

УДК 004.773

РЕАЛИЗАЦИЯ И РАЗВЕРТЫВАНИЕ СЛУЖБЫ МГНОВЕННЫХ СООБЩЕНИЙ ПО ДЕЦЕНТРАЛИЗОВАННОМУ ПРОТОКОЛУ

Бабкин А.А.

*Донецкий национальный технический университет,
кафедра компьютерной инженерии
E-mail: pingvinchick@yandex.ru*

В статье рассматривается протокол децентрализованной службы мгновенных сообщений, возможная реализация необходимого программного обеспечения, а также указан порядок настройки сервиса.

Общая постановка проблемы

В данной статье рассматриваются протокол децентрализованной службы мгновенных сообщений (DMP – Decentralized message protocol), способ его кроссплатформенной реализации, а также развертывание необходимого ПО для запуска сервиса.

Решение поставленной задачи

Описание протокола

Протокол DMP оперирует пакетами фиксированного формата. Структура пакета представлена на рис. 1.

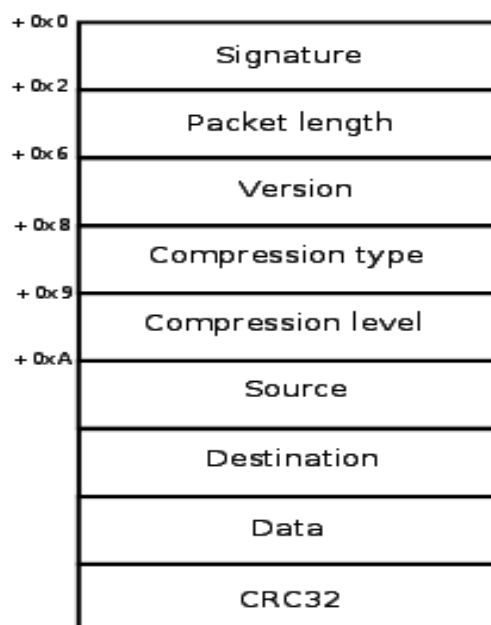


Рисунок 1 - Структура пакета протокола DMP

Описание полей:

1. Signature (2 байта) – признак пакета DMP, равен 0xA829

2. Packet length (4 байта) — общая длина пакета
3. Version (2 байта) – версия протокола
4. Compression type (1 байт) — вид компрессии пакета. Поддерживаются следующие виды сжатия RAW ZIP, GZIP, Zlib. Сжимается только поле Data.
5. Compression level (1 байт) – степень сжатия согласно стандарту компрессии. (От 0 -без сжатия до 9 — максимальное сжатие).
6. Source (переменная длина) — отправитель пакета. Формат: src:<name@server>. Для указания того, что отправителем является сервер в имени пользователя ставится * (звезда).
7. Destination (переменная длина) — получатель пакета. Формат: dst:<name@server>. Для указания того, что получателем является сервер в имени пользователя ставится * (звезда).
8. Data (переменная длина) — сжатые данные. Являются командами.
9. CRC32 (4 байта) – контрольная сумма пакета по алгоритму CRC32.

Структура команд

Команды протокола имеют следующую структуру:

COMMAND<[param1]>[<param2>]...[<param N>][param N+1]

После имени команды должны идти символы « < > » с необязательным параметром. Количество параметров не ограничено, но каждый из них должен быть заключен в символы « < > ». Последний параметр может быть не заключен в данные символы.

Таким образом протокол легко расширяем путем добавления новых команд и параметров.

Список команд довольно велик и не будет рассмотрен в данной статье.

Структура имен

Имя имеет следующую структуру: name@server. Имена чувствительны к регистру. Могут содержать символы латинского алфавита и цифры. Общая длина имени не должна превышать 40 символов.

Name – имя пользователя.

Server – имя сервера. Имя сервера не является DNS именем, а внутрипротокольной идентификационной структурой. Таким образом конечный адрес пользователя не зависит от IP/DNS ни пользователя, ни сервера. Что дает возможность гибкой настройки ПО протокола и возможностью развертывания в локальной сети, а также при отсутствии DNS имени.

При адресации вместо имени пользователя может ставиться символ «*», что является признаком того, что пакет адресован серверу. В качестве имени сервера также может быть поставлен динный символ, но только в командах регистрации пользователя на сервере и создания учетной записи.

Возможности протокола

Протокол DMP версии 1.0 кроме передачи текстовых сообщений и передачи файлов предоставляет следующие возможности:

- смену статуса;
- список контактов;
- список видимости;
- черный список;

- разрешение, отзыв авторизации;
- сообщение статуса.

Топология сети

Топологией сети является звездой.

Возможно соединение серверов напрямую, минуя маршрутизатор, таким образом разгружая канал маршрутизатора. Соединение «сервер-сервер» имеет более высокий приоритет, чем соединение «сервер-маршрутизатор».

На рис. 2 показаны корректная организация сети по протоколу DMP. Между серверами с номерами 1 и 2 составлена связь напрямую, таким образом весь трафик адресованный серверу 2 от сервера 1 и наоборот проходит напрямую, минуя маршрутизатор 1. На рис. 2, а) таблица маршрутизации для роутера 1 не требуется. Анализ получателя происходит автоматически. На рис. 2, б) таблица маршрутизации требуется для роутеров с номерами 2 и 3, так как предполагается перенаправление трафика с маршрутизатора на маршрутизатор.

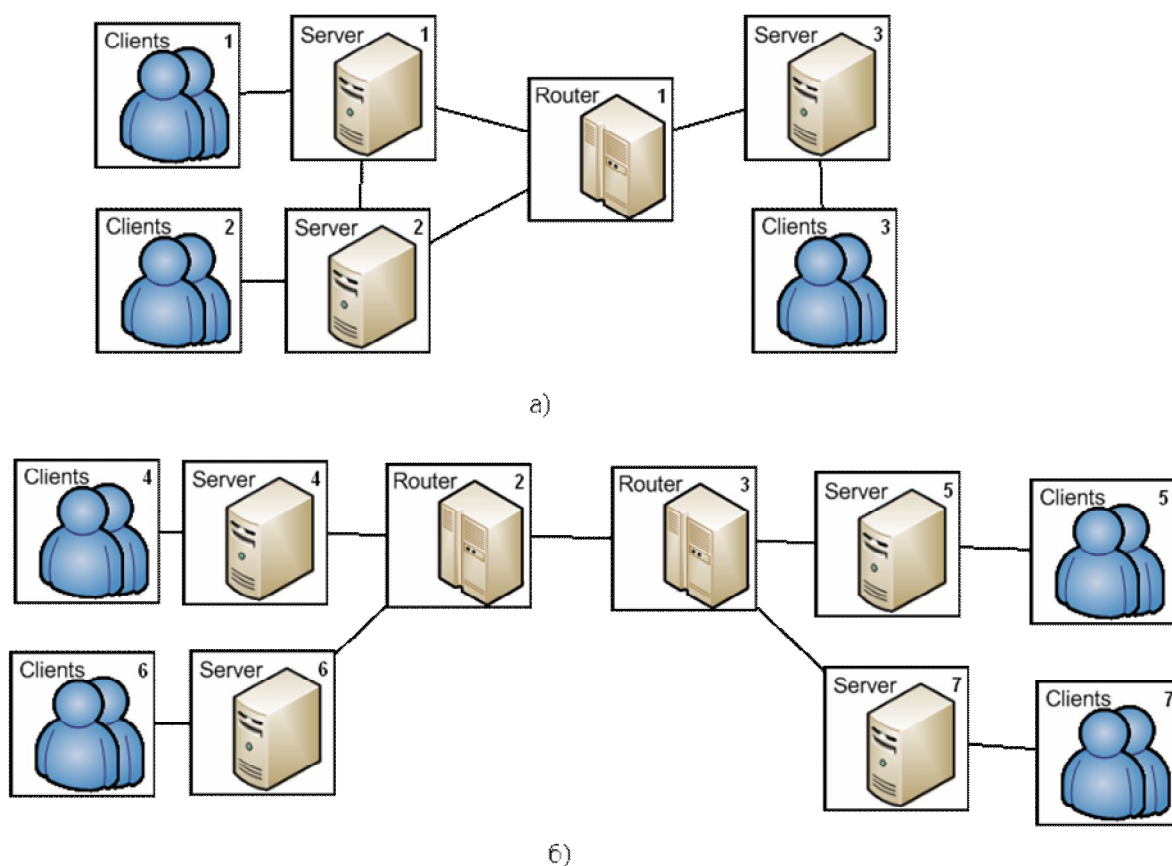


Рисунок 2 – Корректная организация сетей по протоколу DMP

На рис. 3 представлены таблицы маршрутизации для серверов 2 и 3. Для таблиц маршрутизации характерно то, что полях получателей заполнены те серверы, которые не соединены с роутером, для которого составляется таблица маршрутизации. Соединение «маршрутизатор-сервер» имеет более высокий приоритет, чем адрес сервера, полученный из таблицы маршрутизации.

	Destination Server	Gateway (Router)
1	5	3
2	7	3

	Destination Server	Gateway (Router)
1	4	2
2	6	2

Рисунок 3 – Таблицы маршрутизации для серверов 2 и 3

На рис. 4 представлена некорректная DMP сеть. Ошибка заключается в том, что создано кольцо между маршрутизаторами 4–6. При такой организации произойдет перегрузка канала между маршрутизаторами. При отключении устройства от сети (в данном случае любой из серверов 8–11) маршрутизатор выдает по всем интерфейсам команду, информирующую устройства об отключении. При получении такой команду другие маршрутизаторы ее ретранслируют также по всем интерфейсам. Таким образом, при наличии кольца между маршрутизаторами происходит перегрузка канала.

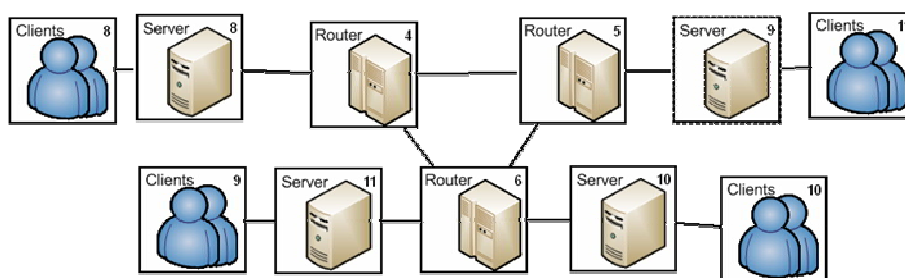


Рисунок 4 – Некорректная DMP сеть

В топологии протокола DMP образование колец между маршрутизаторами недопустимо. Любой сервер может быть подключен только к одному маршрутизатору и к любому количеству других серверов. Маршрутизатор можно подключить к любому количеству серверов.

Структура ПО протокола DMP.

В реализацию протокола входят следующие программные единицы:

- сервер
- маршрутизатор
- клиенты

Сервер осуществляет основные функции протокола. Функционирование сервиса возможно с одним сервером.

Маршрутизатор осуществляет маршрутизацию трафика между серверами на уровне протокола. Таблица маршрутизации заполняется вручную и требуется только в том случае, если требуется перенаправить трафик на другой маршрутизатор.

Сервер и маршрутизатор имеют отдельные потоки для записи/чтения файла конфигурации, файла лога и прослушивания портов.

Кроме того маршрутизатор и сервер имеют базу данных, рассмотренную ниже (рис. 5,6).

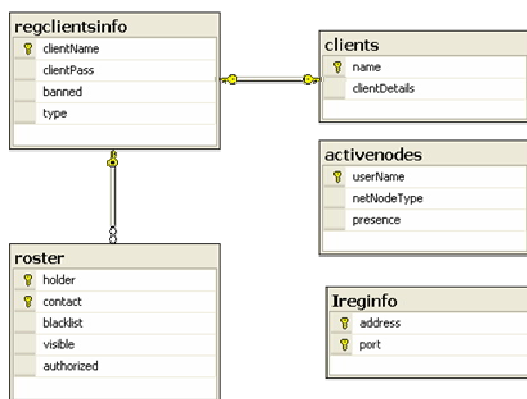


Рисунок 5 – Диаграмма базы данных

На рис. 6 представлены таблицы базы данных сервера. База данных маршрутизатора аналогична, за исключением того, что не имеет таблиц ростера и клиентов, но имеет таблицу маршрутизации.

Имя столбца	Тип данных
clientName	varchar(40)
clientPass	varchar(255)
banned	int
type	int

Имя столбца	Тип данных
userName	varchar(40)
netNodeType	int
presence	int

Имя столбца	Тип данных
holder	varchar(40)
contact	varchar(40)
blacklist	int
visible	int
authorized	int

Имя столбца	Тип данных
address	varchar(50)
port	int

Имя столбца	Тип данных
name	varchar(40)
clientDetails	text

Рисунок 6 – Структура таблиц базы данных

Таблица activenodes содержит все подключенные в данный момент устройства (клиенты, маршрутизатор, серверы). Таблица regclientsinfo содержит имена и пароли устройств, которые могут подключиться к данному серверу (маршрутизатору). Таблица rginfinfo содержит адреса и порты, к которым устройство должно подключиться. Таблица clients содержит персональные данные по клиентам. Таблица roster содержит список контактов каждого пользователя, включая список невидимости, список авторизации, черный список.

ПО написано на C++ с использованием Qt 4.6.3, библиотеки qtioscompressor_2.3.1. Используемая база данных MySQL. С учетом указанных инструментальных средств, ПО является кроссплатформенным.

ПО поддерживает следующие стандарты сжатия трафика: GZIP, Raw ZIP, Zlib.

Настройка ПО

Для корректной работы ПО необходимо правильно заполнить базу данных:

- таблицу маршрутизации

- таблицу регистрации
- таблицу пользователей, которым разрешена регистрация

Кроме того, следует заполнить конфигурационный файл. Пример конфигурационного файла:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
  <database>dmprouter</database>
  <dbuser>root</dbuser>
  <dbpassword>password</dbpassword>
  <dbaddress>localhost</dbaddress>
  <regname>router</regname>
  <regpassword>password</regpassword>
  <port>11656</port>
  <comptype>2</comptype>
  <complevel>6</complevel>
</settings>
```

В файле указаны: имя устройства; пароль устройства; база данных; пароль базы данных; пользователь базы данных; адрес сервера базы данных; порт, который необходимо прослушивать; тип и уровень компрессии.

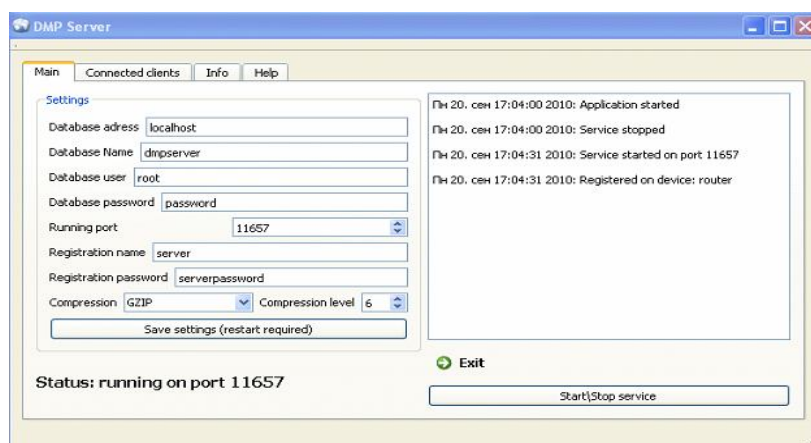


Рисунок 7 – Работающий сервер DMP

Вывод

В данной статье рассмотрены основные возможности и структура протокола децентрализованной службы мгновенных сообщений (DMP). Представлена возможная организация ПО, работающего по данному протоколу, а также рассмотрена настройка данного ПО.

Литература

- [1] Qt Online Reference Documentation [Электронный ресурс]: документация Qt – Режим доступа к ресурсу: <http://doc.qt.nokia.com/>
- [2] MySQL Documentation: MySQL Reference Manuals [Электронный ресурс]: документация MySQL – Режим доступа к ресурсу: <http://dev.mysql.com/doc/>
- [3] QtIOCompressor Class Reference [Электронный ресурс]: документация плагина qtiocompressor – Режим доступа к ресурсу: <http://doc.qt.nokia.com/solutions/4/qtiocompressor/qtiocompressor.html>