

Network Working Group  
Request for Comments: 1701  
Category: Informational

S. Hanks  
NetSmiths, Ltd.  
T. Li  
D. Farinacci  
P. Traina  
cisco Systems  
October 1994

## Базовая инкапсуляция маршрутизации Generic Routing Encapsulation (GRE)

### Статус документа

Этот документ содержит информацию для сообщества Internet. Документ не задаёт каких-либо стандартов Internet и может распространяться свободно.

### Аннотация

Этот документ определяет протокол для инкапсуляции произвольного протокола сетевого уровня в другой произвольный протокол сетевого уровня.

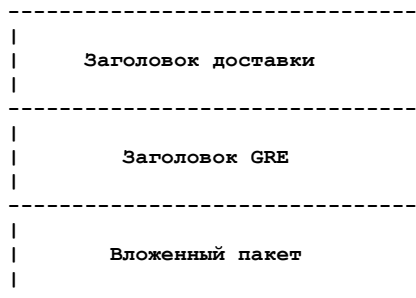
### Введение

В настоящее время существует множество протоколов [RFC 1234, RFC 1226] инкапсуляции одного протокола в другой. Для инкапсуляции IP в IP, требуемой политикой, предложены специальные протоколы [RFC 1241, SDRP, RFC 1479]. В этом документе описан протокол, который очень похож на упомянутые выше протоколы, но является более обобщённым. Для повышения уровня общности многие специфические нюансы протоколов игнорируются. В результате предлагаемый протокол меньше подходит для ситуаций, когда требуется конкретная инкапсуляция протокола X в протокол Y (X over Y). Предлагаемый протокол является попыткой создания простого механизма общего назначения, который переведёт проблему инкапсуляции из текущего состояния  $O(n^2)$  в более управляемое состояние. Протокол также служит попыткой создания облегчённого механизма инкапсуляции для использования в системах маршрутизации на основе правил. Документ не решает в явном виде вопрос определения необходимости инкапсуляции. Документ подтверждает, но не решает проблемы взаимной инкапсуляции [RFC 1326].

В общем случае система имеет пакет, который требуется инкапсулировать и маршрутизировать. Будем называть этот пакет вложенным (payload packet). Вложенный пакет инкапсулируется в пакет GRE, который может также включать маршрут. После этого полученный пакет GRE может инкапсулироваться в тот или иной протокол и пересылается. Этот внешний протокол будем называть протоколом доставки (delivery protocol). Алгоритмы обработки пакета описаны ниже.

### Структура пакета

Инкапсулированный пакет имеет структуру, показанную на рисунке.



### Заголовок пакета

Заголовок пакета GRE имеет форму:

```
      0          1          2          3  
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1  
+-----+-----+-----+-----+-----+-----+-----+-----+  
|C|R|K|S|s|Recur|  Flags  | Ver  |          Protocol Type          |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
|  Checksum (необязательно)  |  Offset (необязательно)  |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
|                               Key (необязательно)                               |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
|                               Sequence Number (необязательно)                               |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
|                               Routing (необязательно)                               |  
+-----+-----+-----+-----+-----+-----+-----+-----+
```

**Флаги и версия (2 октета)**

Флаги GRE размещаются в двух первых октетах. Бит 0 является старшим, а бит 15 - младшим. Биты 13 - 15 зарезервированы для поля Version (версия). Биты 5 - 12 зарезервированы для использования в будущем и должны устанавливаться в 0.

**Checksum Present - C (бит 0)**

Если флаг Checksum Present (контрольная сумма присутствует) установлен, поле Checksum должно присутствовать в заголовке и содержать корректную информацию.

Если установлен любой из битов Checksum Present или Routing Present, **оба** поля Checksum и Offset присутствуют в заголовке пакета GRE.

**Routing Present - R (бит 1)**

Если бит Routing Present (маршрутизация присутствует) имеет значение 1, это говорит о том, что поля Offset и Routing присутствуют в заголовке и содержат корректную информацию.

Если установлен любой из битов Checksum Present или Routing Present, **оба** поля Checksum и Offset присутствуют в заголовке пакета GRE.

**Key Present - K (бит 2)**

Если бит Key Present имеет значение 1, это говорит о наличии поля Key (ключ) в заголовке GRE. В противном случае поле Key в заголовке GRE не включается.

**Sequence Number Present - S (бит 3)**

Если бит Sequence Number Present имеет значение 1, это говорит о присутствии поля Sequence Number (порядковый номер). В противном случае поле Sequence Number в заголовке GRE не включается.

**Strict Source Route - s (бит 4)**

Значение бита Strict Source Route определено в других документах. Рекомендуется устанавливать для этого бита значение 1 только в тех случаях, когда вся маршрутная информация (Routing Information) состоит из маршрутов Strict Source Route.

**Recursion Control (биты 5-7)**

Поле Recursion Control содержит трехбитовое целое число без знака, указывающее допустимое количество дополнительных инкапсуляций. По умолчанию для этого поля **следует** устанавливать значение 0.

**Version Number (биты 13-15)**

Поле Version Number **должно** содержать 0. Другие значения выходят за рамки данного документа.

**Protocol Type (2 октета)**

Поле Protocol Type указывает тип протокола во вложенном пакете. В общем случае это поле будет содержать значение поля типа протокола Ethernet для пакета. Определённые в настоящее время значения типов перечислены ниже. Дополнительные значения поля типа могут быть определены в других документах.

**Offset (2 октета)**

Поле Offset показывает смещение в октетах от начала поля Routing до первого октета активной записи Source Route Entry, которая будет проверяться. Поле присутствует в заголовке, если хотя бы один из битов Routing Present и Checksum Present имеет значение 1; поле содержит корректные сведения лишь при Routing Present = 1.

**Checksum (2 октета)**

Поле Checksum содержит контрольную сумму IP (дополнение до единицы) заголовка GRE и вложенного пакета. Это поле присутствует лишь в тех случаях, когда хотя бы один из битов Routing Present и Checksum Present имеет значение 1; поле содержит корректную информацию лишь при условии Checksum Present = 1.

**Key (4 октета)**

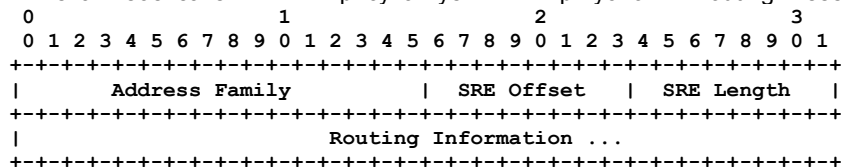
Поле Key содержит 4-октетное значение, которое задаётся при инкапсуляции. Это поле может использоваться получателем для идентификации источника пакета. Методы такой идентификации выходят за рамки настоящего документа. Поле Key присутствует в заголовке лишь при условии Key Present = 1.

**Sequence Number (4 октета)**

Поле порядкового номера содержит 32-битовое целое число без знака, добавляемое при инкапсуляции. Это значение может использоваться получателем для отслеживания порядка передачи пакетов со стороны инкапсулятора. Точный алгоритм генерации порядковых номеров и их семантика выходят за рамки этого документа.

**Routing (переменный размер)**

Поле Routing является необязательным и присутствует лишь при условии Routing Present = 1.



Поле Routing представляет собой список записей SRE<sup>1</sup>. Каждая запись SRE имеет форму, показанную на рисунке. Поле маршрутизации завершается пустой (NULL) записью SRE, содержащей семейство адресов типа 0x0000 и поле размера 0.

**Address Family (2 октета)**

Поле Address Family содержит 2-октетное значение, которое задаёт синтаксис и семантику поля Routing Information. Значения этого поля, а также соответствующий синтаксис и семантика поля Routing Information определяются в других документах.

**SRE Offset (1 октет)**

Поле SRE Offset показывает смещение в октетах от начала поля Routing Information до первого октета активной записи Source Route Entry, которая будет проверяться.

**SRE Length (1 октет)**

Поле SRE Length указывает число октетов в SRE. Если SRE Length = 0, это говорит о том, что данная запись SRE является последней в поле Routing.

**Routing Information (переменный размер)**

Поле Routing Information содержит данные, которые могут использоваться для маршрутизации пакета. Точная семантика поля определяется в других документах.

<sup>1</sup>Source Route Entry.

## Пересылка пакетов GRE

Обычно система, пересылающая пакеты уровня доставки, не будет как-либо отличать пакеты GRE от других пакетов. Однако пакет GRE может быть принят системой. В этом случае системе следует использовать те или иные связанные со способом доставки средства идентификации пакетов GRE. После идентификации такого пакета могут быть проверены значения полей Key, Sequence Number и Checksum, если соответствующие флаги говорят о корректности этих полей. Если бит Routing Present = 1, следует проверить поле Address Family для определения семантики и использования полей SRE Length, SRE Offset и Routing Information. Точная семантика для обработки SRE при разных значениях Address Family определяется в других документах.

После обработки записей SRE заданный отправителем маршрут (source route) будет полным, заголовок GRE следует удалить, значение TTL вложенного пакета (если оно присутствует) **должно** быть уменьшено на 1, а вложенный пакет следует переслать как обычный пакет. Конкретный метод пересылки определяется значением поля Protocol Type.

Семейство протоколов	PType
Резерв	0000
SNA	0004
Сетевой уровень OSI	00FE
PUP	0200
XNS	0600
IP	0800
Chaos	0804
RFC 826 ARP	0806
Frame Relay ARP	0808
VINES	0BAD
VINES Echo	0BAE
VINES Loopback	0BAF
DECnet (Phase IV)	6003
Transparent Ethernet Bridging	6558
Raw Frame Relay	6559
Apollo Domain	8019
Ethertalk (Appletalk)	809B
Novell IPX	8137
RFC 1144 TCP/IP compression	876B
IP Autonomous Systems	876C
Secure Data	876D
Резерв	FFFF

## Текущий список типов протокола

В таблице перечислены определённые к настоящему моменту типы протоколов для GRE. В будущем типы протоколов должны браться из кодирования DIX Ethernet. По историческим причинам для некоторых протоколов используется множество других значений. Приведённая здесь таблица содержит значения, которые **должны** использоваться для указанных в ней протоколов.

Полный список значений типов содержится в реестре IANA<sup>1</sup> для Ether Type.

URL = <ftp://ftp.isi.edu/in-notes/iana/assignments/ethernet-numbers>.

## Литература

- RFC 1479 Steenstrup, M. "Inter-Domain Policy Routing Protocol Specification: Version 1", RFC1479, BBN Systems and Technologies, July 1993.
- RFC 1226 Kantor, B. "Internet Protocol Encapsulation of AX.25 Frames", [RFC 1226](#), University of California, San Diego, May 1991.
- RFC 1234 Provan, D. "Tunneling IPX Traffic through IP Networks", RFC 1234, Novell, Inc., June 1991.
- RFC 1241 Woodburn, R., and D. Mills, "Scheme for an Internet Encapsulation Protocol: Version 1", [RFC 1241](#), SAIC, University of Delaware, July 1991.
- RFC 1326 Tsuchiya, P., "Mutual Encapsulation Considered Dangerous", RFC 1326, Bellcore, May 1992.
- SDRP Estrin, D., Li, T., and Y. Rekhter, "Source Demand Routing Protocol Specification (Version 1)", Work in Progress<sup>2</sup>.
- RFC 1702 Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation over IPv4 networks", [RFC 1702](#), NetSmiths, Ltd., cisco Systems, October 1994.

## Вопросы безопасности

Вопросы безопасности в этом документе не рассматриваются.

## Благодарности

Авторы выражают свою благодарность Yakov Rekhter (IBM) и Deborah Estrin (USC) за их советы, поддержку и ценные комментарии.

<sup>1</sup> Этот реестр поддерживается IEEE и доступен по адресу <http://standards.ieee.org/regauth/ethertype/eth.txt>. Прим. перев.

<sup>2</sup> Работа опубликована в RFC 1940. Прим. перев.

## Адреса авторов

**Stan Hanks**

NetSmiths, Ltd.  
2025 Lincoln Highway  
Edison NJ, 08817  
E-Mail: [stan@netsmiths.com](mailto:stan@netsmiths.com)

**Tony Li**

cisco Systems, Inc.  
1525 O'Brien Drive  
Menlo Park, CA 94025  
E-Mail: [tli@cisco.com](mailto:tli@cisco.com)

**Dino Farinacci**

cisco Systems, Inc.  
1525 O'Brien Drive  
Menlo Park, CA 94025  
E-Mail: [dino@cisco.com](mailto:dino@cisco.com)

**Paul Traina**

cisco Systems, Inc.  
1525 O'Brien Drive  
Menlo Park, CA 94025  
E-Mail: [pst@cisco.com](mailto:pst@cisco.com)

**Перевод на русский язык**

Николай Малых

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