

## Передача файла по UDP

### Структура пакета

```
struct msg_udp_t {  
    sockaddrin addr; // IP Адрес получателя при отправке или отправителя при приеме  
    size_t size; // Размер того что ниже  
    uint16_t hash; // Хэш того что ниже, размером size – sizeof(hash)  
    uint8_t type; // Тип сообщения. Старшие 4 бита задаются случайно( для изменения хэша)  
    uint8_t data[UDP_MAX_SIZE]; // Данные  
}
```

Отправляется пакет начиная с hash размером size.

### Типы пакетов

MSG\_TYPE\_INIT – инициализация передачи, передается: имя файла, размер файла, ключ шифрования и начало файла.

MSG\_TYPE\_DATA – блок данных (часть файла). Номер блока, данные.

MSG\_TYPE\_DATA\_ASK – блок данных (часть файла) с запросом подтверждения. Номер блока, данные.

MSG\_TYPE\_COMMIT – подтверждение приема. Размер блока, номер последнего принятого, список пропущенных блоков.

### MTU

Все пакеты идут с флагом запрета фрагментации. MTU определяется путем отправки нескольких пакетов MSG\_TYPE\_INIT разных размеров (1472, 1464, 1372, 1272 – это за вычетом UDP заголовка). Для передачи берется MTU того пакета, ответ на который придет первым.

### Протокол обмена

**Отправитель** получает полное имя файла, считает хэш, определяет размер файла и посылает MSG\_TYPE\_INIT. Размер пакета отдельно с каждое MTU. Если не пришло подтверждение от получателя, то делаются повторные попытки с увеличением интервала на 100 мс, т.е. через 100 мс, 200 мс, 300 мс, 400 мс.

**Получатель**, приняв MSG\_TYPE\_INIT, иницирует прием под MTU с размер пакета и отправляет подтверждение. Если прием уже иницирован, то повторное подтверждение отправляется только если размер пакета равен ожидаемому (чтобы не слать лишние подтверждения).

**Отправитель**, получив первое подтверждение, переходит в режим отправки.

**Отправитель** каждые 20 мс отправляет одно сообщение MSG\_TYPE\_DATA\_ASK и окно из N пакетов MSG\_TYPE\_DATA (в каждом очередной блок файла, если есть запрос потерянных то сначала их). Отправка может начаться раньше, если придет подтверждение последнего MSG\_TYPE\_DATA\_ASK. В этом случае отсчет 20 мс начинается с отправки первого пакета.

**Получатель** сохраняет принятые блоки и следит за порядком их поступления. Если получен блок не по порядку, то пропущенные заносятся в таблицу пропущенных. При получении ранее пропущенного он убирается из таблицы.

**Получатель** отправляет подтверждение MSG\_TYPE\_COMMIT (список пропущенных)

**Отправитель** по приходу MSG\_TYPE\_COMMIT вносит неполученные в список пропущенных и изменяет размер окна (N) в зависимости от количества потерянных пакетов.

**Получатель** при получении последнего блока сохраняет файл, проверяет хэш, отправляет MSG\_TYPE\_COMMIT для извещения об успешном окончании приема. Переходит в состояние ожидания следующего файла (MSG\_TYPE\_INIT). На MSG\_TYPE\_DATA\_ASK отвечает повтором последнего MSG\_TYPE\_COMMIT.

**Отсутствие активности другой стороны в течении 1.5 сек означает неудачу и обмен прерывается.**

**Формат блока данных**

Для MSG\_TYPE\_DATA и MSG\_TYPE\_DATA\_ASK msg\_udp\_t::data содержат номер блока данных и сами данные этого размера. Номер блока записывается в соответствии с таблицей

Номер блока	Представление
0...127	1 байт как есть
128...32767	2 байта (0x8000   номер)
32768...2097151	3 байта (0xC00000   номер)
2097152...268435455	4 байта (0xE0000000   номер)
268435456...68719476735	5 байт (0xF000000000   номер)

В оставшееся место пишутся данные.