режим. Если клиент был в режиме чтения, и ранее был открыт файл архива, то он закрывается. Клиенту отсылается подтверждение *Cancel* (**00 00**)

FF CS BB - Если ни один из других клиентов не находится в командном режиме для данного бриджа, то устанавливается командный режим для данного клиента для бриджа **BB** (ожидание появления данного бриджа в сети), клиенту отсылается подтверждение *OK* (**00 FF**), иначе отсылается *Cancel* (**00 00**)

01 YY MM DD – инициализация режима чтения файла архива. Если один из клиентов уже работает в режиме чтения (командном режиме для виртуального бриджа FF), то отсылается сообщение *Cancel* (**00 00**), иначе открывается файл архива **sYMMDD.bin** для чтения. При успешном открытии сервер возвращает длину файла в байтах **XX XX XX** (сначала старший байт) в пакете **FF XX XX XX** и устанавливается режим чтения для данного клиента (с заданным таймаутом), иначе возвращается **FF 00 00 00** (файл не найден).

02 KK KK KK – чтение 256 байт данных с ранее открытого файла архива с позиции **KK KK KK** (сначала старший байт). Если файл архива не был открыт, то возвращает **FF 00 00 00**. Если позиция чтения больше длины файла, то возвращает **FF FF FF FF**.

Иначе обновляется таймаут и возвращает пакет фиксированной длины 264 байта:

LL XX XX XX CC CC CC CC [256 байт с файла архива], где **LL** – длина полезных данных минус 1, **XXXXXX** – запрошенная позиция начала передаваемых данных из файла архива (сначала старший байт), **CC CC CC CC** – crc32 от **LL**-1 байт данных (сначала старший байт). Если длина файла архива меньше запрошенной позиции плюс 256 байт, то **LL<FF**, а оставшееся место в 256-блоке данных заполняется нулями.

6. Пакеты от клиентов для управления бриджами:

первый байт FF, длина больше 3-х байт.

Если для клиента-отправителя ранее был активирован командный режим для определенного бриджа, то от полученного пакета отбрасываются первые два байта, и урезанный пакет пересылается данному бриджу (на IP-порт его последней активности).

7. Пакеты от бриджей клиентам в командном режиме (ответы на команды):

первый байт 00

По IP-порту отправителя определяется номер бриджа (сравнивается с данными последней активности). Если имеются активный клиент в командном режиме для данного бриджа, то пакет пересылается ему в неизменном виде.