

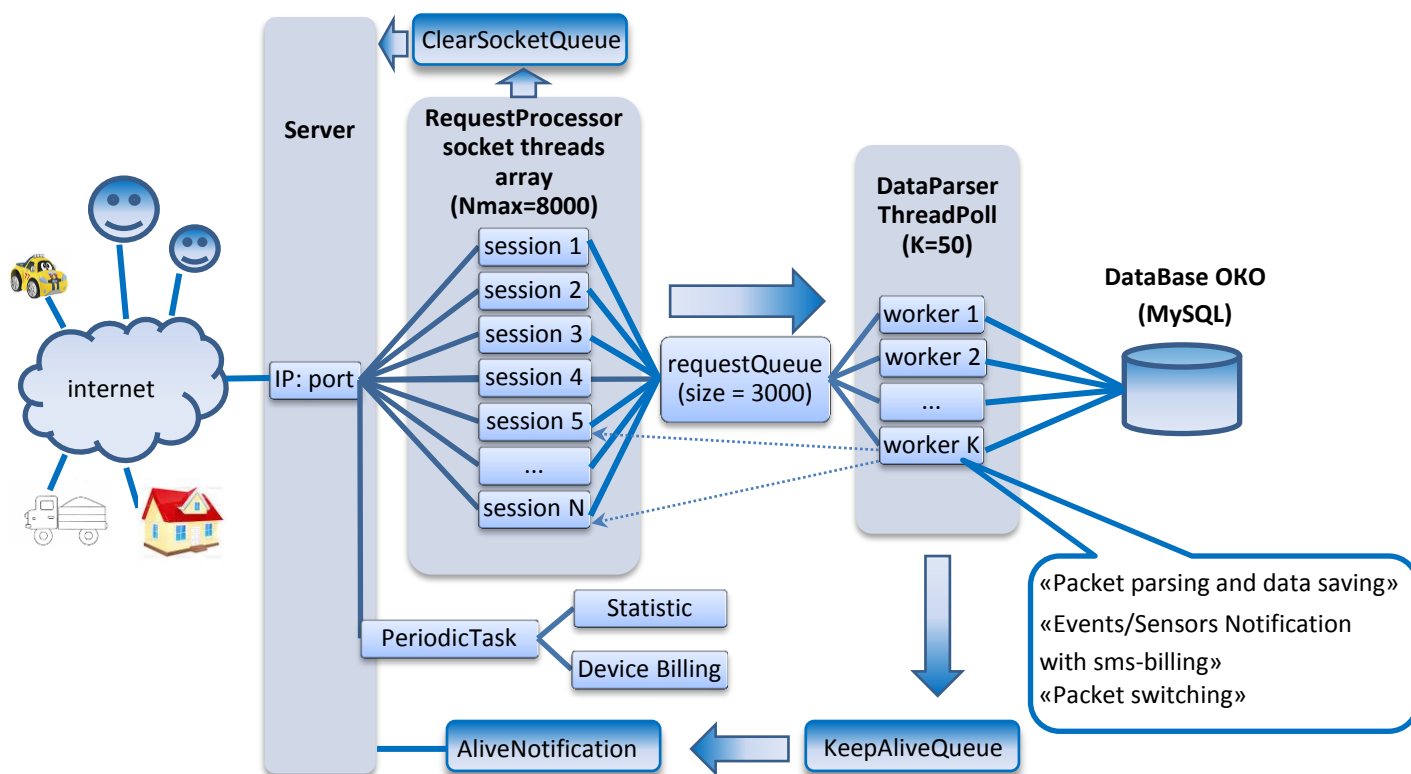
Описание «TCP-сервера ОКО»

TCP-сервер выполняет следующие функции:

- 1) Обеспечивает установление связи с «клиентом» (прибор, приложение), инициатор связи – «клиент». По умолчанию «сервисный» порт равен 31200. Используется **TCP-протокол** в качестве транспортного уровня.
- 2) Обеспечивает одновременное соединения сессий со многим «клиентами» и одновременную обработку их данных. Программное ограничение - 8000 сессий (соединений), которое задается параметром-константой.
- 3) Обеспечивает сохранение поступивших данных от «клиентов» в базу данных ОКО (БД ОКО). Используется СУБД **MySQL**.
- 4) Обеспечивает «коммутацию» пакетов с данными между соответствующими «клиентами» (прибор-приложение).
- 5) Обеспечивает ежесуточный «биллинг» всех учетных записей системы, учитывая количество «активированных» объектов и «ценовую» модель учетной записи.
- 6) Обеспечивает анализ поступивших данных в реальном времени с целью «Оповещения» пользователей через «sms» (используется платный смс-шлюз) и/или через «email». «Триггеры», «условия» и «способы» оповещения ежеминутно обновляются из БД ОКО. Производится «биллинг» в реальном времени соответствующих учетных записей в случае использования платных смс для оповещения.
- 7) Обеспечивает сбор и логирование технической/инженерной информации работоспособности, функциональности, в том числе регулярный сбор статистики используемых ключевых ресурсов данного программного обеспечения.

Программный код написан на языке **Java** и скомпилирован в запускаемый файл «**tcp_server.jar**», в папке «**tcp_server_lib**» содержатся его вспомогательные библиотеки, для отправки оповещения через sms/email используются дополнительные php-скрипты: «**notification_sms.php**», «**notification_email.php**» и др.

СТРУКТУРНАЯ БЛОК-СХЕМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ «ТСР-СЕРВЕРА ОКО»



Назначение структурных блоков программного обеспечения «ТСР-сервера ОКО»

- 1) Стартовый класс «**Server**» обеспечивает «прием» новых соединений «клиентов» на сервисном порте и, в случае появления такого, производит запуск в отдельном «поток» нового процесса «**RequestProcessor**», который в дальнейшем занимается сеансом связи (session) с этим новым «клиентом». Класс «**Server**» ведет учет «открытых» сессий в виде массива активных процессов «**RequestProcessor**», максимальный размер которого ограничен программным способом в размере 8000 (можно изменить соответствующую константу). Все новые входящие соединения «клиентов» свыше данного ограничения «**Server**» сразу же принудительно закрывает, то есть происходит «отказ» в обслуживании «клиентов». Класс «**Server**» при своем старте также запускает «пул потоков» с классом «**DataParser**», который занимается «глубоким» анализом и сохранением данных «клиентов» в БД. При старте «**Server**» также запускается в отдельном «поток» процедура «**PeriodicTask**», которая занимается сбором 15-минутной статистики основных показателей ресурсов ПО в виде лог-файла **statistic.txt**, а также ежесуточным обновлением показаний одометра и «биллингом» учетных записей, исходя из количества активированных приборов, и с автоматической нотификацией на email учетной записи в случае низкого баланса. При старте «**Server**» также запускается в отдельном «поток» класс «**AliveNotification**», который производит анализ необходимости «Оповещения» по триггеру «Отсутствие/Появление» данных от прибора, согласно ежеминутно обновляемому списку «триггеров» контроля связи и настроек оповещений. Пакеты «живучести» приборов этот класс получает из очереди «**KeepAliveQueue**». В случае выполнения необходимых условий производится оповещение на e-mail, используя php-скрипт

«*notification_email.php*» и/или в виде смс, используя php-скрипт «*notification_sms.php*». В случае оповещения через смс «поток» производит «биллинг» соответствующей учетной записи системы.

- 2) Класс «*RequestProcessor*» запускается процессом «*Server*» в отдельном «поток» при каждой новой «сессии». Этот «поток» занимается приемом/передачей данных от/на своего «клиента». Все активные «поток» с сессиями складывают принятые пакеты «клиентов» в общую «блокирующуюся» FIFO-очередь «*requestQueue*». «*RequestProcessor*» также определяет «код» (IMEI) своего «клиента» и его «роль» (прибор, приложение) – эта информация далее используется классами «*Server*» и «*DataParser*» при процессе «коммутации» пакетов с данными между соответствующими «клиентами» (прибор-приложение). Сессия закрывается самим же процессом «*RequestProcessor*» с последующим закрытием своего «поток» по истечению времени ожидания данных от «клиента» или «обрыва» сеанса связи по какой-либо причине. Время ожидания равно 720 секунд и можно изменить соответствующей константой. Перед «закрытием» поток «сообщает» классу «*Server*» свой «идентификатор» через общую очередь закрывающихся сессий «*ClearSocketQueue*».
- 3) При старте класса «*Server*» запускается «пул потоков» с классом «*DataParser*», размер которого задается параметром-константой и по умолчанию равен 50. Каждый «поток» с классом «*DataParser*» имеет свое постоянное соединение с БД ОКО. Каждый «поток» забирает принятые данные «клиентов» из общей очереди «*requestQueue*», обрабатывает их (производит «парсинг» данных с автоматическим опознаванием типа «клиента») и сохраняет в БД ОКО. Также «*DataParser*» определяет необходимую «коммутацию» пакетов данных между соответствующими «клиентами», используя массив активных процессов «*RequestProcessor*», передавая необходимой сессии «*RequestProcessor*» пакет данных для «клиента». Каждый поток «*DataParser*» также производит анализ необходимости «Оповещения», согласно ежеминутно обновляемому списку «триггеров» (События, Сенсоры) и настроек оповещений. В случае выполнения необходимых условий производится оповещение на e-mail, используя php-скрипт «*notification_email.php*» и/или в виде смс, используя php-скрипт «*notification_sms.php*». В случае оповещения через смс «поток» производит «биллинг» соответствующей учетной записи системы. Каждый поток «*DataParser*» также производит анализ необходимости передачи пакетов «живучести» контролируемых приборов в очередь «*KeepAliveQueue*».