Межпротокольный шлюз NAT-PT с функциями DNS-ALG и FTP-ALG для обеспечения взаимодействия между сетями IPv4 и IPv6

Г.В. Ключников, Д.С. Мишин, Д.В. Москалев, А.В. Никешин, В.З. Шнитман

Аннотация. При переводе Internet на протокол нового поколения IPv6 возникает целый ряд проблем, связанных как с обеспечением взаимодействия между двумя или большим числом островков IPv6, изолированных в мире IPv4, так и с установлением связи (или некоторого вида связи) между существующим миром IPv4 и нарождающимся миром IPv6. В статье рассматриваются вопросы реализации шлюзов прикладного уровня DNS-ALG и FTP-ALG для межпротокольного шлюза NAT-PT, соответствующих проекту стандарта IETF RFC 2766 “Network Address Translation - Protocol Translation”, а также особенности применения этих средств для обеспечения плавного и безболезненного перехода на протокол IPv6.

1. Введение

IPv6 [1, 2] представляет собой новую версию IP-протокола, созданную с целью замены протокола IPv4 [3], который был разработан в конце 70-х годов. IPv6 по сравнению с IPv4 имеет ряд преимуществ, которые будут способствовать будущему росту Internet и упростят конфигурирование и администрирование IP-сетей. IPv6 имеет большее адресное пространство, чем IPv4, модель адресации, которая способствует агрегированию маршрутов, и предлагает мощный механизм автоконфигурирования. Ожидается, что со временем рост Internet и потребность в решениях, обеспечивающих возможность немедленного применения (plug-and-play), приведут к широкой адаптации протокола IPv6.

При переводе Internet на протокол нового поколения IPv6 возникает целый ряд проблем, связанных как с обеспечением взаимодействия между двумя или большим числом островков IPv6, изолированных в мире IPv4, так и с установлением связи (или некоторого вида связи) между существующим миром IPv4 и нарождающимся миром IPv6. Как правило, решения первого круга проблем основываются на реализации маршрутизаторов с двойным стеком и организации туннелей IPv4 для передачи трафика IPv6. Для решения второго круга проблем также был предложен ряд механизмов, в основе

которых лежат технологии двойного стека, бесконтекстной и контекстной трансляции протоколов, а также организации туннелей IPv6 для передачи трафика IPv4.

Данная работа связана с реализацией механизма контекстной трансляции протоколов, поэтому, прежде всего, следует определить, что представляют собой IP/ICMP-трансляторы.

Механизм **бесконтекстного IP/ICMP-транслятора** (SIIT) [4] 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 которого, они подвергаются преобразованию в формат протокола IPv6 и пересылаются далее к своим получателям. Ответные пакеты, идущие от IPv6-хостов к IPv4-хостам (это индуцируется специальным типом IPv6-адреса назначения), также должны пройти через IP/ICMP-транслятор, но не обязательно 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 реализацию и использование. К сожалению, спецификация SIIT [4] предполагает, что узлам V6 для организации связи с узлами V4 присваивается V4-адрес (точнее, IPv4-транслированный адрес), но не описывает механизм присваивания этих адресов.

Механизм **контекстной IP/ICMP трансляции (NAT-PT)** [5] является логическим продолжением предыдущего. Для динамического присваивания адресов V6 узлам NAT-PT использует пул V4-адресов, когда через границы V4-V6 инициируются сеансы связи. Предполагается, что V4-адреса являются глобально уникальными. NAT-PT для обеспечения прозрачной маршрутизации дейтаграмм, пересекающих области различной адресации, связывает адреса в сети V6 с адресами в сети V4 и наоборот. Этот механизм не требует проведения каких-либо изменений в оконечных узлах, и маршрутизация IP-пакетов для оконечных узлов оказывается совершенно прозрачной. Однако он требует, чтобы в NAT-PT отслеживались поддерживаемые сеансы связи, и предполагает, что входящие и исходящие дейтаграммы, относящиеся к некоторому сеансу, проходят через один и тот же маршрутизатор с установленным NAT-PT.

Следует отметить, что некоторые приложения в своих данных передают сетевые адреса. NAT-PT ничего не знает о приложениях и не просматривает данные прикладного уровня. Поэтому для обеспечения работы таких приложений через NAT-PT необходимо использовать шлюзы прикладного уровня ALG (Application Level Gateway) [3]. ALG представляет собой специфического для приложения агента, позволяющего узлу IPv6 взаимодействовать с узлом IPv4 и наоборот. Объединение механизма протокольной трансляции SIIT с возможностями динамической трансляции

адресов NAT и соответствующими ALG предоставляет собой полное решение, которое позволит огромному числу широко используемых приложений взаимодействовать между узлами, работающими только на протоколе IPv6, и узлами, работающими только на протоколе IPv4, не требуя внесения никаких изменений в эти приложения. Основное предположение при применении NAT-PT заключается в том, чтобы он использовался, только если не возможны никакие иные средства взаимодействия между узлами – собственно IPv6 или IPv6 через туннели IPv4. Другими словами, цель данного механизма заключается в том, чтобы использовать трансляцию лишь между узлами, работающими только на протоколе IPv6, и узлами, работающими только на протоколе IPv4, в то время как трансляцию между узлами, работающими только на протоколе IPv6, и IPv4-частью узлов с двойным стеком, необходимо реализовывать с помощью других альтернативных механизмов.

Данная работа посвящена, главным образом, реализации шлюзов прикладного уровня службы доменных имен DNS-ALG и службы передачи файлов FTP-ALG, которые являются неотъемлемой частью полного механизма контекстной межпротокольной IP/ICMP-трансляции. Реализация SIIT и базовых функций NAT-PT была выполнена нами ранее при поддержке грантов РФФИ. Ниже приведено детальное описание работы транслятора, за которым следует описание его реализации и применявшихся подходов к тестированию разработанных программных средств.

2. Механизм трансляции протоколов и адресов с сохранением состояния соединений (NAT-PT)

2.1. Разновидности NAT-PT

Существует несколько вариантов построения механизма трансляции протоколов и адресов с сохранением состояния соединений. Эти варианты подробно описаны в спецификации RFC 2663 “IP Network Address Translator (NAT) Terminology and Considerations” [6].

Однонаправленный NAT-PT

В однонаправленном NAT-PT сеансы связи являются однонаправленными, исходящими из сети IPv6. Он отличается от двунаправленного NAT-PT, который позволяет инициировать сеансы связи в обоих направлениях, исходящем и входящем.

В свою очередь, однонаправленный NAT-PT также имеет две разновидности, а именно:

- основной NAT-PT (Network Address Translation – Protocol Translation);
- NAPT-PT (Network Address / Port Translation – Protocol Translation).

В основном NAT-PT резервируется блок адресов IPv4, которые используются для трансляции адресов IPv6-хостов, когда они порождают сеансы связи с IPv4-хостами во внешнем домене. Для пакетов, исходящих из домена IPv6, транслируются IP-адрес источника и связанные с ним поля, например, контрольные суммы заголовков IP, TCP, UDP и ICMP. Для входящих пакетов транслируются IP-адрес места назначения и перечисленные выше контрольные суммы.

Механизм NAPT-PT распространяет понятие трансляции на один шаг дальше, транслируя также и транспортные идентификаторы (например, номера портов TCP и UDP, идентификаторы запросов ICMP). Это позволяет мультиплексировать транспортные идентификаторы некоторого числа IPv6-хостов в транс-портные идентификаторы единственного IPv4-адреса. Таким образом, NAPT-PT позволяет множеству IPv6-хостов разделять один IPv4-адрес. Заметим, что NAPT-PT может быть объединен с основным NAT-PT так, что вместе с трансляцией портов будет использоваться целый пул внешних адресов.

Двунаправленный NAT-PT

При использовании двунаправленного NAT-PT сеансы связи могут порождаться узлами как из IPv4-сети, так и из IPv6-сети. Адреса IPv6-сети связываются с IPv4-адресами статически или динамически в тот момент времени, когда в любом из направлений устанавливаются соединения. Предполагается, что пространство имен между хостами в сетях IPv4 и IPv6 (т.е. их полностью квалифицированные доменные имена) является насквозь уникальным. Хосты в области IPv4 обращаются к хостам в области IPv6, используя для разрешения адресов службу доменных имен DNS. Для отображения имен в адреса совместно с двунаправленным NAT-PT должен применяться шлюз прикладного уровня службы доменных имен DNS-ALG (DNS Application Level Gateway) [7]. В частности, DNS-ALG должен транслировать IPv6-адрес в запросах и ответах DNS в соответствующий ему IPv4-адрес и наоборот, когда пакеты DNS пересекают адресные области IPv6-IPv4.

Для адресации IPv6-узлов используются реальные IPv6-адреса, схему назначения которых NAT-PT не определяет. Узлы IPv6 узнают адреса узлов IPv4, выполняя запросы к серверам DNS в домене IPv4 или к серверу DNS, внутреннему для IPv6-сети. Узел IPv6, которому необходимо взаимодействовать с узлом IPv4, должен использовать специальный префикс (PREFIX::/96) перед IPv4-адресом узла IPv4, который задается конфигурацией NAT-PT. Описанный выше способ позволяет использовать этот префикс без какого-либо конфигурирования узлов.

NAT-PT ничего не знает о приложениях и не просматривает данные прикладного уровня. Поэтому, как уже отмечалось, для обеспечения работы через NAT-PT приложений, которые в своих данных передают сетевые адреса, необходимо использование шлюзов прикладного уровня ALG (Application Level Gateway). По существу, шлюз прикладного уровня является специфическим для конкретного приложения агентом, который позволяет узлу IPv6 взаимодействовать с узлом IPv4 и наоборот. Спецификация NAT-PT [5] в

общих чертах описывает работу только шлюза прикладного уровня службы доменных имен – DNS-ALG и шлюза прикладного уровня службы передачи файлов – FTP-ALG.

2.2. Принципы функционирования NAT-PT

Работа основного NAT-PT

Работа основного NAT-PT иллюстрируется примером, взятым из RFC 2766 [5], и последующим описанием этого примера (рис. 1).

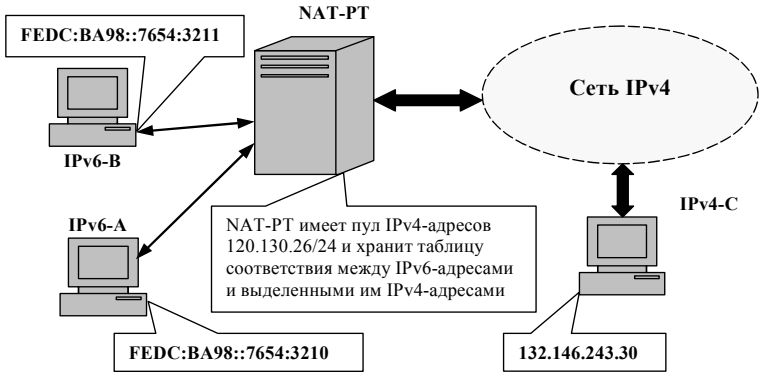

Рис. 1. Работа основного NAT-PT

NAT-PT имеет пул адресов, включающий IPv4-подсеть 120.130.26/24. IPv4-адреса из этого пула могут быть присвоены IPv6-адресам оконечных узлов IPv6 статически с сохранением взаимно однозначного соответствия. В этом случае потребуется столько IPv4-адресов, сколько имеется оконечных узлов IPv6. Однако мы предполагаем, что количество IPv4-адресов в пуле меньше, чем число оконечных узлов IPv6. Поэтому по крайней мере для некоторых из этих узлов требуется динамическое распределение адресов.

Пусть, например, узел IPv6-A хочет взаимодействовать с узлом IPv4-C. Узел IPv6-A создает и посылает пакет со следующими полями:

Source Address, SA = FEDC:BA98::7654:3210
Destination Address, DA = PREFIX::132.146.243.30

Примечание: Префикс PREFIX::/96 выбирается произвольно и задается в конфигурации NAT-PT. Пакеты, адресованные по этому префиксу, будут маршрутизироваться на NAT-PT. Требуется только, чтобы заранее конфигурируемый PREFIX был бы маршрутизируемым в рамках домена IPv6, т.е. это может быть любой маршрутизируемый префикс, который выберет администратор сети.

Этот IPv6-пакет маршрутизируется через шлюз NAT-PT, на котором он транслируется в формат IPv4.

Если исходящий пакет не является пакетом инициализации сеанса связи, NAT-PT должен уже хранить некоторое состояние о соответствующем сеансе, включающее присвоенный IPv4-адрес и другие параметры для трансляции. Если такого состояния не существует, пакет должен молча отбрасываться.

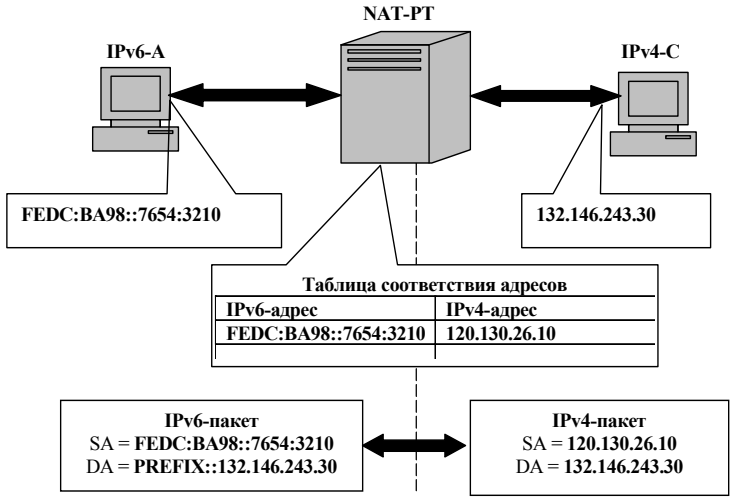

Рис. 2. Трансляция адресов в NAT-PT

Если пакет является пакетом инициализации сеанса связи, NAT-PT локально распределяет адрес из своего пула адресов (например, назначает адрес 120.130.26.10) и транслирует этот пакет в IPv4 (рис.2). Параметры трансляции кэшируются на время сеанса связи, и отображение IPv6 на IPv4 запоминается в NAT-PT. Результирующий IPv4-пакет имеет SA=120.130.26.10 и DA=132.146.243.30.

Любой возвращаемый пакет будет распознаваться NAT-PT как принадлежащий тому же самому сеансу связи. Для этого NAT-PT будет использовать сохраненную информацию о состоянии соединения и соответствующим образом транслировать такой пакет. При этом результирующие адреса будут равны SA=PREFIX::132.146.243.30, DA=FEDC:BA98::7654:3210. Заметим, что пакет с такими адресами может теперь маршрутизироваться как обычный пакет внутри оконечной сети, работающей только по протоколу IPv6.

Работа NAPT-PT

Как уже отмечалось, механизм NAPT-PT (Network Address Port Translation + Protocol Translation) позволяет IPv6-узлам взаимодействовать с IPv4-узлами, используя всего один IPv4-адрес. При этом порты TCP/UDP узлов IPv6 транслируются в порты TCP/UDP этого глобального IPv4-адреса.

Смысл введения механизма NAPT-PT заключается в следующем. Если в обычном NAT-PT исчерпается пул IPv4-адресов, выделенных для целей трансляции, то ни один новый IPv6-узел больше не сможет открывать сеансы связи с внешним миром через NAT-PT. С другой стороны, NAPT-PT позволит открыть максимально до 63K сеансов TCP и до 63K сеансов UDP.

Модифицируем рассмотренный выше пример, предполагая, что на пограничном маршрутизаторе вместо NAT-PT мы имеем NAPT-PT, и все IPv6-адреса будут отображаться на единственный IPv4-адрес 120.130.26.10 (рис. 3).

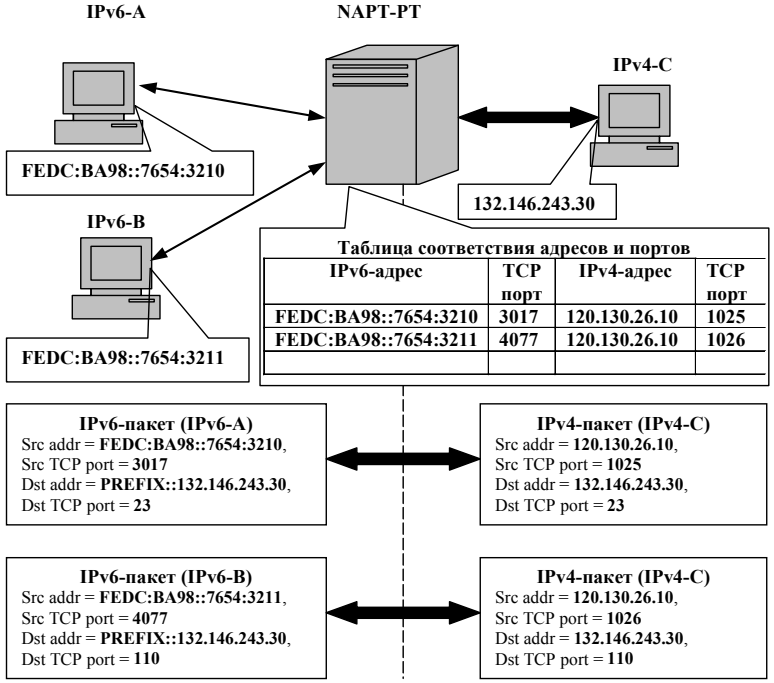

Рис. 3. Работа основного NAPT-PT.

Предположим, что узел IPv6-A создает сеанс TCP с узлом IPv4-C. Узел IPv6-A формирует и посылает пакет со следующими параметрами:

Source Address, SA=FEDC:BA98::7654:3210 , source TCP port = 3017
Destination Address, DA = PREFIX::132.146.243.30, destination TCP port = 23

Когда этот пакет поступит в NAPT-PT, последний присвоит один из TCP-портов выделенного IPv4-адреса для трансляции кортежа (Source Address, Source TCP port) следующим образом:

SA=120.130.26.10, source TCP port = 1025

DA=132.146.243.30, destination TCP port = 23

Возвращаемый трафик от узла с адресом 132.146.243.30 с TCP-порта 23, посылаемый на адрес 120.130.26.10, порт 1025, будет распознан как принадлежащий тому же самому сеансу и будет транслироваться обратно в IPv6 следующим образом:

SA = PREFIX::132.146.243.30, source TCP port = 23
DA = FEDC:BA98::7654:3210, destination TCP port = 3017

Теперь предположим, что узел IPv6-B устанавливает TCP-соединение с узлом IPv4-C на порт 110. Узел IPv6-B формирует и посылает пакет со следующими параметрами:

Source Address, SA=FEDC:BA98::7654:3211 , source TCP port = 4077
Destination Address, DA = PREFIX::132.146.243.30, destination TCP port = 110

Когда этот пакет поступит в NAPT-PT, последний присвоит следующий свободный порт из списка TCP-портов выделенного IPv4-адреса и оттранслирует его в пакет IPv4 со следующими параметрами:

SA=120.130.26.10, source TCP port = 1026
DA=132.146.243.30, destination TCP port = 110

Возвращаемый трафик от узла с адресом 132.146.243.30 с TCP-порта 110, посылаемый на адрес 120.130.26.10 и порт 1026, будет распознан как принадлежащий сеансу связи с узлом IPv6-B и будет транслироваться обратно в IPv6 следующим образом:

SA = PREFIX::132.146.243.30, source TCP port = 110
DA = FEDC:BA98::7654:3211, destination TCP port = 4077

Заметим, что входящие сеансы связи при использовании механизма NAPT-PT ограничиваются одним сервером на сервис, назначаемым с помощью статического отображения TCP/UDP портов. Например, в домене IPv6 только узел IPv6-A из нашего примера может быть HTTP-сервером (port 80). Пусть узел IPv4-C посылает пакет:

SA=132.146.243.30, source TCP port = 1025
DA=120.130.26.10, destination TCP port = 80

NAPT-PT оттранслирует этот пакет следующим образом:

SA=PREFIX::132.146.243.30, source TCP port = 1025
DA=FEDC:BA98::7654:3210, destination TCP port = 80

Таким образом, в приведенном выше примере все сеансы связи, которые достигают NAPT-PT с портом места назначения равным 80, будут переадресованы на один и тот же узел IPv6-A.

Трансляция протоколов

Трансляция заголовков IPv4 в заголовки IPv6 и обратно выполняется точно так же, как и в SIIT (RFC 2765) [4], но имеются некоторые изменения, которые

требуется внести в SIIT, поскольку NAT-PT выполняет также и трансляцию сетевых адресов. Эти изменения незначительны и детально описаны в спецификации NAT-PT RFC 2766 [5].

3. Шлюз прикладного уровня DNS-ALG

Отображения имен на IPv4-адреса хранятся в DNS в виде записей о ресурсах (RR – resource record) типа "A". Отображения имен на IPv6-адреса хранятся в DNS в виде записей о ресурсах типа "AAAA" [7, 8]. Использование записи типа "A6" для отображения имен на IPv6-адреса мы не рассматриваем. Отображения адресов в имена хранятся в виде записей "PTR": для IPv4-адресов в зоне "IN-ADDR.ARPA", для IPv6 адресов – в зоне "IP6.INT". Соответственно, и запросы могут быть типа "A", "AAAA" и "PTR". Только эти типы запросов и записи о ресурсах в ответах подлежат трансляции.

Структурная схема DNS при использовании DNS-сервера внутри сети IPv6 показана на рис. 4. DNS-сервер IPv4 с именем DNS-IPv4 является полномочным сервером для какой-либо зоны, например, зоны "ispras.ru", и его имя – DNS-IPv4.ispras.ru. Сервер DNS-IPv4 делегирует полномочия для подзоны "ipv6.ispras.ru" серверу DNS-IPv6 с именем DNS-IPv6.ipv6.ispras.ru. Для передачи полномочий сервер DNS-IPv4 должен иметь так называемую «связующую запись» (glue record), т.е. запись о ресурсе типа "A", содержащую IPv4-адрес сервера DNS-IPv6.ipv6.ispras.ru (хотя это уже не его зона).

Например:

```
ipv6.ispras.ru.          IN NS   DNS-IPv6.ipv6.ispras.ru.
DNS-IPv6.ipv6.ispras.ru. IN A    195.208.53.20
```

Поэтому для внутреннего DNS-сервера сети IPv6 необходимо выделить фиксированный IPv4-адрес из пула IPv4-адресов NAT-PT и статически отобразить его на IPv6-адрес сервера DNS-IPv6. Таким образом, NAT-PT должен хранить взаимно однозначное соответствие между этим IPv4-адресом и IPv6-адресом внутреннего DNS-IPv6 сервера, и такая запись должна присутствовать в статических записях.

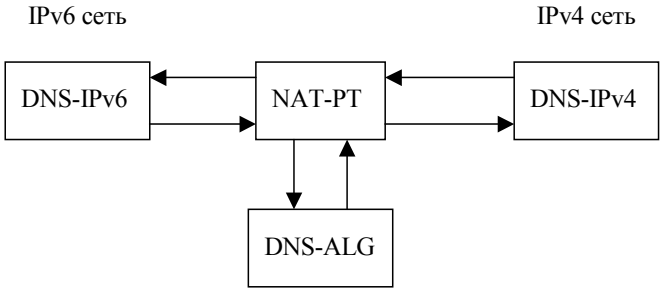

Рис. 4. Структурная схема DNS при использовании NAT-PT.

IPv6-адрес DNS-IPv4 сервера формируется из его IPv4-адреса и префикса (PREFIX::/96). IPv6-узлы должны использовать этот адрес при обращении к DNS-IPv4 серверу из сети IPv6.

Значения времени жизни (TTL) для всех DNS-записей о ресурсах, передающиеся через NAT-PT, должны быть установлены в 0 так, чтобы DNS-серверы/клиенты не кэшировали временно присвоенные RR. Однако спецификация допускает и значение 1, которое в некоторых реализациях DNS-клиентов, содержащих ошибки, работает лучше. В нашей реализации мы используем значение 1. Для статически отображаемых адресов значения TTL должны оставаться неизменными.

В общем случае пакеты DNS переносятся с помощью протоколов TCP или UDP [9, 10]. Сервер DNS слушает запросы на порте 53. Мы рассматриваем работу DNS только по протоколу UDP, поскольку это упрощает реализацию; кроме того, в настоящее время для переноса запросов DNS транспорт TCP практически не используется, а передача зон DNS через границы IPv4/IPv6 не рекомендуется. Таким образом, все пакеты UDP, у которых значение порта источника или места назначения равно 53, рассматриваются как пакеты DNS и должны передаваться для обработки в DNS-ALG.

Ниже представлена подробная спецификация функционирования шлюза DNS ALG.

3.1. Трансляция запросов со стороны IPv4 и ответов на них

Запросы на разрешение имен домена IPv6 от IPv4-узлов направляются на сервер DNS-IPv6. Они могут иметь тип "A", "AAAA" и "PTR", но запросы типа "AAAA" не транслируются и передаются без изменения.

DNS-ALG будет модифицировать DNS-запросы следующим образом:

a) Для запросов "по имени – адрес" типа "A":
   - Изменяет поле QTYPE (тип запроса) с "A" (1) на "AAAA" (28).

b) Для запросов "по адресу – имя" типа "PTR":
   - Если существует однозначное отображение для IPv4-адреса, указанного в запросе, заменяет в поле QNAME строку "IN-ADDR.ARPA" строкой "IP6.ARPA". Кроме того, заменяет октеты адреса IPv4, следующие в обратном порядке и предшествующие строке "IN-ADDR.ARPA", соответствующими октетами адреса IPv6, следующими в обратном порядке. Это означает, что данный адрес выделен конкретному узлу, например, какому-либо серверу, и никакой другой IPv6-адрес для него не будет использоваться (это справедливо для адресов, участвующих только в адресной трансляции). При этом изменяется размер пакета, и модуль NAT-PT должен пересчитать контрольную сумму псевдозаголовка UDP, а также скорректировать IP-заголовок (поле длины пакета).
   - Если отображения не существует, отбрасывает пакет.

После трансляции запроса в IPv6-пакет последний направляется серверу DNS-IPv6. DNS-IPv6 обрабатывает запрос и возвращает ответ, который снова попадает в DNS-ALG и обрабатывается следующим образом:

a) Для ответов на запрос "по имени – адрес" типа "AAAA" в соответствующей записи о ресурсе:
   - Изменяет тип записи в поле TYPE с "AAAA" на "A".
   - Если существует отображение данного IPv6-адреса на IPv4-адрес, заменяет IPv6-адрес в поле RDATA (разрешенный сервером DNS-IPv6) на IPv4-адрес.
   - Если отображения не существует, то именно в этот момент DNS-ALG выделяет для данного IPv6-адреса один из свободных адресов из пула IPv4-адресов NAT-PT и делает соответствующую DNS-запись в своей внутренней таблице. Далее он заменяет IPv6-адрес в поле RDATA на вновь выделенный IPv4-адрес. Если же свободного IPv4-адреса в пуле адресов NAT-PT не существует, отбрасывает такой пакет. В этом случае клиент будет повторно передавать запрос на разрешение имени и, возможно, за это время в пуле адресов появится свободный IPv4-адрес. Кроме того, в этом случае клиентское приложение получит сообщение об ошибке DNS-сервиса. Оно может отложить установление соединения на некоторое время и позже снова попытаться его установить.
   - Корректирует поле RDLENGTH, сокращая его значение на 96 бит.
   - Устанавливает поле TTL в 1.
   - В модуле NAT-PT необходим пересчет контрольных сумм и изменение размера пакета в заголовке IP.

b) Для ответов на запрос "по адресу – имя" типа "PTR" в соответствующей записи о ресурсе:
   - Если ответ присутствует и существует отображение IPv6-адреса, указанного в секции ответов, на IPv4-адрес, то заменяет в поле RDATA строку "IP6.INT" строкой "IN-ADDR.ARPA". А также заменяет октеты адреса IPv6, следующие в обратном порядке и предшествующие строке "IP6.INT", на соответствующие октеты адреса IPv4, следующие в обратном порядке и предшествующие строке "IN-ADDR.ARPA". Если ответ отсутствует, переправляет такую запись без изменений. Если отображения не существует, отбрасывает такой пакет.
   - Корректирует поле RDLENGTH.
   - Устанавливает поле TTL в 1.
   - В модуле NAT-PT необходим пересчет контрольных сумм и изменение размера пакета в заголовке IP.

Пример

Предположим, сервер DNS-IPv6 имеет следующие записи:

```
node-a.ipv6.ispras.ru.      IN AAAA     FEDC:BA98::7654:3210
DNS-IPv6.ipv6.ispras.ru.    IN AAAA     FEDC:BA98::1110:2220
```


В DNS-ALG имеются следующие отображения адресов на IPv4-адреса:

```
FEDC:BA98::1110:2220 <-> 195.208.53.20
FEDC:BA98::7654:3210 <-> 195.208.53.21
```

Со стороны IPv4 приходит запрос на разрешение имени:

```
SrcAddr = 132.146.243.30, source UDP port = 1025
DstAddr = 195.208.53.20, destination UDP port = 53
node-a.ipv6.ispras.ru, type = A, class = IN
```

DNS-ALG транслирует запрос:

```
SrcAddr = PREFIX::132.146.243.30, source UDP port = 1025
DstAddr = FEDC:BA98::1110:2220, destination UDP port = 53
node-a.ipv6.ispras.ru, type = AAAA, class = IN
```

и направляет его в сеть IPv6.

DNS-IPv6 сервер возвращает ответ:

```
DstAddr = PREFIX::132.146.243.30, destination UDP port = 1025
SrcAddr = FEDC:BA98::1110:2220, source UDP port = 53
node-a.ipv6.ispras.ru.  IN AAAA     FEDC:BA98::7654:3210
```

Этот ответ перехватывается и транслируется DNS-ALG в следующий вид:

```
DstAddr = 132.146.243.30, destination UDP port = 1025
SrcAddr = 195.208.53.20, source UDP port = 53
node-a.ipv6.ispras.ru.  IN A        195.208.53.21
```

и отправляется запрашивающему узлу.

3.2. Трансляция запросов со стороны IPv6 и ответов на них

В узлах IPv6 для получения IP-адреса удаленной стороны по имени используются запросы типа "A" и "AAAA", а для получения имени по IP-адресу – запросы типа "PTR". DNS-ALG обрабатывает их следующим образом:

a) Запрос "по имени – адрес" типа "A":
   - Пересылает такой запрос без изменений, но запоминает идентификатор DNS-сообщения (поле ID), чтобы при поступлении ответа иметь возможность определить, что это ответ на запрос типа "A", и не производить его трансляцию в ответ типа "AAAA".

b) Запрос "по имени – адрес" типа "AAAA":

Заметим, что для такого запроса возникает неоднозначность в реализации. С одной стороны, RFC 2766 определяет, что необходимо переслать запрос типа "AAAA" без изменения и дополнительно выслать запрос типа "A" для того же имени. Тогда, если в удаленном узле имеются оба типа записей (например, это двухстековый узел), то запрашивающий IPv6-узел получит два ответа типа "AAAA": с глобальным IPv6-адресом и IPv6-адресом вида PREFIX::X.X.X.X. В

этом случае в узле-источнике запроса возникает проблема выбора адреса места назначения, т.е. не очевидно, какой из адресов он выберет. С другой стороны, если вообще не посылать запись типа "AAAA", предполагая, что NAT-PT предназначен для работы только с IPv4-узлами, то исключается возможность взаимодействия с удаленным узлом, если он, например, двухстековый, используя другие средства типа туннелей. Но мы все же придерживаемся последнего способа (расширить функциональность, связанную с посылкой еще и запроса типа "AAAA", не составит проблем, если это будет необходимо). Поэтому DNS-ALG транслирует запрос следующим образом:

c) Изменяет поле QTYPE (тип запроса) с "AAAA" (28) на "A" (1). Других изменений не происходит, и размер пакета не изменяется.

d) Запрос "по адресу – имя " типа "PTR":
- Если в поле QNAME октеты адреса IPv6, следующие в обратном порядке и предшествующие строке "IP6.INT", содержат PREFIX::/96, то заменяет строку "IP6.INT" строкой "IN-ADDR.ARPA", а сами октеты адреса IPv6 – на соответствующие октеты адреса IPv4, следующие в обратном порядке и предшествующие строке "IN-ADDR.ARPA". При этом в модуле NAT-PT необходим пересчет контрольных сумм и изменение размера пакета в заголовке IP.
- Если в поле QNAME октеты адреса IPv6 PREFIX::/96 не содержат, запрос отправляется без изменений.

При приходе ответа от сервера DNS-IPv4:

a) В ответе на запрос "по имени – адрес" во всех записях о ресурсах типа "A":
- Проверяет по ранее сохраненному идентификатору запроса, не является ли он ответом на запрос типа "A".
- Если является, пересылает ответ без изменений.
- Иначе DNS-ALG выполняет трансляцию:
  - o Изменяет поле TYPE (тип записи) с "A" на "AAAA";
  - o Заменяет в поле RDATA IPv4-адрес на IPv6-адрес вида PREFIX::X.X.X.X, где X.X.X.X представляет данный IPv4-адрес;
  - o Корректирует поле RDLENGTH;
  - o Устанавливает поле TTL в 1.
- В модуле NAT-PT необходим пересчет контрольных сумм и изменение размера пакета в заголовке IP.

b) В ответе на запрос "по адресу – имя" во всех записях о ресурсах типа "PTR":
- Если поле NAME содержит подстроку "IN-ADDR.ARPA"
  - o Заменяет строку "IN-ADDR.ARPA" строкой "IP6.INT", а также октеты адреса IPv4, следующие в обратном порядке и предшествующие строке "IN-ADDR.ARPA", соответствующими октетами адреса IPv6 вида PREFIX::X.X.X.X, следующими в обратном порядке и предшествующими строке "IP6.INT";
  - o Корректирует поле RDLENGTH;
  - o Устанавливает поле TTL в 1.
- В модуле NAT-PT необходим пересчет контрольных сумм и изменение размера пакета в заголовке IP.
- Если подстрока "IN-ADDR.ARPA" не присутствует, пересылает сообщение без изменения.

4. Шлюз прикладного уровня FTP-ALG

Для обеспечения прозрачной работы между узлами сетей IPv4 и IPv6 на прикладном уровне по протоколу FTP межпротокольный шлюз NAT-PT необходимо расширить функциями FTP-ALG. Протокол FTP использует два TCP-соединения: управляющее соединение (control connection) для передачи FTP-команд и соединение для передачи данных (data connection). Командное соединение всегда инициируется клиентом на TCP-порт 21 сервера. Команды на установление соединения данных содержат информацию об IP-адресе и номере порта. RFC 959 "File Transfer Protocol" [11] определяет две такие команды: PORT и PASV. RFC 2428 "FTP Extensions for IPv6 and NATs" [12] фактически заменяет их на команды EPRT и EPSV. Соединение данных устанавливается всякий раз, когда необходимо передать данные, и закрывается по окончании передачи данных.

Все команды FTP передаются по управляющему соединению. Так как для корректной работы протокола FTP необходимо транслировать только управляющие команды PORT, PASV, EPRT, EPSV, шлюз прикладного уровня FTP-ALG должен обрабатывать только управляющие соединения FTP. Соединения данных остаются без изменения (т.е. FTP-ALG их не затрагивает). Управляющее соединение инициируется клиентом на TCP-порт 21 сервера. Поэтому все приходящие TCP-пакеты, у которых значение порта отправителя или получателя равно 21, направляются в модуль FTP-ALG. Модуль FTP-ALG работает с собственным списком записей типа FTP.

Различаются два типа сеансов FTP: порождаемые со стороны IPv4 и порождаемые со стороны IPv6.

4.1. Соединения, порождаемые клиентом IPv4

Клиент IPv4 может порождать команды PORT и PASV, и, если он реализует RFC 2428, команды EPRT и EPSV. При этом команды PORT, PASV транслируются в команды EPRT и EPSV, соответственно. Кроме того, ответ EPSV перед отправкой клиенту IPv4 транслируется в ответ PASV.

Трансляция команды PORT

Команда PORT h1,h2,h3,h4,p1,p2 транслируется в команду

EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>.

Поле <net-prt> устанавливается равным 2 (AF_INET6), IPv4-адрес h1,h2,h3,h4 транслируется в IPv6 <net-addr> в представлении ASCII, IPv4 TCP-порт p1,p2 в десятичное значение IPv6 порта <tcp-port> в представлении ASCII.

Примечание. Кроме того, необходимо транслировать и ответ EPRT в ответ PORT, так как символьная строка ответа может включать описание команды, которая выполнилась или не выполнилась успешно. Например,

Клиент: PORT 195,208,32,215,192,89
После трансляции: EPRT |2|1080::8:800:200C:417A|5282|

Ответ: 200 EPRT command successful.
После трансляции: 200 PORT command successful.

Трансляция команды PASV

Команда PASV транслируется в команду

EPSV<space><net-prt>
Поле <net-prt> устанавливается равным 2 (AF_INET6).

Ответ EPSV:
229 <text indicating server is entering extended passive mode> (<d><d><d><tcp-port><d>)

транслируется в ответ PASV:

227 Entering Passive Mode (h1,h2,h3,h4,p1,p2)

Так как в ответе EPSV не присутствует IPv6-адрес, то он получается из поля адреса отправителя IPv6-пакета и транслируется в IPv4-адрес h1,h2,h3,h4. Этот адрес может быть получен из соответствующего поля записи FTP-ALG. Поле IPv6-порта <tcp-port> из ответа EPSV транслируется в IPv4-порт p1,p2. Если используется NAPT, необходима предварительная трансляция порта в соответствующее ему IPv4-значение перед переводом в p1,p2.

Если клиент использует команды EPRT и EPSV, то FTP-ALG будет транслировать только параметры этих команд.

Трансляция команды EPRT

Поле <net-prt> транслируется из 1 (AF_INET) в 2 (AF_INET6), поле IPv4-адреса <net-addr> – в соответствующий ему IPv6-адрес <net-addr> в представлении ASCII, поле порта <tcp-port> не транслируется.

Трансляция команды EPSV

В команде EPSV транслируется поле <net-prt> из 1 (AF_INET) в 2 (AF_INET6).

В ответе EPSV поле <tcp-port> транслируется только в том случае, если используется порт-трансляция NAPT.

Примечание. При трансляции команд PASV и EPSV со стороны IPv4-клиентов необходимо запомнить, какая именно команда была транслирована, чтобы корректно произвести трансляцию ответа. Если производится трансляция

ответа от команд PORT и EPRT, то для них также необходимо знать, какая была команда запроса.

4.2. Соединения, порождаемые IPv6 клиентом

Возможны два подхода:

1) Транслировать v6-команды EPRT, EPSV в v4-команды EPRT, EPSV, и v4-ответы EPSV в v6-ответы EPSV;
2) Транслировать v6-команды EPRT и EPSV в команды PORT и PASV, и ответы PASV в ответы EPSV.

В первом случае транслируются параметры команд <net-prt>, <net-addr>, <tcp-port> из значений v6 в соответствующие им значения v4. При этом v4-сервер должен поддерживать команды EPRT и EPSV [12], иначе v4-FTP-сервер будет возвращать ошибку типа: “500 'EPSV': command not understood”, и v6-FTP-клиент не сможет работать. Возможным решением является следующее: не переправлять клиенту ответ об ошибке, а запомнить команду (и все необходимые данные), и в случае такой ошибки транслировать команду в PORT или PASV и далее действовать по второму варианту.

Во втором случае будет невозможно транслировать команду “EPSV<space>ALL”, по которой клиент извещает сервер, что в дальнейшей работе он будет использовать только команду EPSV; при этом сервер должен отбрасывать все другие команды данных PORT, PASV и EPRT. Эту ситуацию можно обойти, просто отвечая v6-клиенту на команду “EPSV<space>ALL” сообщением об успешном выполнении команды от имени сервера, но при этом реально не посылая серверу никакую команду. Этому варианту мы и следуем.

Трансляция команды EPRT

Команда EPRT<space><d><net-prt><d><net-addr><d><tcp-port><d>
транслируется в

команду PORT h1,h2,h3,h4,p1,p2.

IPv6-адрес <net-addr> в представлении ASCII транслируется в IPv4-адрес h1,h2,h3,h4, десятичное значение IPv6-порта <tcp-port> в представлении ASCII транслируется в IPv4-значение порта p1,p2.

Примечание. Кроме того, необходимо транслировать и ответ PORT в ответ EPRT, так как символьная стока ответа может включать название команды, которая выполнилась или не выполнилась успешно.

Трансляция команды EPSV

Команда EPSV<space><net-prt>
транслируется в PASV.

Ответ PASV: 227 Entering Passive Mode (h1,h2,h3,h4,p1,p2)
транслируется в ответ EPSV:

229 <text indicating server is entering extended passive mode> (<d><d><d><tcp-port><d>)

Поле IPv4-порта p1,p2 из ответа PASV транслируется в поле IPv6-порта <tcp-port>. Изменения значения порта не производится (нет NAPT).

Примечание. Во всех случаях нет необходимости знать, какая была предыдущая команда.

Примечание для всех случаев. Кроме того, необходимо обрабатывать ошибочные коды возврата. Так, например, в RFC 2428 [12] вводится код ошибки 522 и v4-клиент может его не понимать.

4.3. Коррекция заголовков пакетов управляющих соединений

Трансляция данных управляющих команд может привести к изменению размера пакета. Это приводит к изменению порядковых номеров TCP в управляющем соединении в обоих направлениях (sequence and acknowledgment numbers) и к изменению полей размеров пакета в заголовках IP (IPv4 Total Length и IPv6 Payload Length). Потому FTP-ALG должен хранить дельты порядковых номеров для обоих направлений управляющего соединения, чтобы корректировать порядковые номера. Величина дельты получается путем вычитания длины команды после трансляции от длины команды до трансляции и прибавляется к предыдущему сохраненному значению этой дельты. Заметим, что разница может быть отрицательной.

Обозначим дельту для исходящих соединений (из v6 в v4) как Δout, а для входящих (из v4 в v6) как Δin. Если пришедший пакет является пакетом из v6 в v4-область (исходящий) и в нем изменяется длина команды FTP, то разница между длиной v4 результирующей команды (после трансляции) и длиной оригинальной v6-команды (до трансляции) прибавляется к Δout. При этом если длина v4-команды будет меньше длины v6-команды, разница будет отрицательная.

Если пришедший пакет является пакетом из v4 в v6-область (входящий) и в нем изменяется длина команды FTP, то разница между длиной результирующей v6-команды и длиной v4-команды прибавляется к Δin. При этом, если длина v6-команды будет меньше длины v4-команды, разница будет отрицательная.

Коррекция дельты производится после трансляции пакета, так как порядковый номер пакета указывает на первый байт данных, а номер подтверждения – на следующий байт, который противоположная сторона ожидает принять.

Тогда при приходе пакета от v6-хоста (и это исходящий пакет управляющего соединения) порядковый номер (sequence number, SN) этого пакета увеличивается на Δout, а порядковый номер подтверждения (acknowledgment number, AN) этого же пакета уменьшается на Δin. При приходе пакета от v4-

хоста (и это входящий пакет управляющего соединения) SN этого пакета увеличивается на Δin, а AN этого же пакета уменьшается на Δout.

Поясним вышесказанное **примером**:

1) (OUT) IPv4	←	IPv6	Δout = 0, Δin = 0
L4 = 20	←	L6 = 50	Δ´out = -30, Δ´in = 0
SN´out = SNout + Δout = 100	←	SNout = 100	
AN´out = ANout - Δin = 500	←	ANout = 500	

Результирующие дельты: Δout += Δ´out = -30, Δin += Δ´in = 0

2) (IN) IPv4	→	IPv6	Δout = -30, Δin = 0
L4 = 40	→	L6 = 60	Δ´out = 0, Δ´in = 20
SNin = 500	→	SN´in = SNin + Δin = 500	
ANin = 120	→	AN´in = ANin - Δout = 150	

Результирующие дельты: Δout += Δ´out = -30, Δin += Δ´in = 20

3) (OUT) IPv4	←	IPv6	Δout = -30, Δin = 20
L4 = 40	←	L6 = 40	Δ´out = 0, Δ´in = 0
SN´out = SNout + Δout = 120	←	SNout = 150	
AN´out = ANout - Δin = 540	←	ANout = 560	

Результирующие дельты: Δout += Δ’out = -30, Δin += Δ’in = 20

4) (IN) IPv4	→	IPv6	Δout = -30, Δin = 20
L4 = 40	→	L6 = 60	Δ’out = 0, Δ’in = 20
SNin = 540	→	SN´in = SNin + Δin = 560	
ANin = 160	→	AN´in = ANin - Δout = 190	

Результирующие дельты: Δout += Δ´out = -30, Δin += Δ´in = 40

5) (OUT) IPv4	←	IPv6	Δout = -30, Δin = 40
L4 = 30	←	L6 = 40	Δ´out = -10, Δ´in = 0
SN´out = SNout + Δout = 160	←	SNout = 190	
AN´out = ANout - Δin = 580	←	ANout = 620	

Результирующие дельты: Δout += Δ´out = -40, Δin += Δ´in = 40

Описанный сценарий справедлив для случая, когда на каждый пакет, посланный с одной стороны, приходит ответный пакет с другой. Но когда с одной из сторон последовательно приходит несколько пакетов, а ответы на них приходят позже, происходит неправильная коррекция дельт, и последующие ответы будут транслированы неверно.

Для того чтобы избежать таких ситуаций, необходимо вести историю дельт для каждого пакета, сохраняя значения SN и AN и текущие значения дельт для них и выставлять тайм-аут для каждой записи. Тогда по приходу ответного пакета можно применять для преобразования нужные дельты. Сразу удалять такую запись нельзя, так как возможны повторные передачи пакетов с одинаковыми

SN и AN. Кроме того, возможны ситуации, когда приходит ACK, а затем с той же стороны приходит, например, PUSH, в котором присутствует ACK с тем же номером. Еще нужно учесть возможность прихода ACK не на каждый пакет, а на несколько пакетов PUSH сразу, с указанием последнего принятого байта.

По истечении тайм-аута запись будет удаляться сборщиком мусора.

Если приходит пакет TCP-сеанса, и для номеров SN и AN не существует записи с дельтами, это означает, что данный пакет – либо заблудившийся, либо ошибочный, либо посланный преднамеренно для атаки. Такой пакет можно либо отбросить, либо переправить без изменений на откуп принимающей стороны. Мы выбрали первый вариант.

5. Ограничения, присущие механизму NAT-PT

Использование механизмов трансляции адресов и протоколов связано с целым рядом ограничений. Все ограничения, присущие общему механизму трансляции сетевых адресов (NAT), в той же мере присущи и NAT-PT. Ниже подробно рассматриваются наиболее важные из этих ограничений, а также некоторые ограничения, присущие только механизму NAT-PT.

Ограничения топологии сети

Требуется, чтобы все запросы и ответы, относящиеся к одному сеансу связи, маршрутизировались через один и тот же маршрутизатор NAT-PT. Одним из способов обеспечения этого требования может быть организация NAT-PT на пограничном маршрутизаторе, который является уникальным для оконечного IPv6-домена, когда все IP-пакеты либо порождаются из этого домена, либо этот домен является их местом назначения. Это общая проблема, связанная с NAT, и она полностью описана в [6]. Заметим, что это ограничение не распространяется на пакеты, которые порождаются из узлов с двойным стеком или направляются на узлы с двойным стеком и не требуют трансляции адресов. Это справедливо, поскольку в устройстве с двойным стеком IPv4-адреса, заключенные в V6-адрес, могут быть идентифицированы по формату адреса PREFIX::x.y.z.w, и маршрутизатор с двойным стеком может соответствующим образом маршрутизировать пакет между V4 и узлами с двойным стеком без отслеживания информации о состоянии.

Это ограничение также не должно влиять на взаимодействие между одной сетью IPv6 и другой сетью IPv6, и в действительности трансляцию нужно использовать только в тех случаях, когда невозможно применение никаких других средств связи. Например, NAT-PT может также иметь естественное IPv6-соединение и/или некоторый вид туннелированного IPv6-соединения. Эти виды соединений должны быть предпочтительными по сравнению с использованием механизма трансляции, если они возможны, поскольку NAT-PT представляет собой всего лишь инструмент, используемый для поддержки перехода к естественным взаимодействиям между IPv6 и IPv6.

Ограничения трансляции протоколов

Некоторые поля IPv4 в IPv6 имеют измененный смысл, и трансляция оказывается не прямолинейной. Например, семантика и синтаксис заголовков расширения в IPv6 существенно изменились. Детали трансляции протоколов IPv4 в IPv6 даны в [4].

Эффект трансляции адресов

Поскольку NAT-PT выполняет трансляцию адресов, приложения, которые передают IP-адрес на более высоких уровнях, не будут работать. В этом случае, чтобы обеспечить поддержку таким приложениям, необходимо встроить шлюз прикладного уровня (ALG). Это общая проблема NAT, и она полностью описана в [6].

Отсутствие сквозной защиты

Одним из наиболее важных ограничений механизма NAT-PT является тот факт, что на сетевом уровне невозможно обеспечить сквозную защиту. Кроме того, может оказаться невозможной защита на транспортном и прикладном уровнях для тех приложений, которые передают IP-адреса на прикладном уровне. Это естественное ограничение функции трансляции сетевых адресов. Заметим, что независимо от механизмов адресной трансляции, организовать сквозную защиту IPSec через различные адресные области невозможно. Два оконечных узла, которым необходима защита на сетевом уровне IPSec, должны поддерживать либо IPv4, либо IPv6.

Трансляция DNS и DNSSEC

Описанная выше схема NAT-PT включает в себя трансляцию сообщений DNS. Ясно, что эта схема не может разворачиваться в комбинации с защищенным DNS. Т.е. полномочный сервер имен DNS в домене V6 не может подписывать ответы на запросы, которые порождаются из сети V4. В результате оконечный узел V4, который требует, чтобы DNS-ответы были подписаны, будет забраковывать ответы, которые были изменены NAT-PT. Однако за это ограничение вынуждены расплачиваться только серверы в домене V6, которые должны быть доступны из сети V4, поскольку оконечные узлы V4 могут не обращаться к V6-серверам из-за того, что DNS запросы не подписываются. Заметим, правда, что можно организовать передачи зон между серверами DNSSEC внутри одной и той же сети V6.

6. Область применения механизма NAT-PT

NAT-PT может быть полезным инструментом обеспечения совместимости на границе оконечной сети, которая развернута и базируется только на протоколе IPv6, когда она подсоединяется к сети Internet, которая, в свою очередь, либо полностью базируется только на протоколе IPv4, либо представляет собой комбинацию сетей IPv4 и IPv6.

NAT-PT в своем простейшем виде, без поддержки шлюза DNS-ALG, обеспечивает только одностороннюю связь между оконечным доменом IPv6 и сетью IPv4, означающую, что только сеансы связи, инициализированные узлами IPv6, внутренними по отношению к этому домену IPv6, могут транслироваться, в то время как сеансы связи, инициированные узлами IPv4, просто прерываются. Это делает NAT-PT полезным инструментом для оконечных сетей, базирующихся только на IPv6, которым требуется поддержка коннективности с миром IPv4 без необходимости развертывания серверов, доступных из мира IPv4. NAT-PT, объединенный со шлюзом DNS-ALG, обеспечивает двунаправленную коннективность между оконечным доменом IPv6 и миром IPv4, и позволяет инициировать сеансы связи IPv4 узлам, находящимся вне этого домена IPv6. Это делает NAT-PT полезным для оконечных сетей, базирующихся только на IPv6, которым требуется развертывание серверов, доступных из мира IPv4.

Некоторые приложения для своей работы рассчитывают на определенную степень стабильности адресов. Для таких приложений динамическое повторное использование адресов в NAT-PT может оказаться неприятным. Для хостов, выполняющих эти критические к адресам приложения, NAT-PT может быть сконфигурирован так, чтобы обеспечить статическое отображение адресов между V6-адресом хоста и конкретным V4-адресом. Это гарантирует, что выполняемые NAT-PT связанные с адресами изменения не станут существенным источником ошибок в работе.

7. Реализация механизмов трансляции протоколов и адресов

В рамках проектов, поддержанных грантами РФФИ, нами были реализованы бесконтекстный протокольный транслятор SIIT (Stateless IP/ICMP Translation Algorithm [4]) и адресно-протокольный транслятор с сохранением состояния соединений NAT-PT (Network Address Translation – Protocol Translation [5]) для операционных систем Linux и FreeBSD [15, 16]. Ниже описаны некоторые особенности нашей реализации [18].

7.1. Место транслятора в архитектуре ОС

Транслятор выполнен в виде сетевого модуля ядра. Это позволяет избежать выполнения различных действий над приходящими пакетами непосредственно в трансляторе (например, проверка контрольных сумм), так как такую работу будет выполнять стек IP операционной системы. После загрузки модуля в ядро транслятор выглядит как обычный Ethernet-интерфейс с именем ntpt0. И над ним можно выполнять все операции, доступные для обычных сетевых интерфейсов. Узел, на котором выполняется транслятор, должен быть пограничным маршрутизатором между IPv6-only сетью и IPv4 внешним миром. Т.е. он должен иметь как минимум два реальных сетевых интерфейса, к одному

из которых подключена IPv6-only сеть, а к другому – IPv4 сеть. Следующий рисунок поясняет местоположение транслятора в ядре операционной системы.

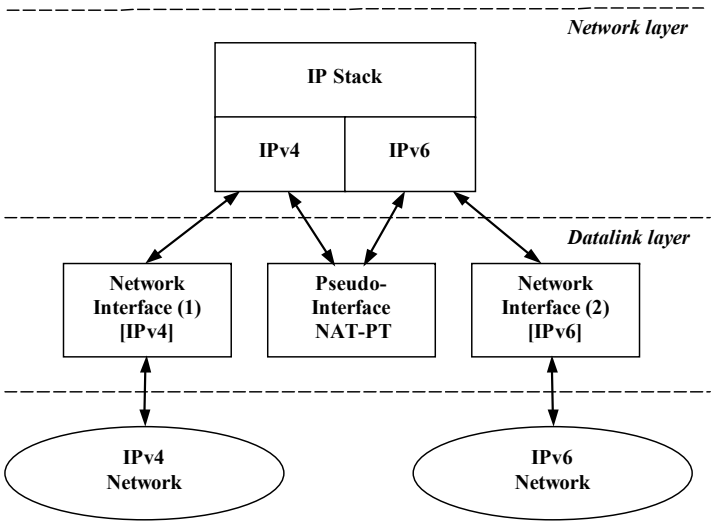

Рис. 5. Место транслятора в сетевой архитектуре ядра.

В таком варианте реализации код транслятора будет выполнять только операции, связанные непосредственно с трансляцией, полагаясь в остальном на IP-стек ядра. Чтобы заставить транслятор работать по схеме, представленной на рисунке, необходимо, используя стандартные средства механизма маршрутизации, направлять на интерфейс транслятора:

- IPv6-пакеты с префиксом адреса назначения, равным PREFIX::/96, и пришедшие на интерфейс IPv6-only сети;
- IPv4-пакеты с адресами назначения из пула адресов, предназначенных для динамического выделения IPv6-only узлам, и пришедшие на интерфейс IPv4 сети.

Для этого необходимо добавить соответствующие записи в таблицу маршрутизации и присвоить нужные адреса интерфейсу транслятора.

Таким образом, пакет, пришедший из IPv4-сети, будет попадать в IP-стек (IPv4) и стандартно на нем обрабатываться. Если этот пакет предназначен для адреса из пула динамических IPv4-адресов IPv6-only узлов, то такой пакет будет направляться в интерфейс транслятора и, если существует соответствие адресов IPv4/IPv6 в таблице транслятора, преобразовываться на нем в IPv6-пакет. Этот IPv6-пакет затем будет передан транслятором обратно в IP-стек (но уже IPv6), который направит его в соответствующий сетевой интерфейс, базируясь на адресе места назначения.

Пришедший из IPv6-сети пакет также попадет в IP-стек (IPv6), и если адрес назначения имеет префикс PREFIX::/96, пакет будет отправлен на транслятор и в случае соответствия адресов в таблице транслятора будет преобразован в IPv4-пакет. И этот пакет снова вернется в IP стек (IPv4) и далее будет направлен в нужный интерфейс для отправки.

Теперь попытаемся пояснить все это на реальном примере нашего рабочего лабораторного стенда, схема которого представлена ниже на рис. 6.

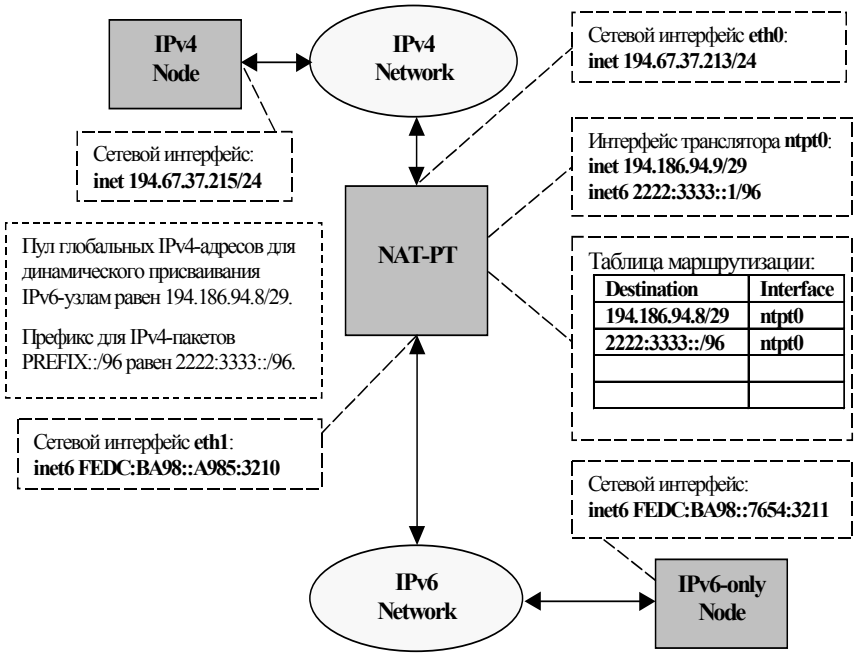

Рис. 6. Схема лабораторного стенда.

Узел, на котором выполняется транслятор, является маршрутизатором и имеет три сетевых интерфейса (помимо интерфейса loopback):

- ***eth0*** – для подключения IPv4 сети,
- ***eth1*** – для подключения IPv6 сети,
- ***ntpt0*** – псевдоинтерфейс NAT-PT транслятора.

Для динамического выделения IPv4-адресов в NAT-PT используются адреса 194.186.94.8/29 (маска 255.255.255.248). Как можно увидеть на рисунке, IPv6-only узел и IPv6-интерфейс eth1 на NAT-PT имеют реальные unicast IPv6-адреса с префиксом FEDC:BA98::/64. Интерфейсу транслятора ntpt0 назначается реальный IPv4-адрес 194.186.94.9/29 из пула динамических адресов и IPv6-

адрес с префиксом 2222:3333::/96. Кроме того, добавляются необходимые записи в таблицу маршрутизации, как показано на рисунке. На IPv6-only узлах необходима запись в таблице маршрутизации для направления на узел с транслятором пакетов с адресами назначения, имеющими префикс 2222:3333::/96, а на IPv4-узлах необходима запись для направления на узел, где выполняется транслятор, пакетов с адресами назначения из диапазона 194.186.94.8/29.

Таким образом, IPv4-пакеты, приходящие на интерфейс eth0 маршрутизатора NAT-PT, с адресом назначения из диапазона 194.186.94.8/29 будут направляться на модуль транслятора ntpt0, на нем транслироваться, и далее направляться на интерфейс eth1 и отправляться в IPv6-сеть. Приходящие на eth1 IPv6-пакеты, у которых адрес назначения имеет префикс 2222:3333::/96, будут направляться на интерфейс ntpt0, преобразовываться на нем в IPv4-пакеты и переправляться на eth0 для отправки в IPv4-сеть.

7.2. Описание реализации транслятора

NAT-PT реализован на основе готовых модулей трансляции протоколов из реализованного нами алгоритма SIIT с добавлением соответствующей этому механизму адресной и портовой трансляции и шлюзов DNS-ALG и FTP-ALG (рис. 7).

Рис. 7. Схема отвечающей за трансляцию части модуля NAT-PT.

Входной точкой и распределительным модулем NAT-PT является функция if_output псевдоинтерфейса. Она вызывается системой всякий раз, когда

интерфейс свободен и у него в очереди находится сетевой пакет. Далее определяется протокол и вызывается соответствующий обработчик. В случае сообщений об ошибке ICMP [13] или ICMPv6 [14] требуется трансляция вложенных IP-заголовков. Это реализовано как рекурсивный вызов обработчика трансляции соответствующего IP.

Однако следует отметить тонкий момент. Дело в том, что в этом вложенном пакете отправитель и получатель поменялись местами по сравнению с обычным прохождением пакета через модуль транслятора NAT-PT. Для корректной обработки таких пакетов используется глобальный флаг, определяющий, рекурсивный это вызов или нет. Переведённый пакет возвращается по цепочке обратно распределительному модулю и помещается во входную очередь соответствующего протокола. Шлюзы прикладного уровня подключены к двум блокам трансляции, а не к распределительному модулю, потому что им также необходимо знать, в каком направлении через NAT-PT следует данный пакет. В принципе, они тоже могут быть разделены на две части, но в нашей реализации мы решили этого не делать, так как некоторая часть трансляции одинакова для обоих направлений следования пакетов. Кроме того, чтобы определить, что этот пакет является DNS/FTP-сообщением, нам необходимо узнать, является ли этот пакет TCP/UDP-сообщением, и если является, узнать его порты. А это невозможно без знания протокола сетевого уровня и поля, хранящего идентификатор протокола транспортного уровня. Поэтому самое логичное, на наш взгляд, место вызова этих модулей – после разбора пришедшего пакета и копирования его данных во временные переменные. Затем, если нужно, вызываются блоки DNS/FTP-трансляции, потом мы транслируем адреса и порты и собираем выходящий пакет. Если где-то на этом пути возникла ошибка, то пакет отбрасывается.

7.3. Особенности реализации шлюза DNS-ALG

Заметим, что предложенный в RFC2766 [5] и реализованный на первом этапе проекта алгоритм работы DNS-ALG описан недостаточно детально и приводит к появлению целого ряда проблем. Спецификация шлюза вызывает оживленную дискуссию в сообществе Internet уже не первый год. В дополнение к определению DNS-шлюза в RFC 2663 был издан документ RFC 2694 “DNS extensions to Network Address Translators (DNS_ALG)” [7], рассматривающий варианты применения NAT с этим шлюзом. Кроме того, существует несколько проектов документов (IETF drafts), описывающих проблемы совместного использования NAT-PT и DNS-ALG и возможные варианты их решения, а именно: http://www.ietf.org/internet-drafts/draft-durand-v6ops-natpt-dns-alg-issues-01.txt, http://www.ietf.org/internet-drafts/draft-hallin-natpt-dns-alg-solutions-02.txt, http://www.ietf.org/internet-drafts/draft-ietf-dnsop-ipv6-dns-issues-03.txt.

Поэтому нам пришлось на основании этих документов и спецификации протокола DNS разработать описанную выше детальную спецификацию DNS-ALG, которую мы использовали в своей реализации.

Кроме того, при разработке и последующем тестировании модуля DNS-ALG мы столкнулись с рядом проблем, не оговоренных ни в одном из вышеупомянутых документов.

Во-первых, применяется метод сжатия сообщений DNS, основанный на использовании ссылок на уже описанное доменное имя вместо повторного его описания. При коррекции сообщения DNS с изменением его размера, мы должны соответствующим образом скорректировать и данные ссылки. Для этой цели мы используем журнал изменений. При последовательном анализе сообщения все ссылки на доменные имена в этом журнале проверяются и, если необходимо, корректируются.

Во-вторых, если в качестве транспортного протокола DNS используется TCP, то требуется коррекция заголовка TCP в связи с изменением размера сообщения. Имеются в виду порядковые номера ack_num (AN) и syn_num (SN). Однако, несмотря на то определение, что «пакеты, пришедшие на 53-й порт протоколов TCP/UDP, должны передаваться в модуль трансляции сообщений DNS», эта особенность протокола TCP в документах IETF полностью проигнорирована. Ввиду данной особенности и редкости использования протокола TCP для разрешения имен, поддержка TCP в DNS-ALG нами не реализована.

В-третьих, рассматриваемое в документах IETF множество вариантов использования NAT со шлюзом DNS-ALG не обладает полнотой. Например, в процессе тестирования нами обнаружен следующий вариант:

Внешний IPv4-хост посылает запрос типа "A" на внутренний DNSv6-сервер, причём запрашивается доменное имя такого же IPv4-only хоста. В этом случае DNSv6-сервер, получив запрос «AAAA» (он был транслирован при пересечении границы IPv4-IPv6) и не найдя у себя соответствующей записи, должен обратиться с этим запросом к внешним DNSv4-серверам, при условии, что он выполняет рекурсивные запросы. Это следует из необходимости обеспечивать работу DNS-сервиса внутренних IPv6-хостов и прямо описано в RFC. Этот запрос, транслируясь в «A», доходит до внешних DNS-служб и получает «A»-ответ, который транслируется в «AAAA» и доставляется на DNSv6-сервер. Последний, не кэшируя ответ, пересылает его наружу, и возникает достаточно интересная ситуация. Шлюз, встретив в ответе незнакомый IPv6-адрес вида PREFIX:X.X.X.X, в который был транслирован IPv4-адрес, полученный от внешних серверов, должен выделить ему временный IPv4-адрес из своего пула. Если такое выделение произойдет и получивший наконец ответ IPv4-хост будет пытаться открыть соединение с запрошенным хостом по полученному им адресу, возникнет ситуация, при которой два IPv4-хоста при наличии между ними прямого IPv4-пути будут взаимодействовать через NAT. Кроме накладных расходов, подобное соединение может быть чревато злонамеренной и успешной DoS-атакой на NAT, так как у установленных соединений тайм-аут обрыва связи и удаления временного соответствия адресов в несколько раз больше соответствующего тайм-аута временных записей. Метод решения этой проблемы прост – либо

выделять IPv4-адреса только для IPv6-адресов, не подпадающих под маску PREFIX:0.0.0.0 (административный вариант), либо, как это сделано у нас, при выделении временных записей проверять IPv6-адрес на соответствие этой маске. Если ответ положительный, то новую запись не выделять, а в качестве IPv4-адреса использовать последние 32 бита. Придумать такие методы нетрудно, но есть вероятность, что недостаточно хорошо протестированные реализации столкнутся с этой проблемой только на практике.

7.4. Особенности реализации шлюза FTP

Как уже было отмечено, все команды FTP передаются по управляющему соединению. Так как для корректной работы протокола FTP необходимо транслировать только управляющие команды PORT, PASV, EPRT, EPSV, FTP-ALG должен обрабатывать только управляющее соединение FTP. Соединения данных остаются без изменения (т.е. FTP-ALG их не затрагивает). Управляющее соединение инициируется клиентом на TCP-порт 21 сервера. Поэтому все приходящие TCP-пакеты, у которых порт отправителя или получателя равен 21, направляются в модуль FTP-ALG. Модуль FTP-ALG работает с собственным списком записей о соединениях FTP. Следует отметить, что алгоритм работы FTP-ALG, предложенный в RFC2766 [5], также описан недостаточно детально. Потому мы доработали этот алгоритм, добавив необходимую функциональность. В нашей реализации для обработки FTP-соединений используется специальная таблица, записи которой хранят информацию об управляющем соединении FTP. Все приходящие TCP-пакеты предварительно обрабатываются модулем обработки TCP-соединений, и пакеты, относящиеся к управляющим соединениям FTP (эти пакеты имеют порт источника или порт назначения 21), направляются в модуль FTP-ALG.

Если соединение новое, в таблицу заносится новая запись, если нет, производится необходимая корректировка существующей записи и трансляция команды FTP, если необходимо. Затем оттранслированные данные возвращаются в модуль обработки TCP, который выполняет окончательное формирование выходного пакета TCP. Особенностью данного шлюза является реализация алгоритма коррекции порядковых номеров ACK и SYN заголовка TCP (этот алгоритм в RFC-спецификации не описан). Отметим ту его особенность, что он корректно работает и в том случае, когда одновременно приходят несколько пакетов с одной из сторон, так как он ведет всю историю дельт номеров SN и AN пакетов FTP-соединения.

7.5. Управление модулем NAT-PT

Для управления модулем NAT-PT и его конфигурирования была реализована специальная утилита. Для получения и передачи управляющей информации в модуль NAT-PT она использует системную функцию ioctl(), подобную хорошо знакомой каждому системному администратору UNIX утилите для настройки сетевых интерфейсов ifconfig. Однако, в отличие от последней, она почти не работает со стандартными системными флагами, предназначенными для этой

цели. Дело в том, что используемой в таких случаях структуры ifreq зачастую бывает недостаточно, и нам нужны другие, большие структуры. Поэтому, поскольку сетевые подсистемы используемых нами операционных систем позволяют разработчикам драйверов сетевых устройств вводить специфические для своего устройства флаги, мы этой возможностью и воспользовались. Реализация сетевого стека передаёт их без изменений функции интерфейса, специфицированной в описывающей его структуре как if_ioctl. На данный момент наша утилита позволяет задавать отображения IPv4-адресов (и комбинаций IPv4-адрес + список портов) на IPv6-адреса и регулировать тайм-ауты.

8. Тестирование сетевых модулей

Одним из важнейших этапов разработки сетевого программного обеспечения является тестирование [19]. Для тестирования нашей реализации транслятора NAT-PT помимо обычных методов проверки работоспособности программ мы использовали методологию UniTesK [17], а именно ее реализацию для языка программирования Си, которая получила название CTesK. Методология UniTesK и продукт CTesK разработаны одной из исследовательских групп нашего института.

В технологии UniTesK используются два подхода к тестированию: "черный ящик", когда поведение тестируемой системы наблюдается извне только на основании выданных стимулов и полученных реакций, и "белый ящик", когда поведение тестируемой системы наблюдается посредством доступа к внутренним данным тестируемой системы. В своих тестах мы использовали подход "черный ящик". Основной целью тестирования была проверка работоспособности модуля NAT-PT в соответствии со спецификацией RFC 2766.

Процесс тестирования в UniTesK, как правило, состоит из следующих фаз: определение интерфейса, разработка спецификаций, разработка тестов, прогон тестов, анализ результатов.

8.1. Определение интерфейса

На этом этапе выделяется множество воздействий на тестируемую систему (стимулов) и реакций в ответ на эти воздействия. Множество стимулов и реакций определяют набор требований, которые должны быть описаны в формальных спецификациях.

В случае NAT-PT стимулами выступают входящие пакеты со стороны IPv4 или IPv6, а реакциями – выходящие пакеты IPv4 или IPv6. Особенностью модуля транслятора NAT-PT является то, что он играет роль маршрутизатора, т.е. IPv4-пакет, полученный на IPv4-интерфейсе и предназначенный для узла IPv6-сети, будет транслирован модулем NAT-PT в IPv6-пакет и передан для отправки на IPv6-интерфейс. Таким образом, если стимулом является IPv4-пакет, то реакцией будет IPv6-пакет и наоборот. Реакции на стимул может не быть в случае, когда данный тип пакетов трансляции не подлежит, такие

пакеты просто отбрасываются транслятором с посылкой ICMP-сообщения отправителю или без посылки такого сообщения. Кроме того, при определенных стимулах реакцией может быть несколько пакетов.

В наших тестах мы ограничились тестированием трансляции пакетов протоколов ICMP и UDP. Также мы исключили из рассмотрения реакции оконечных узлов на получение оттранслированных пакетов.

8.2. Разработка формальных спецификаций

На этапе разработки спецификаций требования записываются в формальном виде. В технологии CTesK формальные спецификации выражаются средствами специального расширения языка Си, получившего название SEC (Specification Extention of the C language).

Спецификация тестируемой системы состоит из следующих компонентов: абстрактного состояния, спецификации стимулов, спецификации реакций и вспомогательных процедур.

Так как для тестирования мы используем модель "черного ящика", нам необходимо моделировать поведение тестируемого модуля NAT-PT. Абстрактное состояние – это набор данных, который моделирует внутренние данные тестируемой системы. Эти данные меняются всякий раз, когда целевая (тестируемая) система подвергается воздействию стимулов или выдает реакцию. В нашем случае в абстрактном состоянии мы сохраняем данные о входящих и исходящих пакетах, на основании чего формируется информация о состоянии соединений и выделении адресов.

Стимулами выступают входящие пакеты ICMP и UDP протокола IPv4 для тестирования трансляции из IPv4 в IPv6 и входящие пакеты ICMP и UDP протокола IPv6 для тестирования трансляции из IPv6 в IPv4. Реакциями на данные стимулы являются исходящие пакеты ICMP и UDP протоколов IPv4 и IPv6 соответственно. Спецификации для стимулов (для входящих пакетов) разбирают входящий пакет и сохраняют данные в абстрактном состоянии. Спецификации реакций (для исходящих пакетов) на основании абстрактного состояния устанавливают, что данный пакет является допустимым, а также проверяют корректность трансляции.

8.3. Разработка тестов

Связь между моделью и реализацией осуществляется посредством медиаторов. Такие медиаторы передают стимулы из тестовой системы в целевую систему, регистрируют реакции целевой системы и передают их в тестовую систему, а также отображают изменения состояния целевой системы в абстрактное состояние.

Для медиаторов, реализующих посылку пакетов в тестируемую систему, используются дополнительные узлы в обоих сегментах сети IPv4 и IPv6, на которых установлен генератор сетевых пакетов, позволяющий выдавать пакеты в сеть по заданным параметрам. Специальные медиаторы стимулов и реакций осуществляют сбор входящих и исходящих пакетов на обоих сетевых

интерфейсах. Для этого устанавливаются специальные программы прослушивания сети. Все полученные пакеты собираются в специальный буфер и передаются в тестовую систему для обработки.

Набор и последовательность стимулов определяется тестовым сценарием. Тестовый сценарий определяет стимул для воздействия на тестовую систему. Медиатор выдачи стимула передает стимул в тестовую систему. После этого специальные медиаторы в течение определенного интервала времени собирают реакции, которые происходят после выдачи стимула, и производят отображение состояния системы в абстрактном состоянии. Затем производится проверка на допустимость набора реакций целевой системы на данный стимул. При допустимости реакций для каждой из них вызываются оракулы стимулов и реакций. Если все оракулы возвращают положительный вердикт, то тестовый сценарий переходит к следующему стимулу, иначе тестовый сценарий сообщает об ошибке и завершает работу.

Тестовый набор состоит из нескольких тестовых сценариев. Тестовые сценарии проверяют трансляцию заголовков IPv4 в IPv6 и обратно, ICMPv4 в ICMPv6 всех типов и обратно, UDPv4 в UDPv6 и обратно.

8.4. Результаты тестирования

Прогон тестов выявил ряд ошибок в реализации NAT-PT. Это позволило локализовать местонахождение ошибок в коде реализации. Выявленные ошибки были связаны как с несоответствием спецификации, так и с программными ошибками в реализации.

8.5. Генератор сетевых пакетов

Для тестирования разработанного нами транслятора NAT-PT потребовались средства генерации и отправки различных тестовых последовательностей пакетов с заданными параметрами. Существующие генераторы по различным причинам нам не подходили, поэтому пришлось создать свой генератор сетевых пакетов, отвечающий нашим требованиям.

Разработанный нами генератор сетевых пакетов позволяет создавать основные типы пакетов TCP, UDP, ICMP протоколов IPv4 и IPv6 и отправлять их в сеть. Генератор позволяет посылать как одиночные пакеты заданного типа, так и серии пакетов. Акцент был сделан на его универсальности: возможности создавать все необходимые типы пакетов, простоте управления параметрами и добавления новых типов пакетов.

Генератор состоит из двух основных модулей. Первый создает пакет, второй отправляет его в сеть. Генерация пакетов происходит путем последовательного формирования его заголовков. Главной особенностью генератора является наличие специального внешнего файла, в котором описаны структуры заголовков. В качестве входных параметров генератор получает файл с описанием заголовков и пользовательские данные.

Файл описания заголовков представляет собой текстовый файл в формате языка XML. Этот язык позволяет представить заголовки пакета и их поля в виде дерева и значительно упрощает поиск нужных элементов. В файле описаны все поля известных заголовков с указанием их длины, обязательности использования и значения по умолчанию. Это дает пользователю возможность передавать только часть полей заголовка. Структура, формируемая самим пользователем, – это список заголовков пакета, включающий только те параметры, которые необходимо изменить. Генератор последовательно просматривает эту структуру и находит соответствующий заголовок в файле описания заголовков. Значения полей формируются из входной структуры, или подставляются значения по умолчанию, или значения вычисляются с помощью специальных функций. Для создания каждого заголовка используется отдельная функция, однако общие принципы формирования заголовков и введение специальных типов данных позволяют для некоторых заголовков использовать одинаковые функции, а значит, для добавления нового заголовка иногда достаточно добавить его описание в файл заголовков. Эта часть генератора является платформеннонезависимой.

Вторая часть генератора отправляет сформированный пакет непосредственно в сеть и позволяет посылать как одиночные пакеты, так и серии пакетов. Эта часть генератора реализована для платформ FreeBSD и Linux.

Таким образом, внешний файл описания заголовков позволяет легко изменять параметры пакетов, значительно сокращает количество данных, передаваемых пользователем, дает возможность добавлять новые типы заголовков, для чего иногда достаточно описать новый заголовок, без добавления соответствующей функции.

Генератор является библиотекой, написанной на языке C, и может встраиваться в другие программные продукты. Генератор прошел тестирование на платформах FreeBSD и Linux и применялся при тестировании нашей системы NAT-PT.

9. Заключение

В результате проведенной работы создан полнофункциональный межпротокольный шлюз NAT-PT, соответствующий проекту стандарта IETF RFC 2766 “Network Address Translation – Protocol Translation” и оснащенный шлюзами прикладного уровня DNS-ALG и FTP-ALG. NAT-PT может быть полезным инструментом обеспечения совместимости на границе оконечной сети, которая развернута и базируется только на протоколе IPv6, когда она подсоединяется к сети Internet, которая, в свою очередь, либо полностью базируется только на протоколе IPv4, либо представляет собой комбинацию сетей IPv4 и IPv6. Реализованный механизм представляется одним из важных средств для обеспечения плавного и безболезненного перехода на протокол IPv6.

На основе документов IETF были разработаны подробные спецификации NAT-PT. Работа по его реализации проведена на базе ОС Linux (с использованием реализации IPv6 проекта USAGI http://www.linux-ipv6.org) и FreeBSD (с

использованием реализации IPv6 проекта KAME http://www.kame.net). На отладочном стенде проведена отладка и тестирование разработанного программного обеспечения. К разработанным программным средствам обеспечен свободный доступ с Web-сайта института (http://ipv6.ispras.ru).

С целью проверки работоспособности транслятора в «боевых» условиях большой сетевой нагрузки в ИСП РАН развернут специальный стенд. Для этого в экспериментальной сети за транслятором NAT-PT развёрнуты IPv6-серверы HTTP и FTP с зеркалами популярных серверов, а также IPv6 DNS-сервер. Серверы доступны извне по адресам http://www.ipv6-test.ispras.ru и ftp://ftp.ipv6-test.ispras.ru.

Литература

1. RFC 2460 “Internet Protocol, Version 6 (IPv6) Specification”, http://www.ietf.org/rfc/rfc2460.txt.
2. RFC 2373 “IP Version 6 Addressing Architecture”, http://www.ietf.org/rfc/rfc2373.txt.
3. RFC 791 “Internet Protocol”, http://www.ietf.org/rfc/rfc791.txt.
4. RFC 2765 “Stateless IP/ICMP Translator (SIIT)”, http://www.ietf.org/rfc/rfc2765.txt.
5. RFC 2766 “Network Address Translation - Protocol Translation (NAT-PT)”, http://www.ietf.org/rfc/rfc2766.txt.
6. RFC 2663 “IP Network Address Translator (NAT) Terminology and Considerations”, http://www.ietf.org/rfc/rfc2663.txt.
7. RFC 2694 “DNS extensions to Network Address Translators (DNS_ALG)”, http://www.ietf.org/rfc/rfc2694.txt.
8. RFC 3596 “DNS Extensions to Support IP Version 6”, http://www.ietf.org/rfc/rfc3596.txt.
9. RFC 793 “Transmission Control Protocol”, http://www.ietf.org/rfc/rfc793.txt.
10. RFC 768 “User Datagram Protocol (UDP)”, http://www.ietf.org/rfc/rfc768.txt.
11. RFC 959 “File Transfer Protocol (FTP)”, http://www.ietf.org/rfc/rfc959.txt.
12. RFC 2428 “FTP Extensions for IPv6 and NATs”, http://www.ietf.org/rfc/rfc2428.txt.
13. RFC 792 “Internet Control Message Protocol Specification“, http://www.ietf.org/rfc/rfc792.txt.
14. RFC 2463 “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification”, http://www.ietf.org/rfc/rfc2463.txt.
15. Linux Man Pages, http://www.tldp.org/docs.html#man.
16. FreeBSD Man Pages, http://www.freebsd.org/cgi/man.cgi.
17. И.Б.Бурдонов, А.С.Косачев, В.В.Кулямин, А.К.Петренко "Подход UniTesK к разработке тестов","Программирование", 2003, №6–стр 25-43.
18. Г.В. Ключников, Д.С. Мишин, Д.В. Москалев, В.З. Шнитман «Механизмы перехода с IPv4 на IPv6. Использование методов трансляции протоколов и адресов для обеспечения совместимости протоколов IPv4 и IPv6» в сб. Тезисов международной конференции «Интернет нового поколения – IPv6», г. Ярославль, 2002, стр. 23-28.
19. Г.В. Ключников, А.В. Никешин, Д.С. Мишин, Д.В. Москалев, В.З. Шнитман «Тестирование сетевых модулей и генератор сетевых пакетов» в сб. Тезисов международной конференции «Интернет нового поколения – IPv6», г. Ярославль, 2003, стр. 26-31.