Internet Engineering Task Force (IETF)
Request for Comments: 9484
Updates: 9298
Category: Standards Track
ISSN: 2070-1721

T. Pauly, Ed.
Apple Inc.
D. Schinazi
A. Chernyakhovsky
Google LLC
M. Kühlewind
M. Westerlund
Ericsson

October 2023

Proxying IP in HTTP

Проксирование IP через HTTP

Аннотация

В этом документе описывается проксирование IP-пакетов в HTTP. Протокол аналогичен протоколу проксирования UDP через HTTP, но предназначен для передачи произвольных пакетов IP. Более конкретно, документ задаёт протокол, позволяющий клиенту HTTP создать туннель IP через сервер HTTP, выступающий как IP-прокси. Документ обновляет RFC 9298.

Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF[1] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[2]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке https://www.rfc-editor.org/info/rfc9484.

Оглавление

[1]Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[2]Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

1. Введение

HTTP поддерживает метод CONNECT (параграф 9.3.6 в [HTTP]) для создания туннеля TCP [TCP] с адресатом и аналогичный механизм для UDP [CONNECT-UDP]. Однако эти механизмы не способны туннелировать другие протоколы IP [IANA-PN], а также передавать поля заголовка IP.

В этом документе описывается протокол для туннелирования IP через сервер HTTP, действующий как IP-прокси через HTTP. Это может применяться в разных случаях, например как VPN для удалённого доступа или между сайтами, защищённые связи «точка-точка», туннелирование общего назначения для пакетов.

Проксирование IP работает аналогично проксированию UDP [CONNECT-UDP], при этом сам прокси идентифицируется абсолютным URL, возможно, включающим адресат трафика. Клиенты генерируют такие URL с помощью шаблона URI (URI Template) [TEMPLATE], как описано в разделе 3.

Протокол поддерживает все имеющиеся версии HTTP за счёт применения дейтаграмм HTTP [HTTP-DGRAM]. Для HTTP/2 [HTTP/2] или HTTP/3 [HTTP/3] применяется HTTP Extended CONNECT, как описано в [EXT-CONNECT2] и [EXT-CONNECT3]. Для HTTP/1.x [HTTP/1.1] применяется HTTP Upgrade, как описано в параграфе 7.8 [HTTP].

Этот документ обновляет [CONNECT-UDP] для изменения общеизвестного URI masque (см. параграф 12.3).

2. Соглашения и определения

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

В этом документе термин IP-прокси (IP proxy) обозначает сервер HTTP, отвечающий на запросы проксирования IP. Термин клиент используется как в HTTP, он создаёт запросы на проксирование IP. Посредники HTTP (как определено в параграфе 3.7 [HTTP]) между клиентом и IP-прокси называются просто посредниками (intermediary). Конечными точками проксирования IP (IP proxying endpoint) называются клиенты и IP-прокси.

В документе используется терминология из [QUIC]. При определении в документе типов протоколов применяется формат определений с использованием нотации из параграфа 1.3 в [QUIC]. В спецификации используется кодирование целых чисел с переменным размером из раздела 16 в [QUIC]. Целые числа с переменным размером не требуется кодировать в минимальное число байтов.

Отметим, что в случаях, когда применяемая версия HTTP не поддерживает мультиплексирование потоков (например, HTTP/1.1), термин поток в этом документе относится ко всему соединению.

3. Настройка клиентов

Клиенты настраиваются на использование проксирования IP через HTTP с помощью шаблона URI [TEMPLATE]. Шаблон URI **может** включать 2 переменные - target и ipproto, см. параграф 4.6. Необязательность переменных нужно учитывать при задании шаблона, чтобы переменная была самоопределяемой или её можно было исключить через синтаксис. Примеры приведены ниже.

```
https://example.org/.well-known/masque/ip/{target}/{ipproto}/
https://proxy.example.org:4443/masque/ip?t={target}&i={ipproto}
https://proxy.example.org:4443/masque/ip{?target,ipproto}
https://masque.example.org/?user=bob
```

Рисунок 1. Примеры шаблонов URI.

Далее перечислены требования к шаблону URI.

- URI Template **должен** быть шаблоном не выше уровня 3.
- Шаблон **должен** иметь абсолютную форму и **должен** включать непустые компоненты scheme, authority, и path.
- Компонент path в URI Template **должен** начинаться с символа /.
- Все переменные шаблона **должны** быть внутри компонентов path или query в URI.
- URI Template **может** включать 2 переменные - target и ipproto, а также **может** содержать другие переменные. При наличии target или ipproto пустые значения в них **недопустимы**. Клиенты могут использовать символ * для указания шаблона или значений без предпочтения (см. параграф 4.6).

- В шаблон URI **недопустимо** включать символы non-ASCII Unicode и **должны** применяться только символы ASCII из диапазона 0x21-0x7E, включительно (разрешено %-кодирование, см. параграф 2.1 в [URI]).
- В URI Template **недопустимо** применять Reserved Expansion (оператор +), Fragment Expansion (оператор #), Label Expansion с Dot-Prefix, Path Segment Expansion с Slash-Prefix, Path-Style Parameter Expansion с Semicolon-Prefix.

Клиентам **следует** проверять выполнение указанных выше требований, однако им **можно** использовать реализацию URI Template общего назначения без такой проверки. Если клиент обнаруживает невыполнение любого из указанных выше требований в URI Template, он **должен** отвергнуть конфигурацию и прервать запрос без отправки на IP-прокси.

Как и при проксировании UDP, некоторые конфигурации клиентов для IP-прокси будут разрешать пользователю настраивать только хост и порт прокси. Клиенты с такими ограничениями **могут** пытаться получить доступ к возможностям IP-прокси, используя принятый по умолчанию шаблон, заданный как https://$PROXY_HOST:$PROXY_PORT/.well-known/masque/ip/{target}/{ipproto}/, где $PROXY_HOST и $PROXY_PORT будут иметь настроенные значения для хоста и порта IP-прокси. Внедрениям IP-прокси **следует** предоставлять сервис, если им нужна совместимость с такими клиентами.

4. Туннелирование IP через HTTP

Для обеспечения возможности создания туннеля для IP по протоколу HTTP в этом документе определён маркер обновления HTTP connect-ip. Создаваемые туннели IP используют капсульный протокол (Capsule Protocol, см. параграф 3.2 в [HTTP-DGRAM]) с дейтаграммами HTTP, формат которых задан в разделе 6.

Для инициирования туннеля IP, связанного с одним потоком HTTP, клиент вводит запрос с маркером обновления connect-ip. При отправке запроса на проксирование IP клиенту **нужно** выполнить расширение URI Template для задания пути и своих потребностей (query), как описано в разделе 3.

По определению капсульного протокола (параграф 3.2 в [HTTP-DGRAM]) запросы проксирования IP не могут включать содержимого. Точно так же, отклики об успешном проксировании IP не включают содержимого.

При проксировании IP по протоколу HTTP **должно** применяться шифрование TLS или QUIC (возможен также иной протокол шифрования) для обеспечения конфиденциальности, целостности и аутентификации.

4.1. Работа IP Proxy

Ниже указаны действия при получении запроса на проксирование IP.

- Если получатель настроен на использование другого сервера HTTP, он будет выступать посредником, пересылая запрос другому серверу HTTP. Отметим, что такому посреднику может потребоваться повторное кодирование запроса, если он пересылает с использованием не той версии HTTP, которая указана в полученном запросе, поскольку кодировка запросов зависит от версии (см. ниже).
- В иных случаях получатель будет выступать как IP-прокси, который может принять или отвергнуть запрос. В случае восприятия запроса извлекаются исходные переменные target и ipproto из URI, восстановленного по заголовкам запроса, декодируется %-представление и организуется туннель IP.

IP-прокси **должны** убедиться, что декодированные переменные target и ipproto соответствуют требованиям параграфа 4.6. Если это не так, IP-прокси **должен** считать запрос некорректным (см. параграф 8.1.1 в [HTTP/2] и 4.1.2 в [HTTP/3]). Если переменная target содержит имя DNS, IP-прокси **должен** выполнить распознавание (для получения адреса IPv4 и/или IPv6 по записям A, AAAA) до ответа на запрос HTTP. Если при этом возникает ошибка, IP-прокси **должен** отклонить запрос и **следует** передать детали отказа в подходящем поле заголовка Proxy-Status [PROXY-STATUS]. Например, при возврате ошибки в процессе распознавания DNS прокси может использовать тип ошибки прокси dns_error из параграфа 2.3.2 в [PROXY-STATUS].

Срок действия туннеля для пересылки IP привязан к потоку запросов проксирования IP. IP-прокси **должен** поддерживать все назначения адресов IP и маршрутов, связанные с туннелем пересылки IP, пока поток запроса открыт. IP-прокси **могут** закрывать туннель по истечении срока бездействия, но при этом они **должны** закрыть поток запросов.

Отклик об успешном проксировании IP (см. параграфы 4.3 и 4.5) показывает, что IP-прокси организовал туннель IP и готов передавать данные IP (payload). Любой другой отклик на запрос проксирования говорит об отклонении запроса и клиент **должен** прервать запрос.

Вместе с откликом об успехе IP-прокси может передать клиенту капсулы для назначения адресов и анонсирования маршрутов (параграф 4.7). Клиент также может назначать адреса и анонсировать маршруты IP-прокси для маршрутизации между сетями.

4.2. Запрос HTTP/1.1

При использовании When using /1.1 [HTTP/1.1] к запросам проксирования IP применяется ряд требований.

- **Нужно** использовать метод GET.
- В запрос **нужно** включать 1 поле заголовка Host, включающее хост и необязательный номер порта IP-прокси.
- В запрос **нужно** включать поле заголовка Connection со значением Upgrade (без учёта регистра символов, как указано в параграфе 7.6.1 [HTTP]).
- В запрос **нужно** включать поле заголовка Upgrade со значением connect-ip.

Запросы проксирования IP, не соответствующие этим требованиям считаются некорректными, получатель такого запроса **должен** возвращать ошибку и для отклика **следует** применять код 400 (Bad Request). Например, если у клиента задан шаблон URI https://example.org/.well-known/masque/ip/{target}/{ipproto}/ и клиент хочет создать туннель для пересылки IP без ограничения цели и протоколов, он может передать показанный ниже запрос.

```
GET https://example.org/.well-known/masque/ip/*/*/ HTTP/1.1
Host: example.org
Connection: Upgrade
Upgrade: connect-ip
Capsule-Protocol: ?1
```

Рисунок 2. Пример запроса HTTP/1.1.

4.3. Отклик HTTP/1.1

Сервер отвечает на успешное выполнение запроса проксирования IP откликом, соответствующим ряду требований.

- В отклике **нужно** указывать код статуса HTTP 101 (Switching Protocols).
- В отклик **нужно** включать поле заголовка Connection со значением Upgrade (без учёта регистра символов, как указано в параграфе 7.6.1 [HTTP]).
- В отклик **нужно** включать 1 поле заголовка Upgrade со значением connect-ip.
- **Нужно** соблюдать требования к откликам HTTP, начинающимся с Capsule Protocol (см. параграф 3.2 в [HTTP-DGRAM]).

Если какое-либо из этих требований не выполняется, клиент **должен** считать эту попытку проксирования неудачной и закрывать соединение.

Например, сервер может передать показанный ниже отклик.

```
HTTP/1.1 101 Switching Protocols
Connection: Upgrade
Upgrade: connect-ip
Capsule-Protocol: ?1
```

Рисунок 3. Пример отклика HTTP/1.1.

4.4. Запросы HTTP/2 и HTTP/3

При использовании HTTP/2 [HTTP/2] или r HTTP/3 [HTTP/3] в запросах проксирования IP применяется метод HTTP Extended CONNECT. Этот требует от сервера передавать HTTP Setting, как задано в [EXT-CONNECT2] и [EXT-CONNECT3], а в запросах должны присутствовать поля псевдозаголовка HTTP, соответствующие ряду требований.

- В поле псевдозаголовка :method **нужно** указывать значение CONNECT.
- В поле псевдозаголовка :protocol **нужно** указывать значение connect-ip.
- В поле псевдозаголовка :authority **нужно** указывать значение полномочия IP-прокси.
- Полям псевдозаголовка :path и :scheme **не следует** быть пустыми. В них **нужно** указывать схему и путь из URI Template после преобразования шаблона URI (см. раздел 3). Переменные в URI Template могут определять область действия запроса, например пересылку через туннель всех пакетов IP или проксирование конкретного потока (см. параграф 4.6).

Запрос проксирования IP, не соответствующий этим требованиям, считается некорректным (см. параграф 8.1.1 в [HTTP/2] и 4.1.2 в [HTTP/3]).

Например, если клиент настроен с шаблоном URI https://example.org/.well-known/masque/ip/{target}/{ipproto}/ и хочет создать туннель для пересылки IP без ограничения цели и протоколов, он может передать показанный ниже запрос.

```
HEADERS
:method = CONNECT
:protocol = connect-ip
:scheme = https
:path = /.well-known/masque/ip/*/*/
:authority = example.org
capsule-protocol = ?1
```

Рисунок 4. Пример запроса HTTP/2 или HTTP/3.

4.5. Отклики HTTP/2 и HTTP/3

Сервер отвечает на успешное выполнение запроса проксирования IP откликом, соответствующим ряду требований.

- В отклике **нужно** указывать код статуса 2xx (Successful).
- **Нужно** соблюдать требования к откликам HTTP, начинающимся с Capsule Protocol (см. параграф 3.2 в [HTTP-DGRAM]).

Если какое-либо из этих требований не выполняется, клиент **должен** считать эту попытку проксирования неудачной и прерывать запрос. Например, любой код 3xx будет считаться отказом, заставляя клиента прервать запрос.

Ниже приведён пример возможного отклика сервера.

```
HEADERS
:status = 200
capsule-protocol = ?1
```

Рисунок 5. Пример отклика HTTP/2 или HTTP/3.

4.6. Ограничение области действия запроса

В отличие от запросов проксирования UDP, требующих указания целевого хоста, запросы для IP могут позволять конечным точкам передавать произвольные пакеты IP любому хосту. Клиент может задать ограничения для данного запроса, задав конкретный префикс или протокол IP путём добавления параметров в запрос. Когда IP-прокси знает, что запрос связан с целевым префиксом или протоколом, он может воспользоваться этими сведениями для оптимизации выделения своих ресурсов. Например, IP-прокси может назначить 1 публичный адрес IP для двух запросов на проксирование IP, которые связаны с разными префиксами и/или протоколами.

Область действия запроса клиент указывает IP-прокси через переменные target и ipproto в шаблоне URI (см. раздел 3), которые необязательны и при отсутствии принимается шаблонное значение *.

target

Переменная target содержит имя хоста или префикс IP конкретного хоста, с которым клиент хочет обмениваться пакетами. При отсутствии переменной принимается значение *, указывающее, что клиент запрашивает взаимодействие с любым разрешённым хостом. В переменной target можно указывать имена DNS, а также префиксы IPv6 и IPv4. Отметим, что идентификаторы зон адресации IPv6 [IPv6-ZONE-ID] не поддерживаются. Если в target указан префикс IP (адрес IP, за которым может следовать представленный %-кодом символ дробной черты / и размер префикса), запрос будет поддерживать только одну версию IP. Если target указывает имя хоста, предполагается, что IP-прокси выполнит распознавание DNS для определения маршрутов, анонсируемых клиенту. IP-прокси **следует** передавать капсулу ROUTE_ADVERTISEMENT, включающую маршруты для всех адресом, распознанных для указанного имени хоста, которые доступны IP-прокси и относятся к семейству адресов, для которого IP передаёт также назначенный адрес.

ipproto

Переменная ipproto содержит номер протокола (Internet Protocol Number) из реестра IANA Assigned Internet Protocol Numbers [IANA-PN]. Наличие протокола говорит, что хост хочет использовать для этого запроса только один протокол IP. Если указано значение * или переменная отсутствует, это говорит о запросе клиентом любых протоколов IP. Протокол IP, указанный в переменной ipproto, представляет дозволенное значение следующего заголовка в заголовке IP, передаваемом напрямую в дейтаграммах HTTP (внешние заголовки IP). Трафик ICMP разрешён всегда независимо от значения данного поля.

В терминах IPv6address, IPv4address, reg-name из [URI] переменные target и ipproto **должны** соответствовать формату, приведённому на рисунке 6, с использованием нотации [ABNF]. Дополнительные требования приведены ниже.

- Если target содержит адрес или префикс IPv6, для двоеточий (:) **должно** применяться %-кодирование. Например адрес 2001:db8::42 будет представлен в URI как 2001%3Adb8%3A%3A42.
- При указании размера префикса IP в target **нужно** включать символ /) с %-кодированием (%2F). Размер префикса IP **должен** указываться десятичным целым числом от 0 до размера IP-адреса в битах.
- Если в target указан префикс IP и его размер строго меньше размера IP-адреса в битах, младшие биты адреса, не входящие в префикс, **должны** иметь значение 0.
- Значением ipproto **должно** быть целое число от 0 до 255 или символ *.

```
target = IPv6prefix / IPv4prefix / reg-name / "*"
IPv6prefix = IPv6address ["%2F" 1*3DIGIT]
IPv4prefix = IPv4address ["%2F" 1*2DIGIT]
ipproto = 1*3DIGIT / "*"
```

Рисунок 6. Формат переменных URI Template.

IP-прокси **могут** применять контроль доступа с использованием предоставленных клиентом сведений о сфере действий, т. е. при отсутствии у клиента права доступа ни к одному из указанных им в области действия адресатов, IP-прокси может незамедлительно отклонить запрос.

4.7. Капсулы

Этот документ задаёт новые типы капсул, позволяющие конечным точкам обмениваться конфигурационными сведениями IP. Обе конечных точки могут передавать любое число таких капсул.

4.7.1. ADDRESS_ASSIGN

Капсула ADDRESS_ASSIGN (тип 0x01) позволяет конечной точке назначить своему партнёру набор адресов или префиксов IP. Каждая капсула содержит полный список префиксов IP, выделенных в настоящий момент её получателю. Любой из этих адресов может быть указан в поле источника пакетов IP от получателя данной капсулы.

```
ADDRESS_ASSIGN Capsule {
  Type (i) = 0x01,
  Length (i),
  Assigned Address (..) ...,
}
```

Рисунок 7. Формат капсулы ADDRESS_ASSIGN.

Капсула ADDRESS_ASSIGN может (но не обязана) включать 1 или несколько Assigned Address.

```
Assigned Address {
  Request ID (i),
  IP Version (8),
  IP Address (32..128),
  IP Prefix Length (8),
}
```

Рисунок 8. Формат назначенного адреса.

Поля назначенного адреса (Assigned Address) перечислены ниже.

Request ID

Идентификатор запроса в форме целого числа с переменным размером. Если это назначение адресов размещено в отклике на Address Request (параграф 4.7.2), в поле **нужно** указывать значение соответствующего поля в запросе. В ином случае в поле **нужно** помещать значение 0.

IP Version

Версия IP для данного назначения адреса, указанная 8-битовым целым числом (**должна** иметь значение 4 или 6).

IP Address

Назначенный адрес IP. Если в поле IP Version указано значение 4, поле IP Address **нужно** делать 32-битовым, а для IP Version = 6 - 128-битовым.

IP Prefix Length

Число битов адреса IP, служащих для задания назначенного префикса , указанное 8-битовым целым числом без знака. Значение **должно** быть не больше размера поля IP Address в битах. Если размер префикса совпадает с размером адреса, получателю капсулы разрешается передавать пакеты с одним адресом источника. Если размер префикса меньше размера адреса IP, получатель капсулы может передавать пакета с любым адресом из этого префикса в поле источника. Если размер префикса строго меньше размера адреса IP, в младших битах поля IP Address, не охватываемых префиксом, **должно** быть установлено значение 0.

Если какое-либо поле принятой капсулы имеет некорректный формат, получатель капсулы **должен** следовать процедуре обработки ошибок, заданной в параграфе 3.3 [HTTP-DGRAM].

Если капсула ADDRESS_ASSIGN не содержит адреса, указанного ранее в другой капсуле ADDRESS_ASSIGN, это говорит об удалении данного адреса. Капсула может быть пустой, указывая удаление всех адресов.

В некоторых внедрениях проксирования IP через HTTP конечной точке требуется получить адрес от партнёра до того, как ана узнает адрес источника для своих пакетов. Например, в варианте удалённого доступа в VPN (параграф 8.1) клиент не может передавать пакеты IP, пока не узнает, какой адрес использовать. В таких ситуациях конечная точка, ожидающая назначения адреса, **должна** передать капсулу ADDRESS_REQUEST. Это не требуется, если конечная точка не нуждается в назначении адреса, например, настроена автономно (out-of-band) со статическим адресом.

Хотя капсулы ADDRESS_ASSIGN обычно передаются в ответ на ADDRESS_REQUEST, конечные точки **могут** передавать ADDRESS_ASSIGN без запроса.

4.7.2. ADDRESS_REQUEST

Капсула ADDRESS_REQUEST (тип 0x02) позволяет конечной точке запросить у партнёра назначение адресов IP. Капсула позволяет конечной точке опционально указать свои предпочтения в части назначения адресов.

```
ADDRESS_REQUEST Capsule {
  Type (i) = 0x02,
  Length (i),
  Requested Address (..) ...,
}
```

Рисунок 9. Формат капсулы ADDRESS_REQUEST.

Капсула ADDRESS_REQUEST включает 1 или несколько Requested Address.

```
Requested Address {
  Request ID (i),
  IP Version (8),
  IP Address (32..128),
  IP Prefix Length (8),
}
```

Рисунок 10. Формат запрошенного адреса.

Поля Requested Address указаны ниже.

Request ID

Идентификатор данного запроса в форме целого числа с переменным размером. Каждый запрос от данной конечной точки имеет свой идентификатор. Конечной точке **недопустимо** использовать Request ID неоднократно и **недопустимо** указывать значение 0.

IP Version

Версия IP для данного запроса адреса, указанная 8-битовым целым числом (**должна** иметь значение 4 или 6).

IP Address

Запрашиваемый адрес IP. При IP Version = 4 поле IP Address **нужно** делать 32-битовым, при IP Version = 6 - 128-битовым.

IP Prefix Length

Размер запрашиваемого префикса IP в битах, указанный 8-битовым целым числом без знака. Значение **должно** быть не более размера поля IP Address в битах. Если размер префикса строго меньше размера адреса IP, в младших битах поля IP Address, не охватываемых префиксом, **должно** быть установлено значение 0.

Если адрес IP включает только 0 (0.0.0.0 или ::), это означает, что отправитель лишь запрашивает адрес из данного семейства, не отдавая предпочтения конкретным значениям. В этом случае размер префикса по-прежнему указывает предпочтения отправителя в части размера запрашиваемого префикса.

Если какое-либо поле принятой капсулы имеет некорректный формат, получатель капсулы **должен** следовать процедуре обработки ошибок, заданной в параграфе 3.3 [HTTP-DGRAM].

При получении капсулы ADDRESS_REQUEST конечной точке **следует** выделить своему партнёру 1 или множество адресов IP и ответить капсулой ADDRESS_ASSIGN для информирования того о назначении. Для каждого Requested Address получателю капсулы ADDRESS_REQUEST **нужно** ответить элементом Assigned Address с соответствующим Request ID. Если запрошенный адрес был выделен, в полях IP Address и IP Prefix Length элемента Assigned Address в отклике **нужно** указать выделенные значения. Если запрошенный адрес не был назначен, поле IP Address нужно заполнить нулями, а для IP Prefix Length SHALL нужно указать максимальный размер (0.0.0.0/32 или ::/128) для указания того, что адрес не выделен. Отклонённые адреса **не следует** включать в последующие капсулы ADDRESS_ASSIGN. Отметим, что в том же отклике ADDRESS_ASSIGN могут содержаться другие записи Assigned Address, не соответствующие какому-либо Request ID.

Если конечная точка получает капсулу ADDRESS_REQUEST без Requested Address, она **должна** прервать поток запроса проксирования IP.

Отметим, что порядок Requested Address не имеет какой-либо семантики, а Request ID служит лишь уникальным идентификатором, не задавая приоритета или важности.

4.7.3. ROUTE_ADVERTISEMENT

Капсула ROUTE_ADVERTISEMENT (тип 0x03) позволяет конечной точке сообщить своему партнёру о готовности маршрутизировать трафик для набора диапазонов адресов IP. Это говорит, что у отправителя капсулы уже есть маршруты к каждому диапазону адресов и уведомляет партнёра, что при отправке получателем капсулы ROUTE_ADVERTISEMENT пакетов IP в один из этих диапазонов в дейтаграммах HTTP, отправитель капсулы перешлёт их по имеющемуся маршруту. Любой из адресов, входящий в один из диапазонов, может быть адресом получателя в пакетах IP, отправляемых получателем капсулы.

```
ROUTE_ADVERTISEMENT Capsule {
  Type (i) = 0x03,
  Length (i),
  IP Address Range (..) ...,
}
```

Рисунок 11. Формат капсулы ROUTE_ADVERTISEMENT.

Капсула ROUTE_ADVERTISEMENT может (но не обязана) содержать 1 или несколько IP Address Range.

```
IP Address Range {
  IP Version (8),
  Start IP Address (32..128),
  End IP Address (32..128),
  IP Protocol (8),
}
```

Рисунок 12. Формат диапазона адресов IP.

Поля IP Address Range описаны ниже.

IP Version

Версия IP для данного диапазона, указанная 8-битовым целым числом (**должна** иметь значение 4 или 6).

Start IP Address и End IP Address

Первый и последний адреса IP в анонсируемом диапазоне. При IP Version = 4 эти поля **нужно** делать 32-битовыми, при IP Version = 6 - 128-битовыми. Значение Start IP Address **должно** быть не меньше End IP Address.

IP Protocol

Номер протокола IP для трафика, который может передаваться в этот диапазон, указанный 8-битовым целым числом без знака. Значение 0 разрешает все протоколы, все прочие представляют дозволенные значения следующего заголовка в заголовке IP, передаваемом напрямую в дейтаграммах HTTP (внешние заголовки IP). Трафик ICMP разрешён всегда независимо от значения данного поля.

Если какое-либо поле принятой капсулы имеет некорректный формат, получатель капсулы **должен** следовать процедуре обработки ошибок, заданной в параграфе 3.3 [HTTP-DGRAM].

После получения капсулы ROUTE_ADVERTISEMENT конечная точка **может** обновить своё локальное состояние в части того, что её партнёр готов маршрутизировать (в соответствии с локальной политикой), как при установке записей в таблице маршрутизации.

Каждая капсула ROUTE_ADVERTISEMENT содержит полный список диапазонов адресов. Если в одном направлении передано множество капсул ROUTE_ADVERTISEMENT, каждая из них заменяет предыдущие. Иными словами, если диапазон адресов был указан в предшествующей капсуле, но его нет в принятой капсуле ROUTE_ADVERTISEMENT, получатель будет считать маршрут отозванным.

Если несколько диапазонов, использующих один протокол IP, перекрываются, некоторые реализации маршрутных таблиц могут их отвергать. Для предотвращения перекрытия диапазоны упорядочиваются, это возлагается на отправителя и существенно упрощает проверку у получателя. Если IP Address Range A предшествует IP Address Range B в одной капсуле ROUTE_ADVERTISEMENT, **должны** соблюдаться приведённые ниже требования.

- Значение IP Version в A **должно** быть не больше IP Version в B.
- Если поля IP Version в A и B совпадают, значение IP Protocol в A **должно** быть не больше IP Protocol в B.
- Если поля IP Version и IP Protocol в A и B совпадают, значение End IP Address в A **должно** быть строго меньше Start IP Address в B.

При получении капсулы ROUTE_ADVERTISEMENT, не соответствующей приведённым требованиям, конечная точка **должна** прервать поток запроса проксирования IP.

Поскольку установка IP Protocol = 0 разрешает все протоколы, в соответствии с приведёнными выше требованиями возможно перекрытие двух маршрутов, в одном из которых указано IP Protocol = 0, а в другом - иное значение. Конечным точкам **недопустимо** передавать капсулы ROUTE_ADVERTISEMENT с маршрутами, перекрывающимися таким способом. Проверка этого **необязательна**, но при обнаружении несоответствия конечная точка **должна** прервать поток запроса проксирования IP.

4.8. Заголовки расширения IPv6

В области действия запроса (параграф 4.6) и капсуле ROUTE_ADVERTISEMENT (параграф 4.7.3) применяются номера протоколов IP. Эти номера представляют вышележащий уровень (см. раздел 2 в [IPv6] с примерами для TCP и UDP) или заголовки расширения IPv6 (см. раздел 4 в [IPv6] с примерами заголовков Fragment и Options). IP-прокси **могут** отклонять запросы на область действия для номеров протоколов, которые используются для заголовков расширения. При получении пакетов реализации, поддерживающие установку области действия или маршрутизацию по номеру протокола IP, **должны** пройти по цепочке расширений для обнаружения самого внешнего номера протокола, не являющегося расширением, для сопоставления с областью действия. Отметим, что в капсулах ROUTE_ADVERTISEMENT используется номер протокола 0 для разрешения всех протоколов. Это не ограничивает маршрут заголовком IPv6 Hop-by-Hop Options (параграф 4.3 в [IPv6]).

5. Идентификаторы контекста

Заданный в этом документе механизм проксирования IP в HTTP позволяет будущим расширениям обмениваться дейтаграммами HTTP с семантикой, отлично от содержимого IP (payload). Некоторые из таких расширений могут дополнять содержимое IP данными или сжимать заголовки IP, а другие - обмениваться данными, которые полностью отделены от содержимого IP. Для этого все дейтаграммы HTTP, связанные с запросами проксирования IP, начинаются с поля Context ID, как описано в разделе 6.

Context ID указывается 62-битовым целым числом (0 262-1). Значения Context ID кодируются с переменным размером, как указано в разделе 16 [QUIC]. Значение 0 зарезервировано для содержимого IP (payload), все прочие выделяются динамически, чётные значения выделяются клиентам, нечётные - прокси. Пространство Context ID привязано к данному запросу HTTP и Context ID с одинаковым значением могут выделяться в разных запросах и иметь разную семантику. Значения Context ID **недопустимо** повторно выделять в данном запросе HTTP, но для них **можно** использовать любой порядок. Ограничения на использование чётных и нечётных Context ID введены для избавления от необходимости синхронизации между конечными точками. Однако после назначения Context ID эти ограничения не применяются к использованию идентификатора и его может использовать как клиент, так и IP-прокси, независимо от того, кто назначил идентификатор.

Регистрация - это действие, с помощью которого конечная точка информирует партнёра о семантике и формате данного Context ID. Этот документ не задаёт процедуру регистрации. Будущие расширения **могут** использовать для регистрации Context ID поля заголовков HTTP или капсулы. В зависимости от применяемого метода возможно получение дейтаграмм с ещё незарегистрированным Context ID, например в результате изменения порядка доставки пакетов с дейтаграммами.

6. Формат содержимого дейтаграмм HTTP

Связанные с запросом проксирования IP поля HTTP Datagram Payload в дейтаграммах HTTP (см. [HTTP-DGRAM]) имеют формат, показанный на рисунке 13. Отметим, что при кодировании дейтаграмм HTTP в кадры QUIC DATAGRAM поле Context ID, определённое ниже, следует сразу за полем Quarter Stream ID, которое размещается вв начале содержимого (payload) кадра QUIC DATAGRAM.

```
IP Proxying HTTP Datagram Payload {
  Context ID (i),
  Payload (..),
}
```

Рисунок 13. Формат дейтаграммы HTTP для проксирования IP.

Поля IP Proxying HTTP Datagram Payload описаны ниже.

Context ID

Целое число переменного размера, содержащее значение Context ID. При получении дейтаграммы HTTP/3 с неизвестным ID получателю **нужно** отбросить дейтаграмму без уведомления или временно буферизовать её (примерно на интервал кругового обхода) в ожидании регистрации соответствующего Context ID.

Payload

Содержимое дейтаграммы, семантика которого зависит от значения предыдущего поля. Поле может быть пустым.

Пакеты IP, кодируются с использованием дейтаграмм HTTP с Context ID - 0. В этом случае поле Payload содержит полный пакет IP (от поля IP Version до последнего байта IP payload).

7. Обработка пакетов IP

Этот документ определяет механизм туннелирования, концептуально являющийся каналом IP. Однако каналы подключены к маршрутизаторам IP, поэтому реализациям придётся принять на себя ответственность за некоторые функции маршрутизации IP, если они не передают это другой реализации, например, ядру.

7.1. Канальные операции

Описанные в этом документе туннели для пересылки IP не являются полнофункциональными «интерфейсами» в смысле архитектуры адресации IPv6 [IPv6-ADDR]. В частности, у них может не быть адресов IPv6 link-local. Кроме того, на этих интерфейсах не применяется автоматическая настройка IPv6 без учёта состояния и обнаружение соседей.

При использовании HTTP/2 или HTTP/3 клиент **может** оптимистично начать отправку проксируемых пакетов IP до получения отклика на ской запрос проксирования, понимая, что IP-прокси может не обработать их, если он ответит отказом или дейтаграммы будут получены IP-прокси до приёма запроса. Поскольку приёмные адреса и маршруты нужны, чтобы знать о возможности передачи пакета через туннель, такие оптимистические пакеты могут отбрасываться IP-прокси, если тот решит предоставить не те адреса или данные маршрутизации, нежели предположил клиент.

Отметим, что в один пакет может быть инкапсулировано несколько проксируемых пакетов IP, поскольку пакет QUIC может включать не один кадр QUIC DATAGRAM. Возможно также разделение проксируемого пакета IP между несколькими инкапсулирующими пакетами, поскольку капсулу DATAGRAM можно распределить между несколькими пакетами QUIC или TCP.

7.2. Маршрутизация

Требования этого раздела повторяют требования к маршрутизаторам IP в целом и могут не относиться к реализациям проксирования IP, полагающимися на маршрутизацию внешними программами.

Когда конечная точка получает дейтаграмму HTTP с пакетом IP, она анализирует заголовок пакета IP, выполняет проверки локальных правил (например, адреса отправителя), просматривает таблицу маршрутизации для определения выходного интерфейса, а затем передаёт пакет IP в этот интерфейс или локальному приложению. Конечная точка может также отбросить любой принятый пакет вместо его пересылки. Если принятый пакет IP не проходит какую-либо проверку (корректность или правила), с точки зрения проксирования IP это будет ошибкой пересылки, а не нарушением протокола (см. параграф 7.2). Конечные точки проксирования IP **могут** реализовать дополнительные правила фильтрации пересылаемых пакетов IP.

В другом направлении при получении конечной точкой пакета IP она проверяет его соответствие маршрутам, сопоставленным с туннелем, и выполняет указанные выше проверки перед отправкой пакета в дейтаграмме HTTP.

При пересылке конечными точками IP-проксирования пакетов IP между разными каналами они декрементируют IP Hop Count (или TTL) при инкапсуляции, но не делают этого при декапсуляции. Иными словами, Hop Count уменьшается непосредственно перед отправкой пакета IP в дейтаграмме HTTP. Это предотвращает петли при наличии маршрутных петель и соответствует вариантам IPsec [IPSEC]. Сказанное не применяется к пакетам IP, созданных самой конечной точкой проксирования IP.

Разработчики должны убедиться, что трафик link-local не пересылается за пределы интерфейса проксирования IP, на котором он был получен. Конечные точки проксирования IP должны также отвечать на пакеты, направленные по групповому адресу link-local.

IPv6 требует на каждом канале значение MTU не менее 1280 байтов [IPv6]. Поскольку при проксировании IP в HTTP пакеты IP передаются в дейтаграммах HTTP, которые, в свою очередь, могут передаваться в кадрах QUIC DATAGRAM, которые не могут фрагментироваться [DGRAM], значение MTU в туннеле IP может быть ограничено MTU соединения QUIC, через которое работает проксирование IP. Это может приводить к нарушению требования IPv6 к минимальному значению MTU. Конечные точки проксирования IP, работающие как маршрутизаторы и поддерживающие IPv6, **должны** убедиться, что MTU в канале IP-туннеля не менее 1280 байтов (т. е. можно отправлять дейтаграммы HTTP с размером данных не менее 1280 байтов). Это можно обеспечить разными методами.

- Если обе конечных точки проксирования IP знают об отсутствии на пути посредников HTTP, они могут заполнять (pad) пакеты QUIC INITIAL внешнего соединения QUIC, через которое работает проксирование IP[1].
- Конечные точки проксирования IP могут также передавать пакеты запросов ICMPv6 echo со 1232 байтами данных для определения MTU канала и разрывать туннель при отсутствии ответа. Если у конечных точек нет автономных (out-of-band) средств, гарантирующих достаточность предыдущих методов, они **должны** применять этот метод. Если конечная точка не знает адреса IPv6 своего партнера, она может передать запрос ICMPv6 echo по групповому адресу link-local для всех узлов (ff02::1).

Если конечная точка использует кадры QUIC DATAGRAM для передачи пакетов IPv6 и обнаруживает, что значение QUIC MTU слишком мало для передачи 1280 байтов, она **должна** прервать поток запроса проксирования IP.

7.2.1. Сигналы об ошибках

Поскольку конечные точки проксирования IP часто пересылают пакеты IP на другие сетевые интерфейсы, им нужно обрабатывать ошибки в процессе такой пересылки. Например, при пересылке может возникнуть отказ, связанный с отсутствием у конечной точки маршрута к адресату, если она настроена правилами на отклонение префикса адресата, или MTU исходящего канала меньше размера пересылаемого пакета. В таких ситуациях конечным точкам проксирования IP **следует** использовать ICMP [ICMP] [ICMPv6] для сигнализации ошибок пересылки своему партнёру в пакетах ICMP, передаваемых серез дейтаграммы HTTP. Конечная точка может самостоятельно выбирать подходящие сообщения ICMP для передачи ошибок. Некоторые примеры, актуальные для проксирования IP указаны ниже.

- Для недействительных адресов источника применяется Destination Unreachable (параграф 3.1 в [ICMPv6]) с кодом 5 Source address failed ingress/egress policy (несоответствие правилам для адреса источника).
- Для немаршрутизируемых адресов получателей применяется Destination Unreachable (параграф 3.1 в [ICMPv6]) с кодом 0 No route to destination (нет маршрута к получателю) или 1 Communication with destination administratively prohibited (связь с адресатом запрещена административно).
- Для пакетов, не помещающихся в размер MTU на выходном канале применяется Packet Too Big (параграф 3.2 в [ICMPv6]).

Для получения таких сведений об ошибках конечные точки должны быть готовы принимать пакеты ICMP. Если конечная точка не передаёт капсулы ROUTE_ADVERTISEMENT, например, из-за того, что клиент открывает поток IP через IP-прокси, ей **следует** обрабатывать проксируемые пакеты ICMP от своего партнёра для получения сведений об ошибках. Отметим, что сообщения ICMP могут приходить с адреса, отличающегося от адреса партнёра и даже извне цели, если применяется установка области действия (см. параграф 4.6).

8. Примеры

IP-проксирование в HTTP позволяет реализовать множество сценариев с использованием преимуществ проксирования и туннелирования пакетов IP. Представленные ниже примеры служат иллюстрациями этого.

8.1. VPN для удалённого доступа

Ниже приведён пример организации туннеля VPN в сеть, где клиент получает набор локальных адресов и может передавать любому удалённому хосту через IP-прокси. Туннель может быть полным или расщепленным.

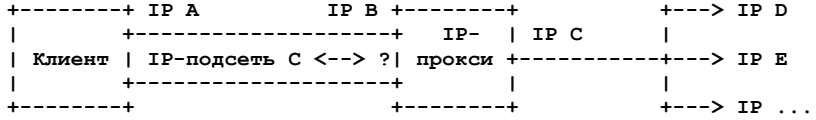

Рисунок 14. Организация туннеля VPN.

Клиент в этом случае не задаёт для запроса область действия. IP-прокси назначает клиенту адрес IPv4 (192.0.2.11) и туннельный маршрут ко всем адресам IPv4 (0.0.0.0/0). Клиент может взаимодействовать с любым хостом IPv4, используя назначенный адрес в качестве адреса источника.

[1]При использовании QUIC версии 1 издержки составляют 1 байт для типа, 20 для максимального размера идентификатора соединения, 4 байта для максимального размера пакетов, 1 байт для типа кадра DATAGRAM, 8 байтов для максимального Quarter Stream ID, 1 байт для Context ID = 0 и 16 байтов ждя тега аутентификации шифрования с аутентификацией и связанными данными (Authenticated Encryption with Associated Data или AEAD), что составляет 51 и соответствует заполнению пакетов QUIC INITIAL до 1331 байта и более.

```
[[ От клиента ]]                [[ От IP-прокси ]]

SETTINGS
  H3_DATAGRAM = 1

                                SETTINGS
                                  ENABLE_CONNECT_PROTOCOL = 1
                                  H3_DATAGRAM = 1

STREAM(44): HEADERS
:method = CONNECT
:protocol = connect-ip
:scheme = https
:path = /vpn
:authority = proxy.example.com
capsule-protocol = ?1

                                STREAM(44): HEADERS
                                :status = 200
                                capsule-protocol = ?1

STREAM(44): DATA
Capsule Type = ADDRESS_REQUEST
(Request ID = 1
 IP Version = 4
 IP Address = 0.0.0.0
 IP Prefix Length = 32)

                                STREAM(44): DATA
                                Capsule Type = ADDRESS_ASSIGN
                                (Request ID = 1
                                 IP Version = 4
                                 IP Address = 192.0.2.11
                                 IP Prefix Length = 32)

                                STREAM(44): DATA
                                Capsule Type = ROUTE_ADVERTISEMENT
                                (IP Version = 4
                                 Start IP Address = 0.0.0.0
                                 End IP Address = 255.255.255.255
                                 IP Protocol = 0) // Any

DATAGRAM
Quarter Stream ID = 11
Context ID = 0
Payload = Encapsulated IP Packet

                                DATAGRAM
                                Quarter Stream ID = 11
                                Context ID = 0
                                Payload = Encapsulated IP Packet
```

Рисунок 15. Пример полного туннеля VPN.

Настройка расщеплённого туннеля VPN (клиент имеет доступ лишь к конкретному набору приватных адресов) достаточно похожа. В этом случае анонсируется маршрут 192.0.2.0/24 вместо 0.0.0.0/0.

```
[[ От клиента ]]                [[ От IP-прокси ]]

                                STREAM(44): DATA
                                Capsule Type = ADDRESS_ASSIGN
                                (Request ID = 0
                                 IP Version = 4
                                 IP Address = 192.0.2.42
                                 IP Prefix Length = 32)

                                STREAM(44): DATA
                                Capsule Type = ROUTE_ADVERTISEMENT
                                (IP Version = 4
                                 Start IP Address = 192.0.2.0
                                 End IP Address = 192.0.2.41
                                 IP Protocol = 0) // Any
                                (IP Version = 4
                                 Start IP Address = 192.0.2.43
                                 End IP Address = 192.0.2.255
                                 IP Protocol = 0) // Any
```

Рисунок 16. Пример расщеплённого туннеля VPN.

8.2. VPN между сайтами

Ниже показано, как подключить сеть филиала к корпоративной сети, чтобы све машины этих сетей могли взаимодействовать. В примере клиент проксирования IP подключён к сети филиала 192.0.2.0/24, а IP-прокси - к корпоративной сети 203.0.113.0/24. В сети филиала имеются унаследованные клиенты, которые поддерживают запросы лишь от машин своей подсети, поэтому IP-прокси назначается адрес IP из этой подсети.

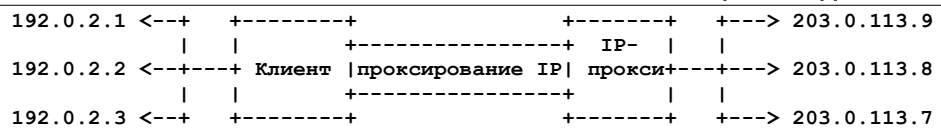

Рисунок 17. Пример VPN между сайтами.

Клиент в этом случае не задаёт для запроса область действия. IP-прокси назначает клиенту адрес IPv4 (203.0.113.100) и маршрут с расщепленным туннелем в корпоративную сеть (203.0.113.0/24). Клиент назначает IP-прокси адрес IPv4 (192.0.2.200) и маршрут с расщепленным туннелем в сеть филиала (192.0.2.0/24). Это позволяет хостам обеих сетейц взаимодействовать между собой, а IP-прокси - поддерживать устаревшие хосты в сети филиала. Отметим, что конечные точки проксирования IP будут декрементировать IP Hop Count (или TTL) при декапсуляции пересылаемых пакетов, поэтому протоколы, требующие с поле значения 255, не будут работать.

```
[[ От клиента ]]                [[ От IP-прокси ]]

SETTINGS
  H3_DATAGRAM = 1

                                SETTINGS
                                  ENABLE_CONNECT_PROTOCOL = 1
                                  H3_DATAGRAM = 1

STREAM(44): HEADERS
:method = CONNECT
:protocol = connect-ip
:scheme = https
:path = /corp
:authority = proxy.example.com
capsule-protocol = ?1

                                STREAM(44): HEADERS
                                :status = 200
                                capsule-protocol = ?1

STREAM(44): DATA
Capsule Type = ADDRESS_ASSIGN
(Request ID = 0
IP Version = 4
IP Address = 192.0.2.200
IP Prefix Length = 32)

STREAM(44): DATA
Capsule Type = ROUTE_ADVERTISEMENT
(IP Version = 4
Start IP Address = 192.0.2.0
End IP Address = 192.0.2.255
IP Protocol = 0) // Any

                                STREAM(44): DATA
                                Capsule Type = ADDRESS_ASSIGN
                                (Request ID = 0
                                 IP Version = 4
                                 IP Address = 203.0.113.100
                                 IP Prefix Length = 32)

                                STREAM(44): DATA
                                Capsule Type = ROUTE_ADVERTISEMENT
                                (IP Version = 4
                                 Start IP Address = 203.0.113.0
                                 End IP Address = 203.0.113.255
                                 IP Protocol = 0) // Any

DATAGRAM
Quarter Stream ID = 11
Context ID = 0
Payload = Encapsulated IP Packet

                                DATAGRAM
                                Quarter Stream ID = 11
                                Context ID = 0
                                Payload = Encapsulated IP Packet
```

Рисунок 18. Пример капсулы для VPN между сайтами.

8.3. Пересылка потока IP

В следующем примере показана организация пересылки потоков IP, где клиент запрашивает создание туннеля для пересылки в target.example.com с использванием протокола управлегния потоковой передачей (Stream Control Transmission Protocol или SCTP, протокол IP 132) и получает 1 локальный адрес и удалённый адрес, который можно применять для передачи пакетов. Аналогичный подход можно применять для любого протокола IP, который не так просто проксировать с помощью имеющихся методов HTTP , например, ICMP, ESP[1] и т. п.

[1]Encapsulating Security Payload - инкапсуляция защищённого содержимого.

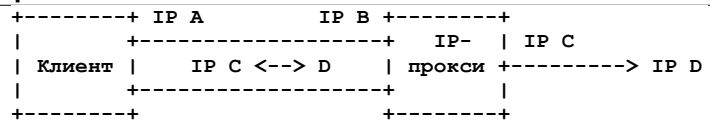

Рисунок 19. Организация потока с проксированием.

В этом случае клиент задаёт имя целевого хоста и номер протокола IP для области действия своего запроса, указывая, что взаимодействовать нужно с единственным хостом. IP-прокси может выполнять распознавание DNS от имени клиента и выделяет клиенту конкретный выходной сокет вместо назначения ему IP-адреса целиком. В этом смысле запрос похож на обычный запрос CONNECT.

IP-прокси назначает клиенту 1 адрес IPv6 (2001:db8:1234::a) и маршрут к 1 хосту IPv6 (2001:db8:3456::b) с привязкой к протоколу SCTP. Клиент может обмениваться с удаленным хостам пакетами SCTP.

```
[[ От клиента ]]                [[ От IP-прокси ]]

SETTINGS
  H3_DATAGRAM = 1

                                SETTINGS
                                  ENABLE_CONNECT_PROTOCOL = 1
                                  H3_DATAGRAM = 1

STREAM(44): HEADERS
:method = CONNECT
:protocol = connect-ip
:scheme = https
:path = /proxy?target=target.example.com&ipproto=132
:authority = proxy.example.com
capsule-protocol = ?1

                                STREAM(44): HEADERS
                                :status = 200
                                capsule-protocol = ?1

                                STREAM(44): DATA
                                Capsule Type = ADDRESS_ASSIGN
                                (Request ID = 0
                                 IP Version = 6
                                 IP Address = 2001:db8:1234::a
                                 IP Prefix Length = 128)

                                STREAM(44): DATA
                                Capsule Type = ROUTE_ADVERTISEMENT
                                (IP Version = 6
                                 Start IP Address = 2001:db8:3456::b
                                 End IP Address = 2001:db8:3456::b
                                 IP Protocol = 132)

DATAGRAM
Quarter Stream ID = 11
Context ID = 0
Payload = Encapsulated SCTP/IP Packet

                                DATAGRAM
                                Quarter Stream ID = 11
                                Context ID = 0
                                Payload = Encapsulated SCTP/IP Packet
```

Рисунок 20. Пример проксирования потока SCTP.

8.4. Одновременное проксирование соединений

В следующем примере показана схема, где клиент проксирует пакеты UDP через IP-прокси для организации одновременных управляющих соединений через IP-прокси, как описано в Happy Eyeballs [HEv2]. Это является вариантом проксируемого потока, но показывает, как проксирование на уровне IP может открывать новые возможности даже для TCP и UDP.

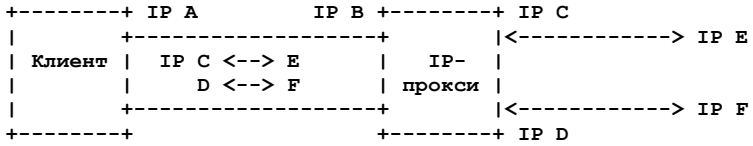

Рисунок 21. Одновременное проксирование соединений.

Как и в случае проксирования потоков, клиент задаёт имя целевого хоста и протокол IP в области действия своего запроса. Когда IP-прокси выполняет распознавание имён DNS от имени клиента, он может передавать клиенту различные варианты удалённого адреса как отдельные маршруты. Можно также убедиться, что клиенты назначены как адреса IPv4, так и IPv6.

IP-прокси выделяет клиенту адреса IPv4 (192.0.2.3) и IPv6 (2001:db8:1234::a), а также маршруты IPv4 (198.51.100.2) и IPv6 (2001:db8:3456::b), представляющие распознанные адреса целевого хоста, с привязкой к UDP. Клиент может обмениваться пакетами UDP IP с одним из адресов IP-прокси для поддержки Happy Eyeballs через IP-прокси.

```
[[ От клиента ]]                [[ От IP-прокси ]]

SETTINGS
  H3_DATAGRAM = 1

                                SETTINGS
                                  ENABLE_CONNECT_PROTOCOL = 1
                                  H3_DATAGRAM = 1

STREAM(44): HEADERS
:method = CONNECT
:protocol = connect-ip
:scheme = https
:path = /proxy?target=target.example.com&ipproto=17
:authority = proxy.example.com
capsule-protocol = ?1

                                STREAM(44): HEADERS
                                :status = 200
                                capsule-protocol = ?1

                                STREAM(44): DATA
                                Capsule Type = ADDRESS_ASSIGN
                                (Request ID = 0
                                 IP Version = 4
                                 IP Address = 192.0.2.3
                                 IP Prefix Length = 32),
                                (Request ID = 0
                                 IP Version = 6
                                 IP Address = 2001:db8::1234:1234
                                 IP Prefix Length = 128)

                                STREAM(44): DATA
                                Capsule Type = ROUTE_ADVERTISEMENT
                                (IP Version = 4
                                 Start IP Address = 198.51.100.2
                                 End IP Address = 198.51.100.2
                                 IP Protocol = 17),
                                (IP Version = 6
                                 Start IP Address = 2001:db8:3456::b
                                 End IP Address = 2001:db8:3456::b
                                 IP Protocol = 17)
...

DATAGRAM
Quarter Stream ID = 11
Context ID = 0
Payload = Encapsulated IPv6 Packet

DATAGRAM
Quarter Stream ID = 11
Context ID = 0
Payload = Encapsulated IPv4 Packet
```

Рисунок 22. Пример одновременных соединений.

9. Вопросы расширяемости

Расширения IP-проксирования в HTTP могут менять поведение описанного механизма. Таким расширениям **следует** определять новые типы капсул для обмена конфигурационными сведениями, если это требуется. Расширениям, меняющим адресацию, **рекомендуется** задавать передачу своих капсул расширения до ADDRESS_ASSIGN и вводить их в действие лишь после анализа капсулы ADDRESS_ASSIGN. Это позволяет обеспечить неделимость изменённого назначения адресов. Расширениям, меняющим маршрутизацию, следует вести себя аналогично по отношению к капсулам ROUTE_ADVERTISEMENT.

10. Вопросы производительности

Трафик с пиками часто вызвает временную корреляцию потери пакетов, что, в свою очередь, может приводить к неоптимальной реакции контроллеров перегрузки в протоколах, работающих внутри туннеля. Для предотвращения этого конечным точкам проксирования IP **следует** избегать роста пиков трафика IP и **не следует** помещать пакеты в очередь с целью снижения из связывания (batching) ниже минимально необходимого для использования преимуществ выгрузки в оборудование.

При использовании работающим в туннеле протоколом контроля перегрузки (например, [TCP] или [QUIC]), проксируемый трафик будет иметь по меньшей мере два вложенных контроллера перегрузок. При передаче туннелируемых пакетов с использованием кадров QUIC DATAGRAM внешнее соединение HTTP **может** отключать контроль перегрузки для пакетов, содержащих лишь кадры QUIC DATAGRAM, инкапсулирующие пакеты IP. Разработчикам будет полезно обратиться к рекомендациям параграфа 3.1.11 в [UDP-USAGE].

При использовании работающим в туннеле протоколом восстановления потерь (например, [TCP] или [QUIC]) и работе внешнего соединения HTTP по протоколу TCP проксируемый трафик будет иметь по меньшей мере два вложенных механизма восстановления потерь. Это может снижать производительность, поскольку иногда оба механизма могут независимо передавать повторно одни и те же данные. Для предотвращения этого проксирование IP **следует** организовывать через HTTP/3, где можно использовать кадры QUIC DATAGRAM.

10.1. MTU

При использовании HTTP/3 с расширением QUIC Datagram [DGRAM] пакеты IP передаются в кадрах QUIC DATAGRAM. Поскольку эти кадры не могут фрагментироваться, в них можно передавать лишь данные, размер которых определяется конфигурацией соединения QUIC и Path MTU (PMTU). Если конечная точка использует кадры QUIC DATAGRAM и пытается маршрутизировать через туннель пакеты IP, которые не помещаются в кадр QUIC DATAGRAM, IP-прокси **не следует** передавать такие пакеты в капсуле DATAGRAM, поскольку это нарушает сквозные параметры надёжности от которых зависят такие методы, как DPLPMTUD[1] [DPLPMTUD]. В этом случае конечной точке **следует** отбросить пакет IP и передать его отправителю сообщение ICMP Packet Too Big (см. параграф 3.2 в [ICMPv6]).

10.2. ECN

Если конечная точка проксирования IP с соединением, содержащим поток запроса проксирования IP, отключает контроль перегрузки, она не может передавать явных уведомлений о перегрузке (Explicit Congestion Notification или ECN) [ECN] в этом внешнем соединении. Т. е. отправитель QUIC **должен** помещать во все заголовки IP код Not-ECT[2] для пакетов QUIC, на которые не распространяется контроль перегрузок. Конечная точка все ещё может передавать обратную связь ECN в кадрах QUIC ACK_ECN или бите TCP ECN-Echo (ECE), поскольку партнёр мог не отключить контроль перегрузки.

Если контроль перегрузки не отключён на внешнем соединении, рекомендации [ECN-TUNNEL] для передачи маркировки ECN между внутренним и внешним заголовком IP не применяются, поскольку внешнее соединение будет корректно реагировать на перегрузки при использовании ECN. Для внутреннего трафика также можно применять ECN независимо от использования на внешнем соединении.

10.3. Дифференцированное обслуживание

Туннелируемые пакеты IP могут иметь коды дифференцированного обслуживания (Differentiated Services Code Point или DSCP) [DSCP] в поле класса трафика в заголовке IP для запроса определённого поведения на этапах пересылки (per-hop behavior). Если конечная точка проксирования IP настроена как часть домена с дифференцированным обслуживанием, она **может** дифференцировать трафик на основе этой маркировки. Однако использование HTTP может ограничивать возможности дифференцированной обработки туннелируемых пакетов IP на пути между конечными точками проксирования IP.

Когда в соединении HTTP контролируется перегрузка, маркировка пакетов разными кодами DSCP может приводить к нарушению порядка доставки, а это, в свою очередь, - к плохой работе транспортного контроллера перегрузки. Если туннелируемые пакеты подвергаются контролю перегрузки на внешнем соединении, для них нужно избегать маркировки DSCP, которая не эквивалентна поведению при пересылке. В этом случае конечной точке проксирования IP **недопустимо** копировать поле DSCP из внутреннего заголовка IP во внешний заголовок. Вместо этого приложение должно использовать отдельные соединения с прокси для каждого кода DSCP. Отметим, что этот документ не задаёт способ связывания области действия запросов с конкертным значением DSCP (оставлено для будущих расширений).

Если туннелируемые пакеты используют дейтаграммы QUIC и не подвергаются контролю перегрузок во внешнем соединении, конечные точки проксирования IP **могут** транслировать значения поля DSCP из туннелируемого трафика во внешний заголовок IP. Конечным точкам проксирования IP **недопустимо** объединять несколько внутренних пакетов в один внешний пакет, если они не имеют одинакового кода DSCP или эквивалентных классов обслуживания. Отметим, что возможность трансляции значений DSCP зависит от принадлежности входа и выхода туннеля к одному или разным доменам дифференцированного обслуживания.

11. Вопросы безопасности

Предоставление произвольным клиентам возможности создавать туннели, позволяющие передавать данные произвольным хостам независимо от привязки этих туннелей у конкретным хостам, сопряжено со значительными рисками. Злоумышленники могут воспользоваться этой возможностью для отправки трафика, приписывая его IP-прокси. Серверам HTTP, поддерживающим проксирование IP, **следует** предоставлять эту возможность лишь уполномоченным пользователям. В зависимости от развёртывания возможные механизмы аутентификации включают TLS между конечными точками проксирования IP, аутентификацию на основе HTTP с помощью заголовка HTTP Authorization [HTTP] и даже маркеры носителя (bearer). Прокси могут применять правила для аутентифицированных пользователей, чтобы дополнительно ограничивать поведение клиентов или бороться с возможными злоупотреблениями. Например, прокси могут ограничивать скорость для отдельных клиентов, передающих слишком большой объем трафика через прокси. Можно также ограничивать назначение клиентам адресов и префиксов на основе неких атрибутов клиента, таких как географическое местоположение.

Назначение адресов может влиять на приватность конечных точек. Например, если прокси разделит своё адресное пространство по числу аутентифицированных клиентов и затем назначит каждому клиенту свой диапазон адресов, целевые хосты могут воспользоваться этими сведениями для определения принадлежности пакетов IP конкретному клиенту. Предотвращение такого отслеживания может быть важно для некоторых внедрений. Прокси **следует** избегать назначения клиенту постоянных адресов или префиксов, если это возможно.

Фальсификация IP-адресов источника часто применяется для организации атак с отказом в обслуживании (denial-of-service или DoS). Реализация описанного здесь механизма должна убедиться, что она не будет способствовать таким атакам. В частности, возможны случаи, когда конечная точка знает, что её партнёру разрешено передавать пакеты IP лишь из определённого префикса. Например, это может быть обусловлено получением конфигурационных данных по отдельному каналу (out-of-band) или обобщением разрешённых префиксов через капсулы ADDRESS_ASSIGN. В таких случаях конечные точки **должны** следовать рекомендациям [BCP38] для предотвращения подмены адреса источника.

Ограничение области действия запроса (параграф 4.6) позволяет двум клиентам использовать один из внешних IP-адресов прокси, если их запросы привязаны к разным номерам протоколов IP. Если прокси получает пакет ICMP, направленный на такой внешний адрес, он может переслать его клиентам. Однако в некоторых пакетах ICMP содержится часть пакета IP, вызвавшего отклик ICMP. Пересылка таких пакетов может привести к случайному

[1] Datagram Packetization Layer PMTU Discovery - определение PMTU на уровне пакетизации дейтаграмм.
[2] Not ECN-Capable Transport - не поддерживающий ECN транспорт.

раскрытию сведений о клиенте другм клиентам. Для предоствращения этого прокси, пересылающие ICMP по общему для клиентов внешнему адресу, **должны** проверять вызывающие отклик пакеты внутри ICMP и пересылать пакет ICMP лишь клиенту, для которого область действия соответствует вызвавшему отклик пакету.

Разрабочикам будет полезно ознакомиться с рекомендациями [TUNNEL-SECURITY]. Поскольку известны риски, связанные с некоторыми заголовками расширения IPv6 (например, [ROUTING-HDR]), разработчики должны следовать последним рекомендациям по работе с такими заголовками.

Перенос маркировки DSCP из внутреннего заголовка ао внешний пакет (параграф 10.3) раскрыват сведения об уровне сквозного потока наблюдателям между конечными точками проксирования IP. Это может приводить к раскрытию отдельного сквозного потока. Поэтому такие использование DSCP в чувствительном к приватности контексте **не рекомендуется**.

Приспосабливающаяся (opportunistic) отправка пакетов IP (параграф 7.1) не разрешается в HTTP/1.x, поскольку сервер может отклонить HTTP Upgrade и пытаться проанализировать пакеты IP как последующие запросы HTTP, что позволит организовать атаку с контрабандой (smuggling) запросов (см. [OPTIMISTIC]). В частности, посреднику, перекодирующему запрос из HTTP/2 или HTTP/3 в HTTP/1.1, **недопустимо** пересылать полученные капсулы, пока он не проанализировал отклик об успешном проксировании IP.

12. Взаимодействие с IANA

12.1. Регистрация маркера HTTP Upgrade

Агентство IANA включило connect-ip в реестр HTTP Upgrade Tokens (https://www.iana.org/assignments/http-upgrade-tokens).

```
Value:  connect-ip
Description:  Proxying of IP Payloads
Expected Version Tokens:  None
References:  RFC 9484
```

12.2. Создание реестра MASQUE URI Suffixes

В IANA создан реестр MASQUE URI Suffixes (https://www.iana.org/assignments/masque) с процедурой регистрации Expert Review (параграф 4.5 в [IANA-POLICY]). В новый реестр включается сегмент пути, следующий сразу после masque в путях, начинающихся с /.well-known/masque/ (элемент masque зарегистрирован в реестре Well-Known URIs, https://www.iana.org/assignments/well-known-uris).

Новый реестр включает три колонки.

Path Segment - сегмент пути
: Строка ASCII с символами, разрешенными для маркеров (см. параграф 5.6.2 в [HTTP]. Записи реестра должны иметь уникальные значения в этой колонке.

Description - описание
: Описание записи.

Reference - документ
: Необязательная ссылка на описание использования записи.

Исходное содержимое реестра приведено в таблице 1.

Таблица 1. Реестр MASQUE URI Suffixes.

Сегмент пути	***Описание***	***Документ***
udp	UDP Proxying	RFC 9298
ip	IP Proxying	RFC 9484

Назначенным экспертам для этого реестра рекомендуется одобрять все запросы, если эксперт считает, что (1) запрошенное значение Path Segment не конфликтует с имеющимися и ожидаемыми в будущих работах IETF и (2) вариант использования связан с проксированием.

12.3. Обновление регистрации общеизвестного URI для masque

Агентство IANA обновило запись для суффикса URI masque в реестре Well-Known URIs (https://www.iana.org/assignments/well-known-uris). Обновлено поле Reference указанием на этот документ, а в поле Related Information указано For sub-suffix allocations (см. https://www.iana.org/assignments/masque).

12.4. Регистрация типов капсул HTTP

Агентство IANA добавило указанные в таблице 2 значения в реестр HTTP Capsule Types (https://www.iana.org/assignments/masque).

Таблица 2. Новые капсулы.

Значение	***Тип капсулы***
0x01	ADDRESS_ASSIGN
0x02	ADDRESS_REQUEST
0x03	ROUTE_ADVERTISEMENT

Значения других полей для всех новых записей показаны ниже.

```
Status:  permanent
Reference:  RFC 9484
Change Controller:  IETF
```

Contact: masque@ietf.org
Notes: None

13. Литература

13.1. Нормативные документы

[ABNF] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[BCP38] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[DGRAM] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", RFC 9221, DOI 10.17487/RFC9221, March 2022, <https://www.rfc-editor.org/info/rfc9221>.

[DSCP] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

[ECN] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[EXT-CONNECT2] McManus, P., "Bootstrapping WebSockets with HTTP/2", RFC 8441, DOI 10.17487/RFC8441, September 2018, <https://www.rfc-editor.org/info/rfc8441>.

[EXT-CONNECT3] Hamilton, R., "Bootstrapping WebSockets with HTTP/3", RFC 9220, DOI 10.17487/RFC9220, June 2022, <https://www.rfc-editor.org/info/rfc9220>.

[HTTP] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[HTTP-DGRAM] Schinazi, D. and L. Pardue, "HTTP Datagrams and the Capsule Protocol", RFC 9297, DOI 10.17487/RFC9297, August 2022, <https://www.rfc-editor.org/info/rfc9297>.

[HTTP/1.1] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <https://www.rfc-editor.org/info/rfc9112>.

[HTTP/2] Thomson, M., Ed. and C. Benfield, Ed., "HTTP/2", RFC 9113, DOI 10.17487/RFC9113, June 2022, <https://www.rfc-editor.org/info/rfc9113>.

[HTTP/3] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <https://www.rfc-editor.org/info/rfc9114>.

[IANA-POLICY] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[ICMP] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[ICMPv6] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[IPv6] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[IPv6-ZONE-ID] Carpenter, B., Cheshire, S., and R. Hinden, "Representing IPv6 Zone Identifiers in Address Literals and Uniform Resource Identifiers", RFC 6874, DOI 10.17487/RFC6874, February 2013, <https://www.rfc-editor.org/info/rfc6874>.

[PROXY-STATUS] Nottingham, M. and P. Sikora, "The Proxy-Status HTTP Response Header Field", RFC 9209, DOI 10.17487/RFC9209, June 2022, <https://www.rfc-editor.org/info/rfc9209>.

[QUIC] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[TCP] Eddy, W., Ed., "Transmission Control Protocol (TCP)", STD 7, RFC 9293, DOI 10.17487/RFC9293, August 2022, <https://www.rfc-editor.org/info/rfc9293>.

[TEMPLATE] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., and D. Orchard, "URI Template", RFC 6570, DOI 10.17487/RFC6570, March 2012, <https://www.rfc-editor.org/info/rfc6570>.

[URI] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

13.2. Дополнительная литература

[CONNECT-UDP] Schinazi, D., "Proxying UDP in HTTP", RFC 9298, DOI 10.17487/RFC9298, August 2022, <https://www.rfc-editor.org/info/rfc9298>.

[DPLPMTUD] Fairhurst, G., Jones, T., Tüxen, M., Rüngeler, I., and T. Völker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <https://www.rfc-editor.org/info/rfc8899>.

[ECN-TUNNEL] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/info/rfc6040>.

[HEv2] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

[IANA-PN] IANA, "Protocol Numbers", <https://www.iana.org/assignments/protocol-numbers>.

[IPSEC] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[IPv6-ADDR] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[OPTIMISTIC] Schwartz, B. M., "Security Considerations for Optimistic Use of HTTP Upgrade", Work in Progress, Internet-Draft, draft-schwartz-httpbis-optimistic-upgrade-00, 21 August 2023, <https://datatracker.ietf.org/doc/html/draft-schwartz-httpbis-optimistic-upgrade-00>.

[PROXY-REQS] Chernyakhovsky, A., McCall, D., and D. Schinazi, "Requirements for a MASQUE Protocol to Proxy IP Traffic", Work in Progress, Internet-Draft, draft-ietf-masque-ip-proxy-reqs-03, 27 August 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-masque-ip-proxy-reqs-03>.

[ROUTING-HDR] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <https://www.rfc-editor.org/info/rfc5095>.

[TUNNEL-SECURITY] Krishnan, S., Thaler, D., and J. Hoagland, "Security Concerns with IP Tunneling", RFC 6169, DOI 10.17487/RFC6169, April 2011, <https://www.rfc-editor.org/info/rfc6169>.

[UDP-USAGE] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

Благодарности

Разработка этого документа была вызвана обсуждениями [PROXY-REQS] в рабочей группе MASQUE. Авторы благодарны участникам этих дискуссий за их отклики. Отдельная благодарность Mike Bishop, Lucas Pardue, Alejandro Sedeño за ценные отклики на документ.

Большая часть текста о конфигурации клиентов основана на соответствующих частях [CONNECT-UDP].

Адреса авторов

Tommy Pauly (editor)
Apple Inc.
Email: tpauly@apple.com

David Schinazi
Google LLC
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America
Email: dschinazi.ietf@gmail.com

Alex Chernyakhovsky
Google LLC
Email: achernya@google.com

Mirja Kühlewind
Ericsson
Email: mirja.kuehlewind@ericsson.com

Magnus Westerlund
Ericsson
Email: magnus.westerlund@ericsson.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru