

## Proposed Standard for Message Encapsulation

### Предложенный стандарт инкапсуляции сообщений

#### Статус документа

Этот документ описывает протокол, предложенный сообществу ARPA-Internet, и служит приглашением к дискуссии в целях развития и совершенствования протокола. Документ может распространяться без ограничений.

#### Введение, сфера действия и мотивация

Услуги, предлагаемые пользовательскими агентами (UA<sup>1</sup>), могут существенно различаться. Хотя всю исходящую почту можно считать идущей по одному пути соединения с системой доставки сообщений (MTS<sup>2</sup>), можно рассматривать предварительное сообщение (message draft) как один из описанных ниже вариантов.

##### **Originate - исходное сообщение**

Новое сообщение, которое с точки зрения UA не связано с предыдущими сообщениями от пользователя.

##### **Reply - ответ**

Сообщение, созданное в ответ на другое сообщение, полученное пользователем. В большинстве случаев UA помогает пользователю создать ответное сообщение, создавая часть заголовка ответа с использованием заголовков принятого ранее сообщения.

##### **Forward - пересылка**

Одно или несколько сообщений, полученных пользователем, форматируются UA как часть отправляемого сообщения. В этом смысле «дайджест» для группы по интересам может считаться пересылкой. Можно также считать пересылкой «скрытые копии» (BCC<sup>3</sup>).

##### **Distribute - распространение**

Сообщение, полученное ранее пользователем и отправленное снова в систему MTS. Распространяемое сообщение идентично исходному, но содержит заголовки ReSent-XXX, добавленные в конец полей заголовков, а заголовок Return-Path указывает адрес пересылающей стороны (см. [RFC-821], где описан заголовок Return-Path).

Большинство пользовательских агентов поддерживают два первых варианта, многие поддерживают третий, а поддержка всех четырех встречается достаточно редко.

Этот документ рассматривает лишь третий вариант - пересылку сообщений. Краткое рассмотрение семантики компонент, относящихся к откликам, приведено в [RFC-822]. Во многих случаях пересылку можно рассматривать как инкапсуляцию одного или нескольких сообщений в другое сообщение. Хотя пересылка полезна для распространения полученных сообщений другим, без декапсуляции пересланные сообщения будут малополезны, поскольку их нельзя распространить дальше, переслать, ответить исходному отправителю или выполнить иные операции как для обычных сообщений.

Отметим, что в RFC 822 распространение ошибочно считается пересылкой (раздел 4.2). В этом документе разъяснено, что эти действия различаются.

Для дайджестов в группах по интересам возможность пакетной рассылки особенно полезна. Это не только позволяет получателям дайджеста отвечать на отдельные сообщения из «пакета», но и позволяет селективно обрабатывать включенные в дайджест сообщения. Например, один выпуск дайджеста может включать несколько тем и подписчику может быть интересна лишь часть тем из данного выпуска. При использовании пакетов подписчик может разобрать дайджест, просканировать заголовки и обработать интересующие его сообщения, игнорируя остальное.

Мотивами для этого документа послужили три проблемы, указанные ниже.

1. Для сборки сообщений в пакет нужно знать, как инкапсулируются отдельные сообщения. В настоящее время нет однозначного стандарта для дайджестов групп по интересам и данный документ предлагает такой стандарт для ARPA-Internet. Хотя дайджесты групп могут казаться соответствующими псевдостандарту, имеются серьезные расхождения между реализациями, создающими такие дайджесты. Предлагаая стандарт, авторы надеются решить проблему неоднозначности реализаций.
2. Существует путаница в трактовке заголовков BCC агентами UA и кажется, что каждый агент в ARPA-Internet, поддерживающий bcc, делает это по-своему. Данный документ не задает стандарта для генерации bcc, но формализует bcc: как особый случай пересылки.
3. Пересылку и распространение можно реализовать одной процедурой, если распространяемые сообщения можно выделить для индивидуальной обработки. С хорошим агентом пакетизации трудно отличить распространяемое сообщения от сообщения, выделенного из пересылки. Этот документ утверждает, что не следует проводить каких-либо различий между такими сообщениями, а для простоты средства распространения в общем случае не следует предоставлять обычным пользователям. Однако документ также утверждает, что такие средства должны быть доступны некоторым доверенным элементам в составе MTS.

<sup>1</sup>User agent.

<sup>2</sup>Message transport system.

<sup>3</sup>Blind-carbon-copy.

Отметим, что этот документ не предлагает упразднить средства распространения, а скорее призывает к обсуждению этого вопроса других членов сообщества ARPA-Internet.

## Инкапсуляция сообщений

Этот документ предлагает протокол инкапсуляции, в котором два агента выступают от имени пользователя. Агент пересылки собирает предварительное сообщение до его отправки, а агент пакетирования разбирает сообщение после доставки.

Предварительное сообщение для пересылки состоит из заголовка и текста сообщения. При наличии текста он отделяется от заголовка пустой строкой. Внутри текста некоторые последовательности символов, называемые границами инкапсуляции (EB<sup>1</sup>), имеют особое значение. В настоящее время (в имеющихся агентах создания дайджестов), EB определяется как строка, начинающаяся с символа дефиса (-, десятичный код 45). Изначально не задается ограничений на размер границы инкапсуляции или число символов после дефиса.

### 1. Заголовок

Этот документ не задает ограничений для заголовка предварительного сообщения, он просто должен соответствовать стандарту RFC 822.

### 2. Текст

Текст предварительного сообщения состоит из 3 частей - начало, инкапсулированные сообщения и завершение.

#### 2.1. Начало текста

Весь текст (необязательный) до первой последовательности EB образует начальную часть предварительного сообщения. Этот документ не задает ограничений для предварительной части. В случае дайджеста предварительная часть обычно содержит «оглавление».

#### 2.2. Завершение текста

Весь текст (необязательный) после финальной последовательности EB образует завершение предварительного сообщения. Этот документ не задает ограничений для завершающей части. В случае дайджеста завершение обычно содержит подпись (sign-off banner, например «End of FOO Digest»).

#### 2.3. Инкапсулированные сообщения

Каждое инкапсулированное сообщение ограничивается двумя последовательностями EB, одна из которых размещается перед сообщением, другая - после него. Для смежных инкапсулированных сообщений финальная последовательность EB первого является начальной для второго. Поэтому две последовательности EB без текста между ними задают увстраивание пустого сообщения, в результате чего две и более смежных последовательности EB эквивалентны одной EB.

Каждое инкапсулированное сообщение включает заголовок и текст, который отделяется от заголовка пустой строкой.

##### 2.3.1. Заголовок

В каждом пересылаемом сообщении должны присутствовать по меньшей мере поля заголовка Date: и From:. Это отличается от RFC 822, где требуется наличие хотя бы одного адреса получателя (в поле To: или cc:) или (возможно пустого) поля Bcc:. Все адреса в заголовках пересылаемого сообщения должны быть полными.

##### 2.3.2. Текст

Этот документ не задает ограничений на формат текстовой части инкапсулированных сообщений<sup>2</sup>.

Перед рассмотрением инкапсуляции и разбора сообщений следует обсудить две проблемы.

## Совместимость с имеющимися агентами UA

Вышеупомянутый протокол инкапсуляции используется многими UA в ARPA-Internet и был исходно создан для минимизации изменений имеющихся реализаций агентов пересылки в ARPA-Internet. Однако этот протокол не совпадает с псевдостандартом, используемым агентами пересылки, которые собирают дайджесты. В частности, для всех инкапсулируемых в дайджест сообщений финальная последовательность EB окружается пустыми строками. Кроме того, пустая строка помещается после начальной EB в первом инкапсулированном сообщении, но обычно не используется перед ней.

Этот документ рекомендует реализациям агентов пересылки, желающим обеспечить совместимость с имеющимися агентами пакетизации, помещать пустые строки вокруг каждой последовательности EB. Следует отметить, что агенты пакетизации должны игнорировать пустую строку после начальной последовательности EB для инкапсулированного сообщения. Кроме того, документ предлагает им игнорировать также пустые строки перед завершающими EB. Эта сделано в целях совместимости с прежними реализациями. Агентам пересылки, соответствующим этому документу, не следует окружать EB пустыми строками.

## Дополнение EB

Следует отметить, что протокол подходит как для обычной пересылки, так и для дайджестов. К сожалению, имеется проблема инкапсуляции сообщений, которая, по-видимому, не решена ни в одном агенте пересылки (по сведениям авторов) в ARPA-Internet - что делать агенту пересылки при наличии EB в текстовой части пересылаемого сообщения? Все имеющиеся агенты игнорируют это обстоятельство.

<sup>1</sup>Encapsulation boundary.

<sup>2</sup>В действительности такие ограничения имеются и рассмотрены ниже.

Для решения этой проблемы данный документ предлагает схему заполнения EB символами. Граница инкапсуляции определяется как строка, начинающаяся с символа дефиса (-). Особым случаем являются строки EB, в которых после дефиса следует пробел (десятичный код 32).

Если при пересылке агент обнаруживает строку, начинающуюся с дефиса, в тексте пересылаемого сообщения, он добавляет в эту строку после дефиса символ пробела.

Если при пакетировании агент обнаруживает строку, начинающуюся с символов дефиса и пробела, он не считает эту строку границей инкапсуляции и просто выводит оставшуюся часть строки.

Эта простая схема дополнения обеспечивает возможность рекурсивной пересылки.

## Правила создания и разбора инкапсуляции

Правила пересылки и пакетирования описаны регулярными выражениями. Автор вывел простой автомат конечных состояний для правил, но не смог четко представить его в этом документе. Предполагается, что разработчики самостоятельно нарисуют этот автомат для понимания грамматики.

Используемые грамматикой соглашения просты. За каждым состоянием следует один или несколько вариантов, разделенных символом |. Каждый вариант начинается с символа, принимаемого на входе (последовательность CRLF считается одним символом). Последним вариантом для состояния является символ "c", который представляет любой символ, не указанный в предшествующих вариантах. После входного символа может следовать строка, указанная в фигурных скобках. Далее следует состояние, в которое должен перейти автомат. Очевидно, что эта грамматика очень проста в реализации и в большинстве случаев может быть реализована достаточно эффективно.

При инкапсуляции сообщения агенту пересылки следует применить приведенный автомат из начального состояния S1.

```
S1 :: CRLF {CRLF} S1
    | "-" {"- -"} S2
    | c {c} S2

S2 :: CRLF {CRLF} S1
    | c {c} S2
```

Это просто говорит о том, что при наличии символа «-» в начале строки выводится последовательность «-».

При декапсуляции предварительного сообщения агентом пакетирования следует применить приведенный ниже автомат из начального состояния S1.

```
S1 :: "-" S3
    | CRLF {CRLF} S1
    | c {c} S2

S2 :: CRLF {CRLF} S1
    | c {c} S2

S3 :: " " S2
    | c S4

S4 :: CRLF S5
    | c S4

S5 :: CRLF S5
    | c {c} S2
```

Эта грамматика несколько сложнее, но все равно достаточно проста. Предположим для простоты, начальный и завершающий текстовые разделы предварительного сообщения являются сообщениями, дополняющими инкапсулируемые сообщения. Сначала текущее сообщение сканируется из состояния S1. Выводятся все символы, пока не будет найдена первая последовательность EB (состояние S3). При обнаружении «-» автомат переходит в состояние S2 и продолжается вывод символов из текущего сообщения. В конечном итоге находится реальная последовательность EB (состояние S4). Когда автомат переходит из состояния S3 в S4, агенту пакетирования следует считать текущее сообщения законченным. Оставшаяся часть EB отбрасывается (состояния S4 и S5). Когда автомат переходит из S5 в S2, агенту пакетирования следует считать начавшимся новое сообщение и выводить первый символ. В состоянии S2 выводятся все символы до обнаружения EB.

## BCC

Многие пользовательские агенты поддерживают скрытые копии (BCC). В этом случае предварительное сообщение имеет видимых и скрытых получателей. Видимые указаны в полях To: и cc:, а скрытые - в Bcc:. Суть заключается в том, что доставляемые получателям копии предварительного сообщения содержат только видимые адреса.

Одним из способов решения задачи является отправка предварительного сообщения без полей Bcc:, а затем при реальной передаче взаимодействие с MTS для отправки копии видимым и скрытым адресатам. К сожалению с этим способом связана одна проблема. Скрытые получатели могут непреднамеренно ответить на предварительное сообщение так, что видимые получатели будут включены в число получателей ответа, что не приемлемо. Поэтому сообщения для видимых и скрытых адресатов должны различаться.

Другим решением является отправка двух предварительных сообщений. Первое предназначено для видимых адресатов и не содержит поля Bcc:. Второе предназначено для скрытых адресатов и является просто предварительным сообщением, с той или иной строкой перед полями To: и cc:. Например, пользовательский агент может добавить префикс BCC-, чтобы скрытые пользователи получали сообщение с полями BCC-To: и Bcc-cc: вместо To: или cc:. К сожалению это часто создает путаницу для скрытых получателей. Случайные ответы становятся невозможными, зачастую сложно понять, что полученное сообщения является результатом BCC.

Данный документ предлагает отправлять два предварительных сообщения - одно для видимых, другое для скрытых адресатов. Первый вариант содержит исходное предварительное сообщение без полей Bcc:, а второй - видимое

сообщение как пересылаемое. Заголовки для второго включают минимальный набор RFC 822 и при наличии поля Subject: в исходном предварительном сообщении это поле также включается. В дополнение к этому пользовательский агент может явно указать, что сообщение для скрытых получателей является результатом BCC путем включения заголовка Vcc: или перед первой границей инкапсуляции в теле сообщения.

## Распространение сообщений

Основной целью распространения сообщений (часто говорят redistribution или resending) является предоставление копий исходного сообщения другим получателям, которые отсутствовали в числе исходных адресатов. Такое распространение часто выполняют администраторы почтовых конференций, просто пересылая полученные сообщения в список распространения. Лучшие агенты автоматического распространения вставляют новое поле заголовка Return-Path, для прямого возврата сообщений об ошибках доставки лицу, поддерживающему список рассылки, а не исходному отправителю. Такая форма распространения является предпочтительной, поскольку она наиболее точно соответствует задачам доставки сообщений подписчикам почтовых конференций. Рассылка выполняется доверенными агентами псевдо-MTS.

Другой вариант распространения используется лицами, которые хотят переслать пригодную для обработки копию сообщения другим получателям. Эта форма не рекомендуется в новых стандартах передачи сообщений. Такие стандарты включают NBS Standard for Mail Interchange [FIPS-98] и недавний предварительный стандарт CCITT MHS<sup>1</sup> X.400 [X.400]. Вместо прямой пересылки полученного сообщения (как нового предварительного сообщения) рекомендуется пересылать полученное сообщение в теле нового предварительного сообщения, из которого получатель может его извлечь для дальнейшей обработки.

В поддержку этих рекомендаций предложен стандарт инкапсуляции и декапсуляции.

## Литература

- [RFC-822] D.H. Crocker. "Standard for the Format of ARPA-Internet Text Messages", University of Delaware. (August, 1982)
- [RFC-821] J.B. Postel. "Simple Mail Transfer Protocol", USC/Information Sciences Institute. (August, 1982).
- [FIPS-98] National Bureau of Standards. "Specification for Message Format for Computer Based Message Systems." (January, 1983).
- [X.400] Consultative Committee on International Telephone and Telegraph. "DRAFT Recommendation X.400. Message Handling Systems: System Model-Service Elements."

## Адреса авторов

### Marshall T. Rose

Department of Computer and Information Sciences  
University of Delaware  
Newark, DE 19716  
MRose@UDel.ARPA

### Einar A. Stefferud

Network Management Associates, Inc.  
17301 Drey Lane  
Huntington Beach, CA 92647  
Stef@UCI.ARPA

### Перевод на русский язык

Николай Малых

[nmalykh@protokols.ru](mailto:nmalykh@protokols.ru)

<sup>1</sup>Mail Handling Systems - системы обработки сообщений.