Обеспечение совместимости протоколов IPv4 и IPv6: бесконтекстный IP/ICMP транслятор в среде Linux[1]

В.З.Шнитман, А.А.Ломака

Аннотация. В связи с исчерпанием адресного пространства и некоторыми другими проблемами протокола IPv4 возникла необходимость в переходе на протокол следующего поколения IPv6. Однако, поскольку установленная база программного и аппаратного обеспечения IPv4 невероятно велика, возникает проблема обеспечения обратной совместимости. Так же существует проблема развертывания новых сетей, основанных на протоколе IPv6. Имеется несколько возможных механизмов решения этих проблем. В данной статье рассмотрена реализация одного из таких механизмов – механизма бесконтекстной трансляции (*Stateless IP/ICMP translator*

1. Введение

Как известно, протокол IPv4 является основой сети INTERNET. Сегодня в мире существуют миллионы машин с установленным стеком протокола IPv4. Однако протокол IPv4 обладает и существенными недостатками, а именно:

- ограниченность адресного пространства
- отсутствие механизма автоматической конфигурации адресов
- проблема перенумерации машин
- проблемы, связанные с механизмом фрагментации
- слабая расширяемость протокола
- отсутствие поддержки качества обслуживания
- проблема безопасности коммуникаций

Рассмотрим коротко некоторые из этих недостатков.

[1] Данная работа выполнена по гранту РФФИ 99-90220.

1.1. Ограниченность адресного пространства

Экспоненциальный рост числа машин, подключенных к сети INTERNET, ведет к быстрому исчерпанию адресного пространства IPv4. Адрес IPv4 имеет длину 32 бита, что дает максимум 4294967296 адресов. Адресное пространство IPv4 не однородно. Существует 5 видов IPv4 адресов: класс A, класс B, класс C, класс D и класс E.

Рис. 1. Форматы адресов IPv4

Любой адрес класса A,B или C глобально идентифицирует один интерфейс в сети INTERNET, либо все интерфейсы некоторой сети (подсети). Адреса класса D (так называемые групповые адреса) используются для идентификации нескольких (т.е. группы) интерфейсов в пределах своей области действия. Адреса класса E зарезервированы и не используются. Таким образом, реальную основу адресации в INTERNET составляют классы A,B и C, т.е. максимум 3758096384 адресов. Эта цифра есть теоретический предел числа хостов в сети INTERNET, в реальности, однако, неэффективность распределения на начальном этапе существенно снизила эту цифру. По расчетам IETF пул глобальных адресов исчерпается примерно в 2005-м – 2011-м году. Следует однако ожидать более быстрого исчерпания адресного пространства в связи появлением и широким распространением портативных устройств, подключенных к глобальным сетям (персональные цифровые помощники, интернет-приставки, электронные записные книжки и т.д.). Стоит также заметить, что применение технологии трансляции адресов (NAT - network address translation) позволяет в некоторых случаях ослабить проблему нехватки адресов.

Адресация IPv4 обладает и другим недостатком - слабой агрегацией адресов, что приводит к катастрофичекому росту таблиц маршрутизации в маршрутизаторах сетей, не имеющих маршрута по умолчанию. Такие маршрутизаторы обязаны знать пути ко всем существующим в INTERNET сетям. Известно, что основу распределения адресов сейчас составляют сети класса C, которых может быть максимум 2097152. Таким образом, магистральные маршрутизаторы INTERNET могут потенциально содержать таблицы маршрутизации, состоящие из миллионов записей. Бесперспективность построения таких маршрутизаторов становится очевидной, если заметить, что вся таблица маршрутизации просматривается при обработке каждого пакета. Для решения этой проблемы в настоящее время используются методы бесклассовой междоменной маршрутизации (CIDR – Classless Inter-Domain Routing).

1.2. Отсутствие механизма автоматической конфигурации адресов

Изначально, с момента создания, в протокол IPv4 не было заложено механизма автоматического назначения адресов хостам сети (интерфейсам хостов). Эта операция обычно проводится сетевым администратором вручную либо полуавтоматически с использованием таких средств, как протоколы DHCP, RARP или BOOTP. Эта процедура является трудоемкой даже в малых сетях класса C, а в больших сетях вручную попросту невозможна.

1.3. Проблема перенумерации машин

С данной проблемой организации сталкиваются при изменении INTERNET провайдера. Она состоит в полном изменении всех IP адресов всех хостов корпоративной сети. При отсутствии механизма автоматического назначения адресов эта перенумерация должна проводится вручную администратором сети. Это черезвычайно трудоемкая операция, учитывая количество машин в современных корпоративных сетях. Для устранения этого недостатка протокола IPv4 был разработан протокол Dynamic Host Configuration Protocol (DHCPv4). Но поскольку это отдельный протокол, не являющийся частью стандарта IPv4, то его реализация имеется далеко не во всех операционных системах, а поэтому не может считаться адекватным решением проблемы управления адресами в сети.

1.4. Проблемы, связанные с механизмом фрагментации

Одной из функций протоколов сетевого уровня является фрагментация слишком больших дейтаграмм перед посылкой их к следующему узлу. Дейтаграммы протокола IPv4 могут теоретически достигать размера 65535 байт, в то время как существующие сетевые технологии ограничивают максимальный размер

передаваемых пакетов несколькими тысячами байт (типичный пример Ethernet – 1500 байт). Для передачи больших дейтаграмм протокол IPv4 фрагментирует их на несколько более мелких, при этом фрагментацию может осуществлять как отправитель, так и любой маршрутизатор на пути следования дейтаграммы. Возможность (и необходимость) фрагментации дейтаграмм промежуточными маршрутизаторами в IPv4 ограничивает производительность этих маршрутизаторов, так как процедура фрагментации является очень дорогой с точки зрения потребляемых ресурсов маршрутизатора. Фрагментации в промежуточных узлах можно избежать, но для этого отправитель должен определять размер максимального блока передачи данных к конкретному получателю на пути, по которому пойдет пакет (PMTU – Path MTU), и фрагментировать пакет в соответствии с полученным значением PMTU. Для вычисления значения PMTU в протоколе IPv4 существует соответствующий механизм - Path MTU discovery, однако его применение не является обязательным для протокола IPv4.

1.5. Слабая расширяемость протокола

В протоколе IPv4 предусмотрен единственный механизм расширения: добавление к заголовку IPv4 дополнительных опций. Однако общая длина всех опций не может превышать 40 байт, что крайне мало в современных условиях.

1.6. Отсутствие поддержки качества обслуживания

Со времени создания протокола IPv4 появились новые сетевые приложения, такие как Streaming Audio, Streaming Video и т.д. Для нормальной работы этим приложениям требуется гарантированное обеспечение таких параметров передачи данных, как пропускная способность, задержка и вариация задержки. Набор таких параметров получил название качества услуг. Протокол IPv4 не может обеспечить предоставление гарантированного качества услуг. Для этой цели в заголовке IPv4 служит поле “Type of service”, но ни механизм интерпретации этого поля, ни механизм резервирования необходимых сетевых ресурсов в IPv4 определены не были, поэтому абсолютное большинство существующих маршрутизаторов попросту игнорируют это поле в заголовке IPv4.

1.7. Проблема безопасности коммуникаций

Повсеместное распространение компьютерных сетей привело к необходимости разграничения доступа к информации, находящейся в этих сетях. Применительно к протоколам, одним из наиболее удобных мест в семиуровневой модели взаимодействия открытых систем ISO/OSI, в котором можно расположить систему безопасности, является сетевой уровень. К сожалению, в протоколе IPv4 не предусмотрено каких-либо средств организации безопасности передачи данных.

Все эти недостатки IPv4 и привели к необходимости разработки IP протокола следующего поколения, который получил название IPv6 или IPng (next generation). В этом протоколе были учтены все недостатки протокола IPv4, а так же были добавлены некоторые новые возможности, повышающие его эффективность и удобство использования. Разработка этого протокола продолжается и по сей день, кроме того для его испытаний в реальных условиях была создана экспериментальная IPv6 сеть 6bone.

Каким бы привлекательным ни был протокол IPv6, очевидно, что нельзя "за одну ночь" перевести INTERNET на протокол IPv6. Гигантская база разработанного и установленного программного и аппаратного обеспечения IPv4 требует сохранения обратной совместимости IPv6 с IPv4. Такие механизмы совместимости начали разрабатываться одновременно с разработкой самого протокола IPv6. Более того, наличие таких механизмов было одним из критериев выбора среди кандидатов на роль IP протокола следующего поколения. Таких механизмов в IPv6 существует несколько, и данная работа есть реализация одного из таких механизмов.

2. Краткий обзор протокола IPv6

Данный обзор не претендует на полноту и не преследует своей целью полностью ознакомить читателя с IPv6.
Создатели IPv6 взяли за основу протокол IPv4 и упростили его путем внесения следующих изменений:

- был определен механизм создания расширенных заголовков IPv6
- все редко используемые возможности протокола IPv4 были вынесены в расширенные заголовки IPv6
- был изменен формат адреса, и вообще всей архитектуры адресации. Размер адреса IPv6 был выбран равным 128 битам, чтобы удовлетворить ближайшим и будущим потребностям, даже с учетом существующей неэффективности распределения адресов.
- базовый заголовок IPv6 был изменен таким образом, чтобы максимально ускорить его обработку. В частности, все оставшиеся поля были выровнены по своим естественным границам, и из базового заголовка IPv6 была исключена контрольная сумма.
- архитектура адресации была создана таким образом, чтобы обеспечить возможность агрегации маршрутной информации. Это необходимо для эффективного функционирования маршрутизации.
- была изменена семантика фрагментации

- был изменен минимально допустимый размер максимального блока передачи данных (MTU)
- была исключена возможность широковещания

Базовый заголовок протокола IPv6 имеет следующий вид:

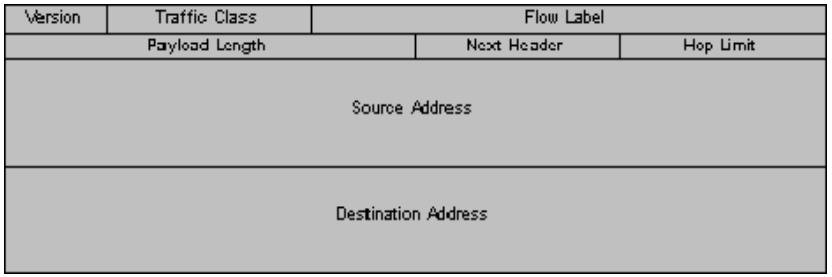

Рис. 2. Базовый заголовок IPv6

Отдельные поля имеют следующее значение:

Version	- поле версии протокола, для IPv6 равно 6. Имеет длину 4 бита.
Traffic class	- поле класса трафика; поле определяет уровень качества услуг, который необходимо предоставить при обработке этого пакета. Имеет такое же значение, что и поле Type of service заголовка IPv4. Определение этого поля дается в отдельном документе: "Differenciated services" (RFC 2474). Механизм использования этого поля пока еще находится в стадии разработки. Имеет длину 8 бит.*
Flow Label	- метка потока; все пакеты, принадлежащие одному и тому же потоку, идентифицируются отправителем одной и той же меткой и обрабатываются промежуточными маршрутизаторами одинаковым образом, т.е. им предоставляется одинаковый уровень качества обслуживания. В данный момент механизм использования этого поля не достаточно определен. Имеет длину 20 бит.
Payload Length	- длина полезной нагрузки в октетах, т.е. длина всего пакета кроме самого базового (и только базового) заголовка IPv6. Имеет длину 16 бит.
Next Header	- тип следующего заголовка, им может быть либо один из расширенных заголовков IPv6, либо заголовок протокола верхнего уровня, такой как TCP, UDP и другие. Имеет длину 8 бит.

Hop Limit	- это поле уменьшается на 1 каждым маршрутизатором по пути следования пакета. Пакет сбрасывается, если это поле становится равным нулю. Имеет длину 8 бит.
Source Address	- адрес отправителя. Имеет длину 128 бит.
Destination Address	- адрес получателя. Это необязательно адрес конечного получателя. Например, если в пакете есть расширенный заголовок Routing Header, то в этом поле будет находиться адрес следующего узла. Имеет длину 128 бит.

---

* В самом начале в протоколе IPv6 не было этого поля, потом его ввели, размером 4 бита; затем для унификации с протоколом IPv4 размер поля увеличили до 8 бит, и согласовали его формат с форматом поля “Type of Service” протокола IPv4.

Для сравнения ниже приведен формат заголовка IPv4:

Рис. 3. Формат заголовка IPv4

Version	- поле версии протокола, для IPv4 равно 4. Имеет длину 4 бита.
IHL	- длина заголовка IPv4 в 32-х битных словах. Имеет длину 4 бита.
Type of Service	- поле класса трафика; поле определяет уровень качества услуг, который необходимо предоставить при обработке этого пакета. Подавляющим большинством маршрутизаторов это поле игнорируется. Определение этого поля дается в отдельном документе: "Differenciated services" (RFC 2474). Имеет длину 8 бит.*
Total Length	- общая длина пакета в октетах, включая сам заголовок. Имеет длину 16 бит.
Identification	- уникальный идентификатор, присваеваемый всем фрагментам одного исходного пакета, подвергшегося фрагментации. Используется получателем при сборке исходного пакета по его фрагментам. Имеет длину 16 бит.

Flags	- поле флагов. Содержит флаги DF (don't fragment) - запрет на фрагментацию этого пакета, и MF (more fragments) - флаг, указывающий на то, что этот фрагмент является не последним и есть еще фрагменты.
Fragment Offset	- смещение фрагмента в октетах от начала исходного пакета. У всех фрагментов, кроме последнего, поле Fragment Offset должно быть кратно 8. Имеет длину 13 бит.
Time to Live	- максимальное время жизни пакета в секундах. Любой промежуточный маршрутизатор обязан уменьшить это поле минимум на 1 независимо от того, сколько времени пакет обрабатывался этим маршрутизатором. Пакет сбрасывается, если это поле становится равным нулю. Имеет длину 8 бит.
Protocol	- тип следующего заголовка, например, TCP, UDP и так далее. Имеет длину 8 бит.
Header Checksum	- контрольная сумма заголовка IPv4. Имеет длину 16 бит.
Source Address	- адрес отправителя. Имеет длину 32 бита.
Destination Address	- адрес получателя. Это адрес необязательно является адресом конечного получателя. Например, если у заголовка есть опция Source routing, то в этом поле будет находиться адрес следующего узла. Имеет длину 32 бит.

---

* До сих пор не существует стандартизированного способа интерпретации этого поля. В IETF ведется разработка двух архитектур определения качества обслуживания и механизмов его реализации: “Differenciated Services” и “Integrated Services”. Эти архитектуры являются протокольно независимыми, т.е. действуют как на протокол IPv4, так и на протокол IPv6.

Адрес IPv6 имеет длину 128 бит, что дает гигантское адресное пространство. IPv6 адрес, так же как и IPv4 адрес, имеет тип. Тип адреса определяет, что идентифицирует собой этот адрес. Так глобальные unicast адреса идентифицируют одиночные интерфейсы.

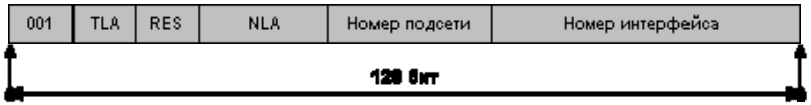

Рис 4. Пример глобального unicast IPv6 адреса

Этот тип адресов составляет 1/8 общего адресного пространства IPv6. Кроме типа IPv6 адрес имеет еще область действия, она определяет ту часть топологии сети, в которой этот адрес однозначно идентифицирует свой объект. Областью действия может быть:

- глобальная сеть INTERNET
- линк - прямое, немаршрутизируемое соединение хостов, например, сегмент сети Ethernet
- сайт – сеть, находящаяся под одним административным управлением, т.е. административный домен

Адресное пространство IPv6 настолько большое, что в нем было предусмотрено место для отображения адресных пространств других протоколов, в частности, было предусмотрено место для отображения IPv4 адресов и IPX адресов. Это позволяет разрабатывать механизмы совместимости протокола IPv6 с другими протоколами.

Так же, как и в IPv4, все unicast адреса IPv6 состоят из номера хоста, номера сети и префикса типа. Номер хоста имеет длину 64 бита. Поля номера сети и префикса формата имеют переменную длину.

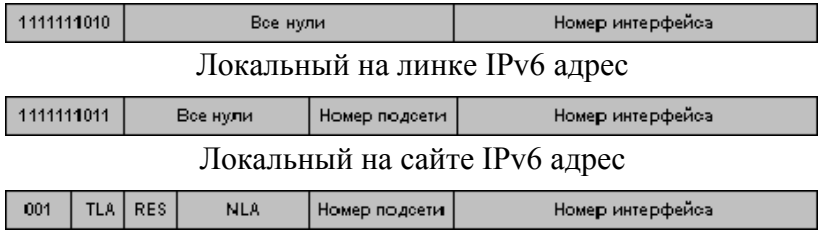

Рис. 5. Unicast адреса

Как видно из рисунка, у глобальных IPv6 адресов номер сети структурирован и состоит из следующих полей:

TLA - агрегат первого уровня, отдельные TLA могут выдаваться только очень крупным первичным операторам связи INTERNET. Имеет длину 13 бит.

RES - зарезервированное поле. Имеет длину 8 бит.

NLA - агрегат следующего уровня, этих агрегатов гораздо больше, чем агрегатов TLA. Первичные операторы связи могут выдавать диапазоны идентификаторов NLA вторичным операторам, которые, в свою очередь, могут выдавать отдельные NLA конкретным организациям. Имеет длину 24 бита.

Subnet ID - номер подсети. Организация, получившая в свое распоряжение отдельный префикс NLA (точнее номер сети), вольна разбить его на несколько подсетей по своему усмотрению. Всего можно организовать до 65535 подсетей с фактически неограниченным числом хостов в каждой из них, так как номер хоста занимает 64 бита.

Эти поля введены в IPv6 для организации агрегации маршрутной информации. Маршрутизация осуществляется с использованием так называемых префиксов. Префикс – это битовая строка, которая побитно сравнивается со старшими битами адреса назначения. Префикс может быть частичным, т.е. иметь длину менее длины адреса IPv6, а может быть полным, т.е. быть адресом IPv6. Маршрутная запись состоит как минимум из поля префикса, адреса следующего узла и поля метрики. Маршрутная таблица состоит из маршрутных записей. При определении исходящего интерфейса для пакета производится поиск префикса максимальной длины, соответствующего адресу назначения пакета. Если совпадение отсутствует, то пакет отсылается по так называемому маршруту по умолчанию (если он есть). Если найдено совпадение, то пакет отсылается по адресу, выбираемому из соответствующей записи таблицы маршрутизации.

Путем иерархического построения адреса IPv6 на основе полей TLA и NLA и при соответствующей организации распределения префиксов TLA и NLA можно предотвратить бесконтрольный рост таблиц маршрутизации магистральных маршрутизаторов сети INTERNET.

В отличие от IPv4, протокол IPv6 имеет встроенные средства назначения адресов хостам (интерфейсам хостов). Такие средства предусматривают регистрацию за хостом одного или нескольких адресов на определенный срок, процедуру "мягкого изъятия" адресов, срок регистрации которых истек, и процедуру продления регистрации адресов. И все это делается без участия человека. Возможно также использование традиционных средств конфигурирования хостов, таких, как протокол DHCP (DHCPv6) и ручное конфигурирование.

В IPv6 так же явно выделено понятие маршрутизатора, чего нет в IPv4, благодаря чему конечным хостам не нужно "подсматривать" за протоколами маршрутизации для выполнения автоматической настройки параметров маршрутизации.

Протокол IPv6 не имеет средств широковещания. Это важное его свойство способствует повышению эффективности использования полосы пропускания. Типичный пример использования широковещания: протокол ARP – составная часть IPv4. При большом числе подключенных к линку хостов (например,

сегмент Ethernet с большим числом подсоединенных хостов) протокол ARP создает так называемые широковещательные штормы, парализующие этот линк.

Протокол IPv6 является расширяемым. Кроме базового заголовка IPv6, в пакете могут присутствовать так называемые расширенные заголовки IPv6. Все расширенные заголовки обрабатываются хостами и маршрутизаторами строго последовательно, в том порядке, в каком они встречаются в пакете. Такими расширенными заголовками являются:

- routing header (заголовок маршрутизации) - имеет тот же смысл, что и source routing опция заголовка IPv4
- destination options - этот заголовок содержит одну или несколько опций и может обрабатываться только получателем пакета
- hop-by-hop options - тоже самое, но обрабатывается всеми промежуточными узлами

Протокол IPv6 требует, чтобы у всех промежуточных каналов размер максимального блока передачи данных (MTU-maximum transfer unit) был как минимум 1280 байт, что отличается от 576 байт для протокола IPv4.

Кроме того, IPv6 отличается от IPv4 в вопросе фрагментации больших пакетов. В IPv4 фрагментацию может производить как отправитель, так и любой промежуточный узел. В IPv6 фрагментацию может производить только хост - отправитель, для этого ему нужно знать MTU пути, по которому пойдет пакет. С этой целью специальная процедура - Path MTU discovery - является составной частью протокола IPv6.

В протокол IPv6 заложены возможности по организации защищенной передачи данных через сети. Основу этих возможностей составляют расширенные заголовки AH - authentication header, заголовок аутентификации, и ESP - encapsulated security payload, заголовок, инкапсулирующий в себе зашифрованный пакет - полезную нагрузку. Способ работы с этими заголовками не является частью самого протокола IPv6, а определяется другими стандартами, такими, как IPsec.

3. Механизмы совместимости IPv6 с IPv4

Существующие подходы к проблеме совместимости IPv6 с IPv4 можно разделить на две категории:

- механизмы, обеспечивающие работу наложенных IPv6 сетей поверх существующих IPv4 сетей – эти механизмы обеспечивают взаимодействие IPv6 хостов, используя в качестве среды передачи существующую сеть IPv4
  механизмы, обеспечивающие взаимодействие IPv6 и IPv4 хостов

К первой категории относятся механизмы туннелирования, а среди механизмов, относящихся ко второй категории, следует отметить двойной стек, шлюз прикладного уровня и IP/ICMP трансляцию. Существует несколько расширений механизма туннелирования и механизма трансляции заголовков, призванные упростить их использование, но эти расширения не меняют семантики соответствующего механизма. Опишем вкратце эти технологии.

Двойной стек - в этом случае на каждом IPv6 хосте, которому требуется взаимодействие с IPv4 хостами, устанавливается стек протокола IPv4 и ему выделяется IPv4 адрес. После этого этот хост может взаимодействовать как с IPv4 хостами, так и с IPv6 хостами. Данный метод является самым простым и самым радикальным методом решения проблемы совместимости, однако обладает некоторыми недостатками. Во-первых, он требует установки дополнительного программного обеспечения и его конфигурирования на каждом хосте, что выливается в дополнительную работу сетевому администратору и в повышенные требования к ресурсам хостов. Во-вторых, он требует, чтобы все промежуточные маршрутизаторы могли работать как с протоколом IPv4, так и с протоколом IPv6. Другим существенным недостатком такого "чистого" решения проблемы совместимости является необходимость переписывания всего парка прикладного программного обеспечения для того, чтобы оно смогло работать поверх протокола IPv6. Не стоит и говорить даже о том, что такая задача потребует огромного количества времени и усилий.

Шлюз прикладного уровня (**ALG** - application level gateway) - данный метод предполагает, что для каждого используемого сетевого приложения создается специальное прикладное программное обеспечение, осуществляющее преобразование трафика этого сетевого приложения из трафика IPv4 в трафик IPv6, и наоборот. Недостатки этого метода очевидны: сколько существует сетевых приложений, столько же необходимо создать соответствующих ALG-шлюзов. Следует, однако, заметить, что при организации взаимодействия IPv4 и IPv6 сетей полностью избежать построения ALG-шлюзов, вероятно, не удастся, например, в случае использования службы доменных имен.

Туннелирование - данный метод предназначен для создания IPv6 туннелей сквозь существующие IPv4 сети (в частности INTERNET), не поддерживающие протокол IPv6. Такие туннели создаются вручную либо автоматически различными способами и объединяют отдельные IPv6 сети между собой. Пакеты IPv6, входя в такой туннель, инкапсулируются в пакеты IPv4 и пересылаются по IPv4 сети на другой конец туннеля. Там они деинкапсулируются и обрабатываются далее как обычные IPv6 пакеты. На основе таких туннелей, в частности, построена экспериментальная глобальная IPv6 сеть **6bone**. Данное решение проблемы совместимости является частичным, оно обеспечивает создание наложенных IPv6 сетей поверх существующей сетевой инфраструктуры. Оно не обеспечивает взаимодействия IPv4 хостов с IPv6 хостами. В настоящий момент именно этот

механизм получил наибольшую поддержку среди разработчиков и пользователей в основном потому, что он является основой построения экспериментальной сети **6bone.**

IP/ICMP трансляция может быть реализована двумя основными методами: контекстным и бесконтекстным. Механизм бесконтекстной IP/ICMP трансляции предполагает установку на границе IPv6 сети специального агента, осуществляющего трансляцию протоколов. При этом IPv6 хостам присваиваются специальные, так называемые IPv4-транслированные, адреса. Приходящие извне IPv4 пакеты перенаправляются этому агенту, проходя который, они подвергаются преобразованию к формату протокола IPv6 и пересылаются далее к своим получателям. Ответные пакеты, идущие от IPv6 хостов к IPv4 хостам (это индицируется специальным типом IPv6 адреса назначения), так же должны пройти через IP/ICMP транслятор, но необязательно через тот же самый, так как сам транслятор является бесконтекстным. Пройдя транслятор, IPv6 пакеты становятся IPv4 пакетами и доставляются по назначению. Удобством этой схемы является ее прозрачность для взаимодействующих хостов и полная бесконтекстность, что существенно облегчает ее реализацию и использование. Данный механизм является относительно новым.

При использовании контекстного метода трансляции адресов каждому IPv4 адресу, который может участвовать в процессе трансляции, т.е. каждому возможному IPv4 адресу назначения и каждому возможному IPv4 адресу отправителя, ставится в соответствие некоторый IPv6 адрес. Такая схема позволяет использовать “настоящие” IPv6 адреса, например, те, которые хосты получают в процессе автоматической конфигурации адресов. Однако за такое удобство приходится платить ограничением в числе IPv4 хостов, с которыми возможно взаимодействие. Действительно, каждому IPv4 хосту, с которым потенциально необходимо взаимодействовать, нужно выделить некоторый IPv6 адрес, который будет идентифицировать этот IPv4 хост в IPv6 части сети. Если таких IPv4 хостов мало, то это сделать нетрудно. Однако, если, например, потенциально возможно взаимодействие с любым хостом сети INTERNET, то такое отображение создать просто невозможно.

Использование нормальных, а не специальных IPv6 адресов, приводит так же к необходимости подвергать трансляции заголовок транспортного уровня. Действительно, протоколы верхних уровней, например, TCP и UDP, используют псевдозаголовок IP при вычислении своих контрольных сумм. Специальные адреса IPv6 подобраны таким образом, что не меняют значение контрольной суммы при трансляции, в то время как нормальные IPv6 адреса таким свойством не обладают. Поэтому при построении контекстного транслятора необходимо корректировать значения контрольных сумм в транспортных заголовках. Такая схема имеет и побочный эффект: транслятор становится не способным работать с любыми транспортными протоколами, кроме TCP и UDP. Применение механизма бесконтекстной трансляции позволяет избежать такой зависимости.

Ниже приведено детальное описание работы бесконтекстного IP/ICMP транслятора, за которым следует описание его реализации.

4. Бесконтекстный межпротокольный IP/ICMP транслятор

Межпротокольный транслятор работает на сетевом уровне семиуровневой модели взаимодействия открытых систем OSI/ISO и осуществляет преобразование заголовков IPv6<->IPv4, не изменяя при этом заголовков и данных протоколов вышестоящих уровней (за исключением протокола ICMP).

4.1. Область применимости

Предполагается, что механизм межпротокольной трансляции будет использован на раннем этапе перехода к протоколу IPv6, а в дальнейшем его применение будет ограничено. Схема IP/ICMP трансляции является односторонней в том смысле, что она предназначена для интеграции IPv6 сетей с IPv4 Internet, а не наоборот. Это выражается в требовании выделения IPv6 хостам IPv4 адресов (точнее IPv4-транслированных IPv6 адресов, полученных на основе выделенных IPv4 адресов). Действительно, для того, чтобы IPv6 хост мог послать пакет IPv4 хосту, ему необходим его IPv6 адрес, который он пропишет в поле Destination Address пакета IPv6. Аналогично, для того чтобы послать пакет, IPv4 хосту необходим IPv4 адрес назначения (destination address). Для этих целей механизм бесконтекстной трансляции требует выделения диапазона IPv4 адресов (т.е. подсети) IPv6 хостам и выделения из этого диапазона каждому IPv6 хосту IPv4 адреса.

В дальнейшем, когда будут преобладать IPv6 сети, необходимость в бесконтекстной IP/ICMP трансляции отпадет, и, возможно, встанет проблема совместимости оставшихся IPv4 сетей с IPv6 сетями. Для этой цели схема бесконтекстной IP/ICMP трансляции не подходит, так как иначе пришлось бы присваивать IPv4 адреса удаленным IPv6 хостам, в том числе и находящимся под другим административным управлением.

Свойство односторонности механизма трансляции проистекает из такого фундаментального факта, как различная емкость адресных пространств у протоколов IPv6 и IPv4. Действительно, адрес IPv6 имеет 128 бит, в то время как адрес IPv4 имеет только 32 бита. Это различие приводит к невозможности взаимно однозначного отображения адресных пространств этих протоколов. Отобразить 32 битный адрес IPv4 в 128 битный адрес IPv6 можно, а вот сделать наоборот нельзя. Именно в силу этого факта транслятор может быть использован для организации взаимодействия локальной сети IPv6 (т.е. ограниченного числа хостов, находящихся под одним административным управлением) с IPv4 Internet, и не может быть использован для организации взаимодействия локальной IPv4 сети с IPv6 Internet.

Механизм межпротокольной трансляции предполагает, что транслятор находится на границе сетей IPv4 и IPv6 так, как это показано на рисунке:

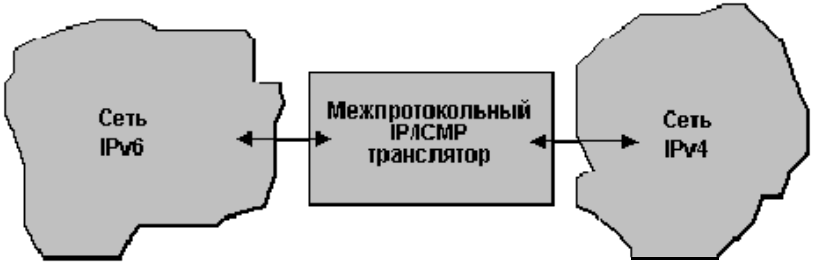

Рис. 6. Модель предполагаемой конфигурации сети

Весь подлежащий трансляции трафик должен проходить через IP/ICMP транслятор.

Транслятор называется бесконтекстным потому, что в процедуре трансляции не используется никакой сохраненной ранее информации. Транслятор получает на входе пакет, проверяет, удовлетворяет ли этот пакет критериям трансляции, и затем конвертирует его, используя при этом только ту информацию, которая содержится в самом пакете. При необходимости между IPv4 и IPv6 сетями можно установить более одного транслятора, и это никак не скажется на корректности функционирования схемы трансляции.

В отличие от бесконтекстного транслятора, контекстный транслятор осуществляет преобразование адресов пакетов на основании заранее предопределенной таблицы соответствия адресов. Такая таблица содержит соответствие "IPv6 адрес – IPv4 адрес" и должна создаваться вручную. При наличии в сети более одного транслятора необходимо, чтобы их таблицы соответствия были согласованными. Вместо этого, путем использования специальных IPv6 адресов, схема бесконтекстной IP/ICMP трансляции позволяет организовать взаимодействие без необходимости создания таблиц контекстов.

4.2. Адреса, используемые при трансляции

Протоколы верхних уровней, такие, как TCP и UDP, для защиты данных от искажения используют контрольные суммы, которые вычисляют на основании данных пакета, своего заголовка и псевдозаголовка протокола нижележащего уровня, т.е. IPv4 или IPv6. В этот псевдозаголовок входят адреса отправителя, получателя, идентификатор протокола верхнего уровня и размер блока данных этого протокола верхнего уровня. Последние два значения не изменяются при трансляции. При трансляции изменяются только адреса. Для того, чтобы при трансляции избежать необходимости пересчитывать контрольные суммы

заголовков протоколов верхних уровней, используются IPv6 адреса специальных типов, причем такие, которые не влияют на значение контрольной суммы.

При трансляции используются следующие специальные виды IPv6 адресов:

- IPv6 хосты должны иметь ***IPv4-транслированный IPv6 адрес***. Этот адрес соответствует префиксу ::FFFF:0:0:0/96, т.е. старшие 96 бит этого адреса таковы: 0000:0000:0000:0000:FFFF:0000. Младшие 32 бита адреса содержат IPv4 адрес, присвоенный хосту:

Рис. 7. Формат IPv4-транслированного IPv6 адреса

- ***IPv4-отображенный IPv6 адрес*** используется IPv6 хостами для посылки пакетов IPv4 хостам. Старшие 96 бит этого адреса содержат префикс 0000:0000:0000:0000:0000:FFFF, младшие 32 бита адреса содержат IPv4 адрес хоста, для которого предназначен пакет:

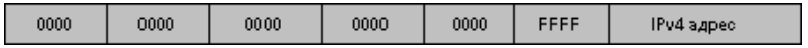

Рис 8. Формат IPv4-отображенного IPv6 адреса

4.3. Маршрутизация в модели предполагаемой сети

Как было сказано выше, транслятор должен располагаться на границе IPv4 и IPv6 сетей. IPv6 хостам, для которых требуется возможность взаимодействия с IPv4 хостами, должны быть присвоены IPv4-транслированные адреса. Вопросы присвоения таких адресов IPv6 хостам, а так же установления соответствующей маршрутизации не определяются самой технологией трансляции. Весь трафик, подлежащий трансляции, должен проходить через IP/ICMP транслятор. Для этого необходимо, чтобы в IPv4 сети пакеты, идущие по IPv4 адресам, выделенным IPv6 хостам, пересылались на IP/ICMP транслятор, а в IPv6 сети на IP/ICMP транслятор попадали пакеты, идущие по IPv4-отображенным IPv6 адресам.

4.4. Трансляция пакетов протокола IPv4 в формат IPv6

4.4.1. Общая схема

При приходе нормального, нефрагментированного пакета IPv4, его данные копируются без изменения, а новый IPv6 заголовок строится следующим образом:

Version: 6

Traffic Class: Копируется из поля “Type of Service” заголовка IPv4

Flow Label: 0 (все нули)

Payload Length: Значение поля “Total Length” заголовка IPv4 минус размер самого заголовка и IPv4 опциий, если они присутствуют

Next Header: Копируется из поля “Protocol” заголовка IPv4

Hop Limit: Значение поля “TTL” плюс 1, так как пакет проходит через два стека: сначала стек IP6, затем стек IPv4 (или наоборот, см. описание реализации)

Source Address: В младшие 32 бита копируется IPv4 source address, в старшие 96 бита копируется префикс: 0000:0000:0000:0000:0000:FFFF

Destination Address: В младшие 32 бита копируется IPv4 destination address, в старшие 96 бита копируется префикс: 0000:0000:0000:0000:FFFF:0000

Если в пакете присутствуют IPv4 опции, то они игнорируются.

4.4.2. Обработка фрагментов

Если в пакете сброшен флаг DF (don't fragment), либо пакет является фрагментом, то при трансляции к базовому заголовку IPv6 необходимо добавить расширенный заголовок Fragment Header. Поля выставляются так, как описано выше, за следующим исключением:

IPv6 заголовок:

Payload Length: “Total length” заголовка IPv4 плюс 8, минус размер IPv4 заголовка с опциями

Next Header: 44 (fragment header)

Fragment Header:

Next Header: Копируется из поля “Protocol” заголовка IPv4

Fragment Offset: Копируется из поля “Fragment Offset” заголовка IPv4

M flag: Копируется флаг “More Fragments” заголовка IPv4

Identification: Копируются младшие 16 бит поля “Identification” заголовка IPv4

4.4.3. Трансляция информационных сообщений ICMPv4

Из всех возможных информационных ICMPv4 сообщений трансляции подлежат только сообщения Echo и Echo Reply. Они транслируются в ICMPv6 сообщения Echo Request и Echo Reply, соответственно. При трансляции так же пересчитывается поле контрольной суммы, в вычисление которого включается псевдозаголовок IPv6. Данные пакета копируются без изменений.

4.4.4. Трансляция сообщений ICMPv4 об ошибках

Любое ICMPv4 сообщение об ошибке содержит в себе пакет, вызвавший ошибку (точнее столько байт этого пакета, чтобы суммарная длина пакета сообщения об ошибке не превышала 576 байт). Поэтому необходимо транслировать не только сам заголовок ICMPv4, но и пакет, вызвавший ошибку, т.е. производить рекурсивную трансляцию. Трансляция заголовка ICMPv4 происходит по следующей схеме:

Destinaton Unreachable (Type 3):

если не определено иначе, поле Type выставляется равным 1

Code 0 и 1: выставить поле Code равным 0 (no route to destination)

Code 2: транслировать в сообщение ICMPv6 Parameter Problem (Type 4, Code 1), в котором выставить поле Pointer на IPv6 поле Next Header

Code 3: выставить поле Code равным 4 (port unreachable)

Code 4: транслировать в сообщение ICMPv6 Packet Too Big (Type 2) с Code равным 0. Поле MTU необходимо скорректировать на разницу между размером заголовка IPv4 и IPv6

Code 5: выставить поле Code равным 2 (not a neighbor)

Code 6 и 7: выставить поле Code равным 0 (no route to destination)

Code 8: выставить поле Code равным 0 (no route to destination)

Code 9 и 10: выставить поле Code равным 1 (communication with destination administratively prohibited)

Code 11 и 12: выставить поле Code равным 0 (no route to destination)

Time Exceeded (Type 11):

выставить поле Type равным 3, поле Code остается без изменения

Parameter Problem (Type 12):

выставить поле Type равным 4. Поле Pointer должно указывать на соответствующее поле в транслированном IP заголовке

Все остальные сообщения об ошибках должны просто сбрасываться.

4.5. Трансляция пакетов протокола IPv6 в формат IPv4

4.5.1. Общая схема

При приходе пакета IPv6, не содержащего заголовка фрагментации, его данные копируются без изменения, а новый IPv6 заголовок строится следующим образом:

Version:	4
Internet Header Length:	5
Type of Service and Precedence:	Копируется из поля “Traffic Class” заголовка IPv6
Total Length:	Значение поля “Payload Length” заголовка IPv6, плюс размер IPv4 заголовка
Identification:	0
Flags:	MF выставляется в 0, DF выставляется в 1
Fragment Offset:	0
Time to Live:	Значение поля “Hop Limit” плюс 1, так как пакет проходит через два стека: сначала стек IP6, затем стек IPv4 (или наоборот, см. описание реализации)
Protocol:	Значение поля “Next Header” заголовка IPv6

Header Checksum:	Пересчитывается после заполнения остальных полей
Source Address:	Если IPv6 Source address есть IPv4-транслированный IPv6 адрес, то из него копируются младшие 32 бита, иначе это поле выставляется равным 127.0.0.1
Destination Address:	Копируются младшие 32 бита IPv6 адреса назначения

Если в пакете присутствуют расширенные заголовки, то они просто игнорируются.

4.5.2. Обработка заголовка фрагментации

При наличии в исходном IPv6 пакете заголовка фрагментации IPv4 заголовок строится так, как показано выше, за следующим исключением:

Total Length:	Значение поля “Payload Length” заголовка IPv6 минус размер заголовка Fragment Header, плюс размер заголовка IPv4
Identification:	Копируются младшие 16 бит поля “Identification” заголовка Fragment Header
Flags:	DF выставляется в 0, MF копируется из флага M заголовка Fragment Header
Fragment Offset:	Копируется из поля “Fragment Offset” заголовка Fragment Header
Protocol:	Копируется из поля “Next Header” заголовка Fragment Header

4.5.3. Трансляция информационных сообщений ICMPv6

Из всех возможных информационных сообщений ICMPv6 трансляции подлежат только сообщения Echo Request и Echo Reply. Они транслируются в ICMPv4 сообщения Echo и Echo Reply, соответственно. При трансляции также пересчитывается поле контрольной суммы. Данные пакета копируются без изменений.

4.5.4. Трансляция сообщений ICMPv6 об ошибках

Так же, как и в протоколе ICMPv4, любое ICMPv6 сообщение об ошибке содержит в себе пакет, вызвавший ошибку (точнее столько байт этого пакета, чтобы суммарная длина пакета сообщения об ошибке не превышала 1280 байт). Поэтому необходимо транслировать не только сам заголовок ICMPv6, но и пакет, вызвавший ошибку, т.е. производить рекурсивную трансляцию. Трансляция заголовка ICMPv6 происходит по следующей схеме:

Destination Unreachable (Type 1):

Выставить поле Type в 3, а поле Code транслировать следующим образом:

Code 0: выставить Code в 1 (host unreachable)

Code 1: выставить Code в 10 (communication with destination host administratively prohibited)

Code 2: выставить Code в 5 (source route failed)

Code 3: выставить Code в 1 (host unreachable)

Code 4: выставить Code в 3 (port unreachable)

Packet Too Big (Type 2):

Транслировать в сообщение ICMPv4 Destination Unreachable с Code 4. Поле MTU необходимо скорректировать на разницу между размером заголовка IPv4 и размером заголовка IPv6 с учетом наличия в пакете расширенных заголовков.

Time Exceeded (Type 3):

Выставить поле Type в 11, поле Code не изменять.

Parameter Problem (Type 4):

Если Code есть 1, то транслировать в сообщение Port unreachable (Type 3, Code 2). Иначе выставить поле Type в 12, а поле Code в 0. Поле Pointer должно указывать на поле, вызвавшее ошибку в исходном пакете.

Все остальные сообщения об ошибке необходимо молча сбрасывать.

4.6. Особенности работы транслятора

Протоколы IPv6 и IPv4 во многом похожи, но между ними, кроме чисто синтаксических различий, есть и семантические, которые усложняют задачу трансляции. Такими семантическими различиями являются:

- различие в минимально допустимом размере MTU
- наличие в протоколе IPv6 расширенных заголовков
- различие в механизме фрагментации
- различие в механизме вычисления контрольной суммы протокола ICMP

4.6.1. Различие в минимально допустимом размере MTU

Протокол IPv4 не может работать на линиях, у которых максимальный блок передачи данных меньше 576 октетов. Точнее говоря, он не может работать на линиях, у которых нижележащий уровень сетевого интерфейса не может

пересылать блоки данных размером большим либо равным 576 октетам. У протокола IPv6 минимальное значение MTU было повышено до 1280 октетов.

4.6.2. Наличие в протоколе IPv6 расширенных заголовков

В отличие от протокола IPv4, в протоколе IPv6 в основной заголовок входит только базовая информация о пакете. Вся дополнительная информация содержится в расширенных заголовках IPv6. Таких заголовков существует несколько, и, кроме того, могут появляться новые заголовки. Транслятор использует только один такой расширенный заголовок: Fragment Header – заголовок фрагментации. Все остальные расширенные заголовки, имеющиеся в пакете, транслятор обязан игнорировать (те, которые он понимает). Проект стандарта обязывает каждую реализацию знать (и игнорировать) как минимум следующие заголовки:

- Hop by hop options – опции, обрабатываемые каждым промежуточным маршрутизатором по пути движения пакета
- Destination options – опции, обрабатываемые только получателем пакета
- Routing Header – заголовок маршрутизации

4.6.3. Различие в механизме фрагментации

Как уже отмечалось, одной из функций протоколов сетевого уровня является фрагментация слишком больших дейтаграмм перед посылкой к следующему узлу. Протоколы IPv6 и IPv4 осуществляют эту функцию по-разному.

Напомним, что для вычисления значения PMTU в протоколе IPv4 существует соответствующий механизм - Path MTU discovery, однако его применение не является обязательным для протокола IPv4. В протоколе IPv6 было решено избавиться от возможности фрагментации дейтаграмм промежуточными маршрутизаторами, поскольку процедура фрагментации является достаточно дорогой с точки зрения потребляемых ресурсов. Вместо этого, в IPv6 было решено сделать процедуру определения PMTU обязательной.

Таким образом, потенциально через транслятор могут взаимодействовать IPv6 хост, осуществляющий определение PMTU, и хост IPv4, не осуществляющий процедуру определения PMTU. При этом транслятор должен:

- поддерживать механизм определения PMTU со стороны IPv6
- осуществлять при необходимости фрагментацию входящих дейтаграмм со стороны IPv4

Поддержка механизма определения PMTU со стороны IPv6 осуществляется следующим образом. При приходе нефрагментированного (т.е. не содержащего заголовка Fragment Header) IPv6 пакета транслятор конвертирует

его в пакет IPv4, в заголовке которого выставляет флаг DF (don't fragment – не фрагментировать). Таким образом, если размер пакета был больше, чем PMTU в IPv4 части сети, то, дойдя до "узкого горла", пакет будет сброшен, и соответствующее уведомление (с новым значением PMTU) будет выслано посредством протокола ICMPv4. Транслятор, преобразовав это сообщение в ICMPv6 и переслав его получателю, завершит таким образом процедуру определения PMTU со стороны IPv6.

Примечание: если окажется, что значение PMTU, полученное в IPv4 части сети, меньше, чем минимальное значение MTU для протокола IPv6, то в этом случае протокол IPv6 обязует отправителя высылать пакеты с размером равным минимальному значению MTU протокола IPv6, но при этом добавлять к пакету заголовок Fragment Header. При приходе пакета IPv6, в котором есть заголовок Fragment Header, транслятор конвертирует его в пакет IPv4 со сброшенным флагом DF, позволяя, таким образом, промежуточным IPv4 маршрутизаторам фрагментировать этот пакет в случае необходимости.

При приходе пакета IPv4 с выставленным флагом DF (наличие у пакета выставленного флага DF означает, что отправитель этого пакета поддерживает механизм определения PMTU) транслятор конвертирует его в пакет IPv6, не содержащий заголовка фрагментации Fragment Header. Если размер этого пакета был больше, чем PMTU в IPv6 части сети, то, дойдя до "узкого горла", пакет будет сброшен маршрутизатором IPv6, и соответствующее уведомление будет выслано с новым значением PMTU. После конвертирования этого сообщения на трансляторе и доставки его отправителю, тот сможет вычислить это новое значение PMTU.

По отношению к фрагментации пакетов IPv4 транслятор может вести себя следующим образом:

- фрагментировать (если это необходимо) пакет до размера PMTU, т.е. поддерживать механизм определения PMTU протокола IPv6
- фрагментировать (если это необходимо) пакет до размера минимального значения PMTU протокола IPv6

Необходимость во фрагментации может возникнуть, если входящий пакет IPv4 содержит сброшенный флаг DF, либо если пакет уже является фрагментом. В обоих случаях (даже если фрагментация не производилась) транслятор обязан добавить к пакету заголовок фрагментации Fragment Header.

4.6.4. Различие в механизме вычисления контрольной суммы протокола ICMP

Как протокол ICMPv4, так и протокол ICMPv6 содержит в своем заголовке поле контрольной суммы пакета. Однако механизм вычисления этого поля для

этих протоколов различается: протокол ICMPv6 при вычислении использует псевдозаголовок IPv6, в то время как ICMPv4 не использует псевдозаголовка IPv4, поэтому при трансляции пакета ICMP любой версии необходимо заново пересчитывать поле контрольной суммы.

5. Реализация

Предлагаемая реализация механизма бесконтекстной IP/ICMP трансляции выполнена в виде загружаемого модуля ядра операционной системы Linux. Этот модуль в момент загрузки регистрирует себя в ядре в качестве драйвера дополнительного (псевдо)сетевого интерфейса с именем "siit". Для функционирования схемы трансляции в системе должен существовать еще, по крайней мере, один сетевой интерфейс. Таким образом, после загрузки модуля трансляции в системе будет существовать как минимум два сетевых интерфейса.

На представленном ниже рисунке показан пример подключения транслятора:

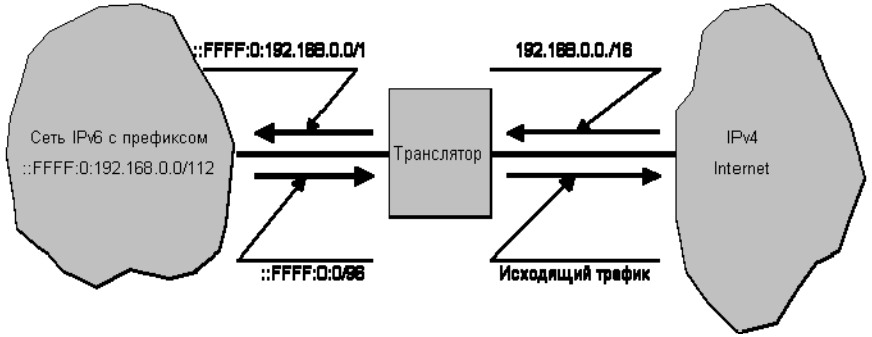

Рис. 9. Место транслятора в топологии сети

В этом примере локальная IPv6 подсеть подключена к Internet через транслятор. Этой подсети выделен диапазон IPv4 адресов 192.168.0.0/16, т.е. сеть класса B. При этом в IPv6 подсети хосты имеют адреса вида ::FFFF:0:192.168.x.x, где x.x - конкретный номер хоста. Хосты IPv6 адресуют хосты в IPv4 Internet с помощью IPv4-отображенных IPv6 адресов, т.е. адресов вида ::FFFF:x.x.x.x, где x.x.x.x - IPv4 адрес. Хосты в IPv4 Internet адресуют IPv6 хосты в данной IPv6 подсети с помощью IPv4 адресов 192.168.x.x.

Для того, чтобы включить механизм трансляции, необходимо включить в системе пересылку пакетов между интерфейсами системы (forwarding), т.е сделать систему маршрутизатором. Теперь, после того как трансляция включена, для того чтобы она заработала, необходимо просто подать на систему трафик, подлежащий трансляции. Для этого необходимо установить

машрутизацию IPv6 пакетов, идущих по адресам, начинающимся с префикса ::FFFF:0:0/96 на интерфейс siit (см. рис.10). Кроме того, необходимо установить в системе машрутизацию IPv4 пакетов из диапазона, предназначенного для IPv6 хостов, на интерфейс siit. Тогда приходящие в систему IPv6 пакеты с IPv4-отображенным адресом назначения будут проходить через IPv6 стек и попадать из него на интерфейс siit, на котором они будут претерпевать трансляцию в формат IPv4. После трансляции пакеты будут попадать из интерфейса siit в стек протокола IPv4, и, поскольку включена маршрутизация, пересылаться этим стеком на реальный исходящий интерфейс и уходить в сеть. Аналогичные события будут происходить и с IPv4 пакетами, приходящими на один из IPv4 адресов, выделенных для IPv6 хостов.

Реализация транслятора в виде драйвера (псевдо)сетевого интерфейса позволяет существенно упростить сам транслятор. Так, например, IP/ICMP транслятор, так же как и обычный маршрутизатор, должен отслеживать значение полей TTL и Hop Limit протоколов IPv4 и IPv6, соответственно. Входящие пакеты, у которых эти поля нулевые, транслятор должен сбрасывать и отсылать отправителю ICMP сообщение Time Exceeded. При реализации транслятора в виде драйвера сетевого интерфейса задача проверки этих полей ложится на сам стек IPv4 или IPv6. Точно так же, все необходимые проверки, которые должен выполнять маршрутизатор, а именно: проверка контрольных сумм заголовка IPv4, проверка соответствия размера пакета информации о размере, содержащейся в заголовке IP, проверка превышения длины пакета размера MTU на исходящем сетевом интерфейсе и т.д. - осуществляются самими стеками IPv4 и IPv6, а не транслятором.

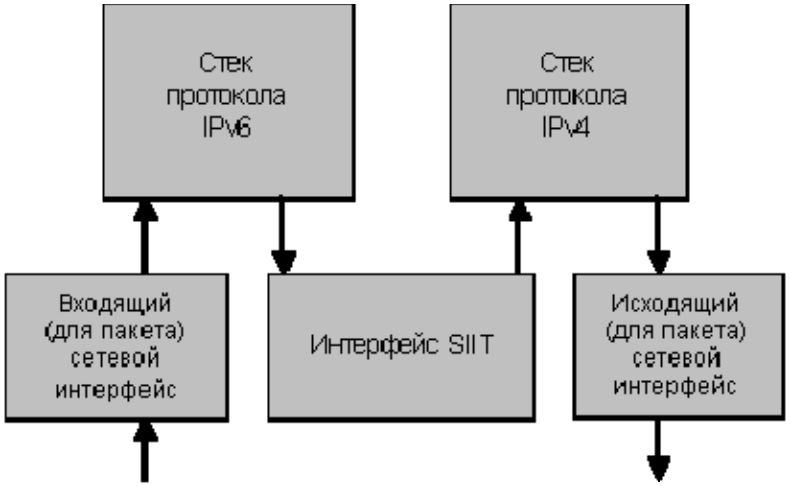

Рис. 10. Путь прохождения пакетов, подлежащих трансляции

6. Заключение

В результате данной работы был реализован один из механизмов обеспечения совместимости протоколов IPv6 и IPv4 – бесконтекстный IP/ICMP транслятор, способный, вместе с другими механизмами, обеспечить плавный и безболезненный переход на протокол IPv6. В процессе работы были изучены особенности и тонкости реализации данного транслирующего механизма в среде операционной системы Linux.

Литература

1. E. Nordmark, ***"Stateless IP/ICMP Translator (SIIT)"***, draft-ietf-ngtrans-siit-04.txt, December 1998.
2. S. Deering, R. Hinden, Editors, ***"Internet Protocol, Version 6 (IPv6) Specification"***, RFC 2460, December 1998.
3. S. Deering, R. Hinden, Editors, ***"IP Version 6 Addressing Architecture"*,** RFC 2373, July 1998.
4. R. Gilligan, E. Nordmark, ***"Transition Mechanisms for IPv6 Hosts and Routers"***, RFC 1933, April 1996.
5. S. Deering, A. Conta**, *"Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)"***, RFC 2463, December 1998.
6. J. Postel, ***"Internet Protocol"***, RFC 791, September 1981.
7. J. Postel, ***"Internet Control Message Protocol"***, RFC 792, September 1981.
8. J. Mogul, S. Deering, ***"Path MTU Discovery"***, RFC 1191, November 1990.
9. J. McCann, S. Deering, J. Mogul, ***"Path MTU Discovery for IP version 6"***, RFC 1981, August 1996.
10. K. Nichols, S. Blake, F. Baker, and D. L. Black***, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers"***, RFC 2474, December 1998.
11. M. Fiuczynski, V. Lam, B. Bershad**, *"The Design and Implementation of an IPv6/IPv4 Network Address and Protocol Translator"***, technical report of Department of Computer Science and Engineering, University of Washington
12. P. Srisuresh, K. Egevang, ***"The IP Network Address Translator (NAT)"***, RFC 1631, May 1994