

Описание протокола для взаимодействия  
между собой системы из терминалов и  
АРМ.

Андриенко Евгений

02.08.2009

Создаваемый сервер должен координировать совместную работу некоторого заранее неизвестного количества АРМ<sup>1</sup> и терминалов. В создаваемом прикладном протоколе необходимо реализовать рассылку пакетов:

- По адресу – адрес терминала(АРМ) определяется парой значений IP\_address:port
- Всем доступным получателям (broadcasting).

Список адресов доступных терминалов/АРМ заранее определяется в конфигурационном файле программ для АРМ и терминала. Эти же адреса сохраняются при подключении клиента во внутренних структурах данных сервера: при помощи них проверяется доступность клиента при попытке отослать ему данные.

Посылаемые пакеты с данными должны быть зашифрованы методом шифрования с открытым ключом<sup>2</sup>. У АРМ есть закрытый ключ, которым они расшифровывают сообщения от терминалов, а у терминалов – открытый ключ, которым они зашифровывают данные для АРМ. И для пакетов с данными также должна быть реализована проверка контрольной суммы.

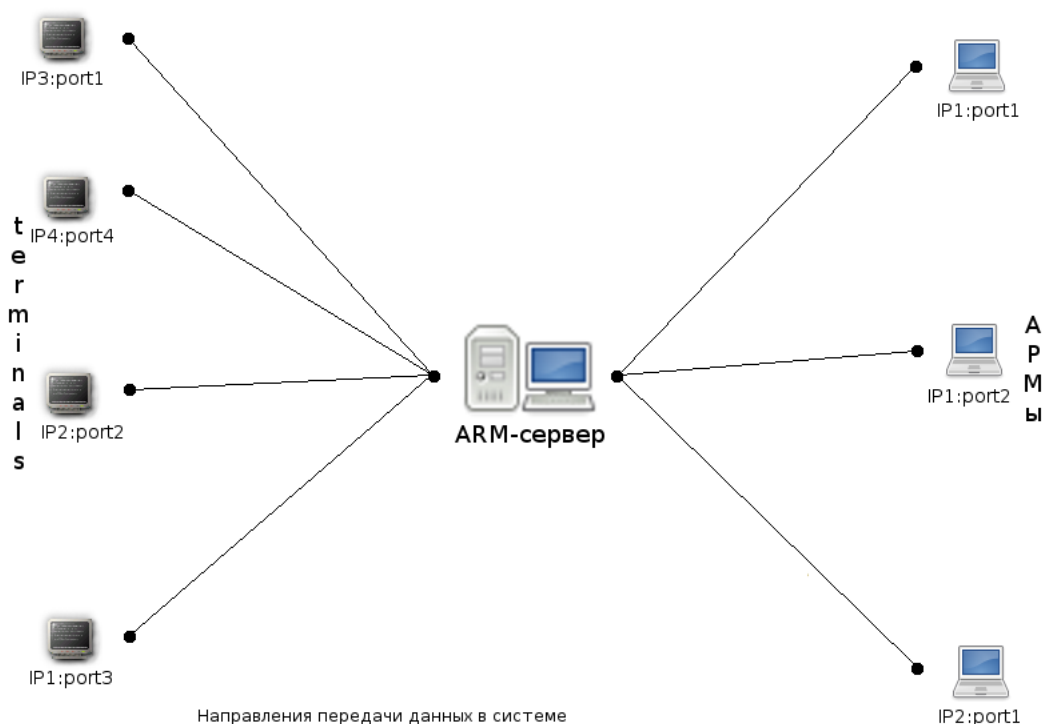
Максимально возможный размер пакета: 512 Мб.

Все двух- и четырехбайтовые поля в пакете (порт, размер блока данных) должны использовать сетевой порядок байтов.

---

<sup>1</sup>Автоматизированное Рабочее Место

<sup>2</sup>Можно RSA. Можно использовать библиотеку OpenSSL.  
[http://www.opennet.ru/docs/RUS/use\\_openssl/](http://www.opennet.ru/docs/RUS/use_openssl/)



Типы пакетов:

- сервисные пакеты (могут рассылаться всеми членами сети):
  - keep-alive ping<sup>3</sup>
  - пакет с error code
- пакеты с данными с датчиков (terminal  $\Rightarrow$  APM)
- пакеты с управляющими командами (APM  $\Rightarrow$  terminal)

## Формат пакетов:

Заголовок, общий для всех типов пакетов:

	0	1	2	3	4	5	6
байт типа пакета	IP адрес					port	
	адрес получателя пакета						

<sup>3</sup>Вроде бы что-то вроде этого уже реализовано в протоколе TCP. Если оно меня удовлетворит – выкину keep-alive ping отсюда.

Пример формирования адреса получателя для 192.168.50.1:26000

0	1	2	3	4	5	6
0x–	0xc0	0xa8	0x32	0x01	0x6590	

Возможные байты типа пакета:

**0xff** сервисный пакет

**0xda** пакет с данными

**0xc0** пакет с управляющими командами

Для broadcast-пакетов поля IP:port имеют следующий вид:

- широковещательная рассылка для терминалов:

0	1	2	3	4	5	6
0x–	0x00	0x00	0x00	0x00	0xffff	

- широковещательная рассылка для АРМ:

0	1	2	3	4	5	6
0x–	0x00	0x00	0x00	0x00	0x0000	

## Данные, специфичные для каждого типа пакетов:

### Сервисные пакеты

0	1	2	3	4	5	6	7
IP отправителя				порт от- правителя	error code	0x00	

Для keep-alive ping error code должен быть равным 0x00. Этот пакет должен посылатся сервером клиенту, каждые 10 минут. Далее клиент должен немедленно ответить таким же пакетом серверу. Максимальное время ожидания ответа от клиента – 10 минут. Если ответа нет – клиент должен быть немедленно отключен от сервера.

Остальные возможные значения error code:

**0xss** не совпадает CRC блока данных с CRC, присланной в пакете

**0x71** не совпадает адрес получателя, присланный в пакете, с реальным адресом получателя (используется терминалами и АРМ).

**0x82** адрес отправителя некорректен (не существует).

**0x72** адрес получателя некорректен (не существует).

**0x1c** некорректный код команды.

**0xd0** нулевой размер блока с данными.

**0xff** неизвестная ошибка.

**0xbc** некорректно указана цель широковещательной рассылки пакетов.

**0x27** некорректно указан тип пакета.

**0x01** null-terminate байт не встречен там, где ожидается.

## Пакеты с командами от АРМ

0	1	2	3	4	5	6	7
IP отправителя				порт от- правителя	command code	0x00	

Возможные значения command code:

**0xdc** запросить пакет с данными от терминала.

**0xf2** остановить работу терминала (дальнейшая работа с терминалом невозможна, до его включения).

## Пакеты с данными от терминалов

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
IP отправителя				порт от- правителя	размер блока данных					блок данных		CRC		0x00

Максимальный размер блока данных (указывается размер уже за-криптованного блока, который и посылается в составе пакета): 256 Мб. Содержимое блока данных не является жестко заданным – оно определяется и интерпретируется программным обеспечением терминала и АРМ, что позволяет адаптировать протокол под определенные цели. Контрольная сумма вычисляется по табличному алгоритму CRC16 (наиболее быстрый, см. <http://ru.wikipedia.org/wiki/CRC>). CRC вычисляется у незашифрованного блока данных.

## Приоритеты пакетов

Каждый передаваемый пакет имеет свой приоритет. Пакеты с большим приоритетом, доставляются быстрее из пакетов с одинаковыми адресами доставки. Приоритеты расставляются в зависимости от типа пакета; типы приведены по убыванию их приоритета:

- сервисные сообщения
- данные от терминалов
- команды от АРМ

Два последних пункта могут меняться в зависимости от задачи. Для команд от АРМ приоритеты выставляются в зависимости от кода операции. Максимальный приоритет у кода `0xdc`, затем `0xf2` – это обеспечивает гарантированное исполнение служебных команд до остановки терминала.

## Взаимодействие и алгоритмы работы узлов в сети

### Клиент – терминал

При получении управляющего пакета, клиент выполняет содержащееся в нем действие, например отправляет пакет с данными. При получении сервисного пакета, клиент выполняет меры по коррекции ошибки; как минимум заносит код ошибки в лог. Если получен сервисный пакет с `keep-alive ring`, то немедленно отвечаем отправителю таким же пакетом. Если внезапно пришел пакет с данными – игнорируем его.

Возможный алгоритм работы узла:

1. Подключаемся к серверу.
2. Ждем команду от АРМ. Если команда не пришла в течение `TIME` секунд – прерываемся. Повторяем попытку прочитать команду `NRETRY` раз. Если это так и не получилось, то переходим к следующему пункту.
3. Проверяем буфер отправки данных – если он не пуст – отправляем данные из него по сети.
4. Переходим к пункту #2.
5. Отключаемся от сервера.

## Клиент – АРМ

При получении пакета с данными, клиент выводит полученную информацию на экран пользователю или заносит в лог. При получении сервисного пакета, клиент выполняет меры по коррекции ошибки или, как минимум, выводит информацию об ошибке на экран и заносит ее в лог. При получении управляющего пакета, клиент игнорирует его.

Действия, при получении сервисного пакета с keep-alive ping аналогичны тем же у терминала.

Возможный алгоритм работы узла:

1. Подключаемся к серверу.
2. Проверяем буфер отправки команд. Если он не пуст – отсылаем команды из него терминалам.
3. Ждем данные от терминалов. Если данные не пришли в течение TIME секунд – прерываемся. Повторяем попытку прочесть данные NRETRY раз. Если не получилось – переходим к следующему шагу.  
Значения TIME и NRETRY должны быть достаточно маленькими, чтобы не создавать задержек при отсылке команд.
4. Переходим к пункту #2
5. Отключаемся от сервера.

## Сервер

Сервер должен обеспечивать простую пересылку пакетов между клиентами в сети, руководствуясь указанным в заголовке пакета адресом получателя. Если нужно послать broadcast-пакет, он рассылается всем клиентам указанной группы.

Возможный алгоритм работы сервера:

1. Запускаются 2 прослушивающих сокета – отдельно один для клиента-АРМ и один для клиента-терминала.
2. Подключившиеся клиенты ставятся в соответствующие им очереди на ожидание событий "ready to read" и "ready to write".
3. Сокеты из набора для АРМ проверяются на возникновение события "ready to read". Если оно произошло – читаем полученный пакет в буфер для АРМ.

4. Сокеты из набора для терминалов проверяются на возникновение события "ready to read". Если оно произошло – читаем полученный пакет в буфер для терминалов.
5. Запускаем функции, упорядочивающие пакеты в определенном буфере соответственно их приоритетам.
6. Сокеты из набора для терминалов проверяются на возникновение события "ready to write". Если оно произошло – в соответствующем буфере пакетов ищутся пакеты с соответствующим адресом получателя и отправляются по сети.
7. Сокеты из набора для АРМ проверяются на возникновение события "ready to write". Если оно произошло – в соответствующем буфере пакетов ищутся пакеты с соответствующим адресом получателя и отправляются по сети.
8. Переходим к шагу #2.
9. Закрываем все сокеты, завершаем сервер.

## Работа протокола с точки зрения семиуровневой сетевой модели OSI

Отправкой и получением пакетов занимается сеансовый уровень. Ошибка, произошедшая при получении пакета, на любом из представленных трех уровней, вызывает прекращение обработки пакета и формирование кода ошибки и послылку сервисного пакета с этим кодом по сети.

Сервер, работает на сеансовом уровне.

### Прикладной уровень

Формируем блок данных или код ошибки/команды. Все двух- четырех-байтовые значения должны посылаться в сетевом порядке байтов.

### Представительский уровень

У блока данных, полученного с прикладного уровня, вычисляется CRC, он шифруется и вычисляется его размер. Вся эта информация добавляется в пакет. Код ошибки/команды, полученный с предыдущего уровня, не модифицируется.

Далее, к пакету добавляется адрес отправителя и null-terminate байт.



## **Сеансовый уровень**

К пакету, сформированному на представительском уровне, добавляется адрес получателя пакета и байт типа пакета. Затем, уже сформированный пакет отправляется в сеть.