

Network Working Group  
Request for Comments: 3404  
Obsoletes: 2915, 2168  
Category: Standards Track

M. Mealling  
VeriSign  
October 2002

## Система DDDS. Часть 4 – Приложение для преобразования URI.

### Dynamic Delegation Discovery System (DDDS) Part Four: The Uniform Resource Identifiers (URI) Resolution Application

#### Статус документа

Этот документ содержит спецификацию стандартного протокола, предложенного сообществу Internet, и служит приглашением к дискуссии в целях развития. Текущее состояние стандартизации и статус описанного здесь протокола можно узнать из документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

#### Авторские права

Copyright (C) The Internet Society (2002). All Rights Reserved.

#### Тезисы

Этот документ описывает спецификацию приложения для поиска информации о URI<sup>1</sup> путем определения полномочных серверов по значению URI. Данный метод используется для того, чтобы показать, что полномочный сервер относится к системе DDDS<sup>2</sup>.

Документ также является частью серии, полностью указанной в документе Dynamic Delegation Discovery System (DDDS) Part One: The Comprehensive DDDS (RFC 3401). Важно подчеркнуть, что понимание любого документа этой серии невозможно без прочтения других документов.

## Оглавление

1. Введение.....	1
2. Терминология.....	2
3. Различия между URN и URI.....	2
4. Спецификации приложений для преобразования URI и URN.....	2
4.1 Уникальная строка Приложения.....	2
4.2 Первое общеизвестное правило.....	2
4.3 Флаги.....	3
4.4 Параметры сервиса.....	3
4.4.1 Службы.....	3
4.4.2 Протоколы.....	3
4.4.3 Применимость сервиса.....	3
4.5 Корректные базы данных.....	4
5. Примеры.....	4
5.1 Пример использования URN.....	4
5.2 Пример схемы CID URI.....	5
5.3 Преобразование для схемы HTTP URI.....	5
6. Примечания.....	6
7. Согласование с IANA.....	6
8. Вопросы безопасности.....	6
9. Благодарности.....	6
Литература.....	6
Приложение А. Псевдокод.....	7
Адрес автора.....	8
Полное заявление авторских прав.....	8

### 1. Введение

Система DDDS обеспечивает возможность организации связей между строками данных для поддержки систем передачи полномочий (делегирования) с динамической конфигурацией. Работа DDDS основана на отображении некой уникальной строки на данные, хранящиеся в Базе данных DDDS путем итеративного применения правил преобразования строк до выполнения условий завершения.

<sup>1</sup>Uniform Resource Identifiers – однотипные идентификаторы ресурсов.

<sup>2</sup>Dynamic Delegation Discovery System – динамическая система детектирования передачи полномочий (делегирования).

Этот документ описывает Приложение DDDS для преобразования идентификаторов URI. Документ не определяет Алгоритм и Базу данных DDDS. Серия документов, которая задает эти спецификации, указана в документе Dynamic Delegation Discovery System (DDDS) Part One: The Comprehensive DDDS (RFC 3401) [1]. Важно подчеркнуть, что понимание любого документа этой серии невозможно без прочтения других документов.

Идентификаторы URI обеспечивают значительные преимущества при поиске ресурсов, доступных через Internet. Однако в процессе работы с этими идентификаторами было обнаружено множество проблем. Рабочая группа Uniform Resource Identifier предложила разработку URN<sup>1</sup> [8] для использования в качестве постоянных и не зависящих от места расположения идентификаторов ресурсов Internet. Эти идентификаторы позволяют решить большинство проблем, возникающих при использовании URI. В RFC 1737 [6] были определены требования к URN.

В течение срока работы группы URI-WG было внесено множество предложений по URN. В процессе разработки этих предложений происходило несколько встреч, приведших к выработке компромиссного решения, известного как Knoxville framework. Основным результатом схемы Knoxville является то, что система преобразования должна быть отделена от пути присваивания имен. Это расходится с большинством URI, которые указывают хост для контакта и используемый протокол. Читателям рекомендуется обратиться к работе [7], содержащей информацию о схеме Knoxville, а также о контексте и целях этого предложения.

Разделение путей преобразования и создания имен обеспечивает несколько преимуществ. Возможно использование множества вариантов именования и преобразования, а также использование различных протоколов и преобразователей (resolver). Но в связи с разделением возникает одна проблема – как преобразовать имя, когда оно не указывает нам направление на преобразователь имен?

На первый взгляд система DNS<sup>2</sup> представляется основным кандидатом для схемы преобразования, поскольку эта система широко распространена и понятна. Однако DNS не подходит для поддержки информации на поресурсной основе. Прежде всего, система DNS никогда не была рассчитана на обслуживание столь большого числа записей. Во-вторых, ограниченный размер записи не подходит для каталожной информации. В-третьих, доменные имена не подходят для URN.

В результате здесь предлагается использовать DDDS для поиска «преобразователей», которые могут обеспечить информацию об отдельном ресурсе, потенциально включая сам этот ресурс. Для достижения цели мы «перепишем» URI в Ключ, следуя правилам, полученным с помощью DDDS. В этом документе описывается URI Resolution, как приложение DDDS и задается спецификация использования по крайней мере одной Базы данных на основе DNS.

## 2. Терминология

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с RFC 2119.

Все прочие термины, особенно выделенные шрифтом, заимствованы из спецификации алгоритма DDDS в RFC 3403 [3].

## 3. Различия между URN и URI

С точки зрения данной системы теоретически разница между URI в общем случае и URN в специальном случае отсутствует. На практике же существует разница, которая сдерживает распространение URI. Если преобразование URN коллапсирует в базовое преобразование URI, системы URN могут пострадать от недостатков адаптации преобразования URI.

Решение состоит в дозволении сокращать преобразование URN. В приведенной далее спецификации базовое преобразование URI начинается со вставки правил для известных схем URI в реестр 'uri.agra.'. Для схемы 'URN:' одно из правил, найденных в 'uri.agra.', будет для схемы 'urn'. Это правило будет просто передавать полномочия зоне 'urn.agra.' для получения дополнительных записей NAPTR на основе пространства имен URN. Существенным является то, что Правило перезаписи URI Resolution для 'URN:' является Первым общеизвестным правилом для Приложения URN Resolution.

Следовательно, этот документ содержит спецификации двух Приложений DDDS. Одна спецификация относится к URI Resolution, а другая – к URN Resolution. Обе спецификации технически идентичны, но разделение обеспечивает возможность независимой обработки.

## 4. Спецификации приложений для преобразования URI и URN

Этот шаблон определяет Приложение DDDS по преобразованию URI и URN, согласно правилам и требованиям документа [3]. База данных DDDS, используемая этим Приложением, описана в документе [4], который определяет тип записей NAPTR<sup>3</sup> в системе DNS.

### 4.1 Уникальная строка Приложения

Строкой AUS<sup>4</sup> служит URI или URN, для которого отыскивает полномочный сервер. Эти URI или URN **должны** быть в каноническом представлении и представляться в 16-ричном виде согласно "absolute-uri" из документа Collected ABNF (RFC 2396) [15].

### 4.2 Первое общеизвестное правило

В случае URI первый известный ключ создается путем выполнения схемы URI. В случае URN первым известным ключом является Namespace Identifier. Например, для URI 'http://www.example.com/' первым Ключом будет http. Для URN 'urn:foo:foospace' первым Ключом будет 'foo'.

<sup>1</sup>Uniform Resource Names – однотипные имена ресурсов.

<sup>2</sup>Domain Name System – система доменных имен

<sup>3</sup>Naming Authority Pointer - указатель на уполномоченный сервер именования

<sup>4</sup>Application Unique String – уникальная строка приложения

### 4.3 Флаги

В настоящее время определены только 4 флага - S, A, U и P. Флаги S, A и U используются для завершающего просмотра. Это означает, что Правило является последним и флаг определяет, что будет происходить на следующем этапе. Флаг S означает, что выводом данного Правила является доменное имя, для которого существует одна или несколько записей SRV [9]. Использование типа SRV для преобразования URI и URN рассматривается в главе 5. Флаг A означает, что выводом Правила является доменное имя и следует использовать для этого домена поиск записи типа A, AAAA или A6. Флаг U говорит, что выводом Правила является URI [15].

Флаг P говорит, что оставшаяся часть Алгоритма DDDS игнорируется, а дальнейший процесс зависит от приложения вы выходит за рамки данной спецификации. Приложение может использовать компоненту Protocol, найденную в поле Services, для связанного с Приложением набора правил, который следует выполнять далее. Запись с флагом P является последней записью, которая интерпретируется правилами настоящего документа. Можно трактовать флаг P, как индикатор завершающего поиска, но это будет некорректно, поскольку "завершающее" Правило является концепцией DDDS, а этот флаг показывает, что дальнейшее просто не имеет отношения к DDDS.

Оставшиеся алфавитные флаги зарезервированы для будущих версий данной спецификации. Цифровые флаги могут использоваться для локальных экспериментов. Флаги S, A, U и P являются взаимоисключающими и библиотеки преобразования **могут** генерировать сообщение об ошибке при обнаружении противоречивых флагов (экспериментальный код и программы для помощи при создании Правил перезаписи будут генерировать такие сообщения с большей вероятностью, нежели клиентские программы типа браузеров). Предполагается, что в будущем может быть разрешено включение множества флагов, поэтому для реализаций **недопустимо** предполагать, что поле флагов не может содержать более 1 символа. Если клиент считает, что в записи содержится неизвестный флаг, он **должен** игнорировать такой флаг и перейти к следующему Правилу. Такая проверка имеет более высокий приоритет, нежели какое-либо упорядочение, поскольку флаги могут управлять интерпретацией полей. Новый флаг может изменить интерпретацию поля gehexr и/или replacement так, что станет невозможно определить соответствие записи данной цели.

Флаги S, A и U называются завершающими (terminal), поскольку они прерывают цикл выполнения алгоритма DDDS. Если эти флаги отсутствуют, клиент может предполагать существование другого Правила для Ключа, произведенного текущим Правилем перезаписи.

### 4.4 Параметры сервиса

Параметры сервиса для этого Приложения имеют форму строки символов, следующей приведенной ниже форме ABNF

```
service_field = [ [protocol] *("+" rs) ]
protocol      = ALPHA *31ALPHANUM
rs            = ALPHA *31ALPHANUM
```

; Поля protocol и rs fields ограничены 32 символами и должны начинаться с буквы.

Иными словами, за необязательной спецификацией протокола может следовать 0 или более служб преобразования. Каждый сервис преобразования указывается начальным символом '+'.

Допускаются также пустые строки. Они обычно будут появляться в начале последовательности Правил, когда невозможно узнать, какие службы и протоколы будут предложены в конце конкретного пути передачи полномочий.

#### 4.4.1 Службы

Идентификаторы сервиса, указываемые в поле rs, совпадают для преобразований URI и URN, поскольку сами типы входных значений основаны на схеме URI. Список корректных служб определен в документе [11].

Примерами сервиса могут служить:

I2L: по данному URI возвращается идентификатор URI, указывающий место, где был найден исходный идентификатор URI.

I2Ls: по данному URI возвращается один или множество идентификаторов URI, указывающих места, где можно найти исходный URI.

I2R: по данному URI возвращается один экземпляр ресурса, указанного данным URI.

I2Rs: по данному URI возвращается один или множество экземпляров ресурсов, указываемых данным URI.

I2C: по данному URI возвращается один экземпляр описания ресурса.

I2N: по данному URI возвращается одно имя URN, которое именуется ресурс (Отметим, что эквивалентность применительно к URN является нетривиальным вопросом; см. документ [6], объясняющий причины этого).

#### 4.4.2 Протоколы

Идентификаторы протоколов, которые являются корректными значениями поля Protocol, **должны** определяться документами, относящимися к преобразованию URI. В настоящий момент единственным таким протоколом является THTTP [10].

Очень важно обеспечить, чтобы простого указания протокола в поле Services было недостаточно, поскольку имеется дополнительная семантика, относящаяся к преобразованию URI, которая не определена в протоколе. Например, если протокол Z39.50 указан в качестве корректного, нужно будет дополнительно определить как этот протокол будет представлять запросы для конкретных служб, как будут представляться идентификаторы URI и какая информация будет возвращаться.

#### 4.4.3 Применимость сервиса

В силу возможности существования сложных наборов возможных протоколов и служб, у приложений часто может возникать необходимость использования более сложного процесса выбора, нежели простой просмотр упорядоченного списка протоколов. Например, при наличии 4 применимых правил в последнем значении поля Service может оказаться

более предпочтительным, чем в первом. Но, поскольку клиент может быть удовлетворен первым, он просто не узнает о четвертом, который может оказаться "лучше".

Для упрощения этой задачи у клиента может возникнуть желание несколько изменить алгоритм DDDS (только для данного приложения!), чтобы определять наличие более предпочтительных протоколов и/или служб. Это можно сделать без опасений для данного приложения, используя более сложные итерации между п. 3 и п. 4 алгоритма DDDS для поиска оптимального пути. Например, после того, как клиент нашел правило, в котором Выражение для замены (Substitution Expression) дает результат с подходящим описанием сервиса, клиент может отметить этот факт и продолжить просмотр правил (в порядке, заданном значением Order!) для поиска наилучшего. Если лучшего правила не будет найдено, клиент сможет использовать отмеченное ранее правило.

Следует принимать во внимание, что для безопасной реализации сказанного входные данные п. 3 и выходные данные п. 4 **должны** быть идентичны базовому алгоритму. Для клиентской программы **недопустимы** попытки проведения оптимизации за пределами одного набора Правил перезаписи (т. е., со сменой путей передачи полномочий).

## 4.5 Корректные базы данных

В настоящий момент для Приложения задана только одна База данных DDDS. Документ Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database (RFC 3403) [4] говорит, что это База данных DDDS Database, которая использует DNS-записи NAPTR для хранения правил перезаписи. Ключи для базы данных представляются как доменные имена.

Выводом First Well Known Rule для Приложения URI Resolution является схема URI. Для преобразования этого вывода в уникальный ключ Базы данных к результату в конце добавляется строка '.uri.arpa.'. Полученное в результате доменное имя используется для запроса записей NAPTR, порождающих новые ключи в формате доменных имен.

Выводом First Well Known Rule для Приложения URN Resolution является идентификатор из пространства имен URN. Для преобразования его в ключ в конце идентификатора добавляется строка '.urn.arpa.'. Полученное доменное имя используется для запроса записей NAPTR, которые порождают новые ключи в форме доменных имен.

Серверы DNS **могут** интерпретировать значения Flag и использовать полученную информацию для включения соответствующих записей SRV и A в раздел Additional Information пакета DNS. Клиентам рекомендуется (но не требуется) проверять дополнительную информацию. Более подробное описание работы с разделом Additional Information и записями в откликах DNS можно найти в RFC 3403<sup>1</sup>.

Для представления выражений для замены используется набор символов UTF-8. На входе допускаются все символы, которые могут встречаться в URI. Для Ключей допустимы те символы, которые разрешены в доменных именах DNS. Флаг "i" в выражениях для замены служит для обозначения того, что проверка соответствия выполняется без учета регистра символов.

## 5. Примеры

### 5.1 Пример использования URN

Рассмотрим URN из гипотетического пространства имен FOO. Номера FOO служат идентификаторами приблизительно для 30 миллионов зарегистрированных компаний по всему миру; идентификаторы распределяет и поддерживает компания Fred, Otto and Orvil, Inc. URN могут иметь вид:

```
urn:foo:002372413:annual-report-1997
```

Первым шагом процесса преобразования будет поиск информации о пространстве имен FOO. Идентификатор пространства имен [8] "foo" создается из URN путем добавления к '.urn.arpa.' префикса 'foo', что дает в результате 'foo.urn.arpa.'. Далее делается запрос к DNS для получения записей NAPTR для этого домена. Результат запроса имеет вид:

```
foo.urn.arpa.
;;      order pref flags service          regexp          replacement
IN NAPTR 100 10 "s" "foolink+I2L+I2C" "" foolink.udp.example.com.
IN NAPTR 100 20 "s" "rcds+I2C" "" rcds.udp.example.com.
IN NAPTR 100 30 "s" "thttp+I2L+I2C+I2R" "" thttp.tcp.example.com.
```

Поле order содержит одинаковые значения, показывающие отсутствие упорядочения. Поле preference показывает, что провайдер предпочитает, чтобы клиенты использовали специальный протокол foolink, после этого RCDS и в последнюю очередь THTTP. Все записи содержат флаг s, показывающий, что запись является завершающей и следующим шагом является получение от DNS записи SRV для данного доменного имени.

Поле service говорит, что при использовании протокола foolink можно вводить запросы I2L, I2C или I2R для получения URI или задавать некоторые более сложные вопросы о ресурсе. Можно использовать сервис RCDS<sup>2</sup> [12] для получения некоторых метаданных, тогда как THTTP можно использовать для получения URI, соответствующего текущему местоположению ресурса.

Предположим, что клиент не знает протокола foolink, но знает RCDS. Следующим шагом будет поиск записей SRV RR для имени rcds.udp.example.com, который даст нам список хостов, способных предоставить требуемые услуги преобразования. Этот список может иметь вид:

```
;;      Pref Weight Port Target
rcds.udp.example.com IN SRV 0 0 1000 deffoo.example.com.
IN SRV 0 0 1000 dbexample.com.au.
IN SRV 0 0 1000 ukexample.com.uk.
```

Список содержит три хоста, способных выполнить преобразование и указывает номер порта, который следует использовать для связи с сервером RCDS (получить сведения об интерпретации приведенных выше полей можно из

<sup>1</sup>В оригинале ошибочно дается ссылка на RFC 3404. Прим. перев.

<sup>2</sup>Resource Cataloging and Distribution Service - служба каталогизации и распространения.

спецификации SRV [9]). Здесь существует возможность оптимизации. RFC 3403<sup>3</sup> определяет возможность использования раздела Additional Information. В этом случае записи SRV могут возвращаться в качестве дополнительной информации при завершающем поиске NAPTR (вместе с записями A для этих SRV). Эти записи обеспечивают возможность существенной оптимизации. При использовании этого в комбинации со большими значениями TTL для записей \*.urn.arpa. среднее число обращений к DNS для преобразования большинства URI будет приближаться к 1.

Отметим, что приведенные выше примеры записей NAPTR предназначены для представления результатов поиска NAPTR с использованием некой клиентской программы типа nslookup; администраторам зон следует поддерживать документацию для своих серверов имен с точной информацией о синтаксисе используемых файлов зон.

Отметим также, что может использоваться дополнительный первый шаг, на котором URN преобразуется как базовый вариант URI путем просмотра urn.uri.arpa. Результирующее правило будет задавать, что NID выделяется и URN и суффикс '.urn.arpa.' добавляется для получения первого ключа 'foo.urn.arpa.'.

## 5.2 Пример схемы CID URI

Рассмотрим схему URI на основе MIME Content-Id. URI может иметь вид:

```
cid:199606121851.1@bar.example.com
```

(отметим, что этот пример выбран с учебными целями и не соответствует схеме CID URI).

Первым шагом процесса преобразования будет нахождение схемы CID. Эта схема выделяется из URI путем добавления в конце суффикса '.uri.arpa.' и поиска NAPTR для имени 'cid.uri.arpa.'. Результат может иметь вид:

```
cid.uri.arpa.
;;      order pref flags service      regexp      replacement
IN NAPTR 100 10 "" "" "!\^cid:.\+@([\^\.]+\.) (.*)$!\2!i" .
```

Поскольку здесь имеется лишь одна запись, упорядочение не представляет проблемы. Поле replacement пусто, поэтому используется шаблон, указанный в поле regexp. Это регулярное выражение применяется ко всему значению URI для проверки соответствия и дает позитивный результат. Компонента \2 выражения для замены возвращает строку "example.com". Поскольку поле флагов пусто, поиск не является завершающим и далее следует запрос к DNS для получения новых записей NAPTR, связанных с доменом 'example.com'.

Отметим, что правило не выделяет полное доменное имя из CID, предполагая, что CID происходит от хоста и выделяя доменное имя этого хоста. Хотя все хосты (такие, как bar) могут иметь свои собственные записи NAPTR, поддержка независимых записей для всех хостов сайта может оказаться слишком обременительной. Шаблоны здесь не подходят, поскольку их использование будет возвращать результат лишь при отсутствии точного соответствия имен в системе.

Записи, возвращенные по запросу для имени example.com, могут иметь вид:

```
example.com.
;;      order pref flags service      regexp      replacement
IN NAPTR 100 50 "s" "z3950+I2L+I2C" "" z3950.tcp.example.com.
IN NAPTR 100 50 "s" "rescap+I2C" "" rescap.udp.example.com.
IN NAPTR 100 50 "s" "thttp+I2L+I2C+I2R" "" thttp.tcp.example.com.
```

Продолжая рассмотрение этого примера, отметим, значения поля order совпадают для всех записей и клиент может выбрать любую запись. Приложение определяет флаг 's' как признак завершения поиска и указания на то, что результатом перезаписи будет доменное имя, для которого следует запросить запись SRV. Когда клиент сделает такой запрос, он получит информацию о хосте, номере порта, протоколе и службах, доступных по этому протоколу. Этой информации клиенту достаточно для контакта с сервером и запроса у него информации о CID URI.

Напомним, что в регулярном выражении последовательность \2 служит для выделения доменного имени из CID, а \. для соответствия символам точки, разделяющим компоненты доменного имени. Поскольку \. используется в качестве escape-символа, вхождение этого символа как литерала должно использовать дополнительный escape-символ \. Для приведенной выше записи cid.uri.arpa регулярное выражение в master-файле должно иметь форму "!\^cid:.\+@([\^\.]+\.) (.\*)\$!\2!i". Когда клиентская программа получает запись, шаблон преобразуется в "!\^cid:.\+@([\^\.]+\.) (.\*)\$!\2!i".

## 5.3 Преобразование для схемы HTTP URI

Даже при широком развертывании систем URN сохранится значительное число URI, связанных с хостами. Следует обеспечить возможность разработки систем преобразования URI, способных независимо определять местоположение для таких URI.

Предположим, что имеется URI для весьма популярной программы, которую желательно опубликовать на многочисленных зеркалах по всему миру:

```
http://www.example.com/software/latest-beta.exe
```

Мы выделяем префикс "http" и ищем записи NAPTR для 'http.uri.arpa.'. Результат может иметь вид:

```
http.uri.arpa. IN NAPTR
;;      order pref flags service      regexp      replacement
100 90 "" "" "!\^http://([\^/:;]+)!1!i" .
```

Это выражение возвращает все символы, расположенные между двойной дробной чертой (//) и следующей дробной чертой или двоеточием. Для разграничения компонент выражения для замены используется восклицательный знак '!', поскольку в противном случае пришлось бы использовать escape-символы \ перед символами / и регулярное выражение для зоны имело бы вид:

```
"!\^http://\//([\^\//:;]+)\//1/i"
```

Применим шаблон к URI для извлечения www.example.com и найдем для этого имени записи NAPTR:

```
www.example.com.
```

<sup>3</sup>В оригинале ошибочно дается ссылка на RFC 3404. Прим. перев.

```
;;      order pref flags  service  regexp  replacement
IN NAPTR 100 100 "s" "thttp+L2R" "" thttp.example.com.
IN NAPTR 100 100 "s" "ftp+L2R" "" ftp.example.com.
```

Найдем записи SRV для thttp.example.com, которые будут содержать информацию о хостах, которые домен example.com указал в качестве зеркал. Клиентская программа может указать пользователю один из таких хостов.

## 6. Примечания

- Процедуры регистрации для зон DNS 'urn.arpa.' и 'uri.arpa.' описаны в документе "Dynamic Delegation Discovery System (DDDS) Part Five: URI.ARPA Assignment Procedures" (RFC 3405) [5].
- Если запись с определенным значением поля order соответствует URI, но клиенту не известен указанный в записи протокол и сервис, клиенту **следует** продолжить просмотр записей с таким же значением поля order. Для клиента в этом случае **недопустимо** рассматривать записи с большими значениями поля order. Это необходимо для обеспечения работоспособности при частичной передаче полномочий для зоны. Поле order позволяет администраторам сказать: "Все запросы для URI, соответствующие шаблону x, ведут к серверу 1, а все прочие – к серверу 2".
- Отметим, что записи SRV RR предъявляют к клиентам дополнительные требования.

## 7. Согласование с IANA

Использование зон "urn.arpa." и "uri.arpa." требует разработки политики и процедур регистрации, а также операций по поддержке этих зон DNS. Эта политика и процедуры описаны в документе Dynamic Delegation Discovery System (DDDS) Part Five: URI.ARPA Assignment Procedures (RFC 3405) [5]. Обеспечение работы и администрирование этих зон накладывает ответственность на агентство IANA.

Для регистрации значений полей Services и Flags требуется одобрение IESG и публикация RFC со статусом Informational или Standards track.

Правила регистрации для URI описаны в документе RFC 2717 [17], а правила регистрации URN NID – в RFC 2611 [16].

## 8. Вопросы безопасности

Использование "urn.arpa." и "uri.arpa." в качестве реестров для пространства имен может быть объектом DoS-атак и других атак DNS spoofing. В настоящее время изучается вопрос взаимодействия с DNSSEC. Предполагается, что записи NAPTR будут подписываться ключом SIG при развертывании системы DNSSEC.

Правила перезаписи делают идентификаторы из других пространств имен объектами тех же атак, которым подвержены доменные имена. Поскольку такие атаки не удавалось легко реализовать раньше их можно рассматривать или не рассматривать как проблему.

Корректность регулярных выражений требует более серьезной проверки, нежели простая передача во что-то типа PERL.

В этом документе обсуждается способ нахождения трансляторов, но не рассматриваются какие-либо детали взаимодействия с найденным транслятором. При взаимодействии с преобразователями может возникнуть множество вопросов, связанных с безопасностью, но рассмотрение этих вопросов выходит за пределы данной спецификации и должно включаться в спецификации соответствующих протоколов взаимодействия с трансляторами.

## 9. Благодарности

Редакторы выражают свою благодарность Keith Mooge за консультации в процессе подготовки этого документа. Мы благодарим также Paul Vixie за помощь при отладке реализации и ответы на наши вопросы. В заключение мы хотим поблагодарить участников конференции в Knoxville, а также членов рабочих групп URI и URN.

Специально хотим отметить Ron Daniel, который был соавтором исходного варианта этого документа. Его реализации и ясность мыслей внесли неоценимый вклад в прояснение множества вопросов.

## Литература

- [1] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part One: The Comprehensive DDDS", RFC 3401<sup>1</sup>, October 2002.
- [2] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Two: The Algorithm", RFC 3402<sup>1</sup>, October 2002.
- [3] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403<sup>1</sup>, October 2002.
- [4] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Four: The Uniform Resource Identifiers (URI) Resolution Application", RFC 3404, October 2002.
- [5] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Five: URI.ARPA Assignment Procedures", RFC 3405<sup>1</sup>, October 2002.
- [6] Sollins, K. and L. Masinter, "Functional Requirements for Uniform Resource Names", RFC 1737, December 1994.
- [7] Arms, B., "The URN Implementors, Uniform Resource Names: A Progress Report", D-Lib Magazine, February 1996.
- [8] Moats, R., "URN Syntax", RFC 2141, May 1997.
- [9] Gulbrandsen, A., Vixie, P. and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

<sup>1</sup>Перевод этого документа имеется на сайте <http://www.protocols.ru>. Прим. перев.

- [10] Daniel, R., "A Trivial Convention for using HTTP in URN Resolution", RFC 2169, June 1997.
- [11] Mealling, M., "URI Resolution Services Necessary for URN Resolution", RFC 2483, January 1999.
- [12] Moore, K., Browne, S., Cox, J. and J. Gettler, "Resource Cataloging and Distribution System", Technical Report CS-97-346, December 1996.
- [13] Sollins, K., "Architectural Principles of Uniform Resource Name Resolution", RFC 2276, January 1998.
- [14] Daniel, R. and M. Mealling, "Resolution of Uniform Resource Identifiers using the Domain Name System", RFC 2168, June 1997.
- [15] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.
- [16] Daigle, L., van Gulik, D., Iannella, R. and P. Falstrom, "URN Namespace Definition Mechanisms", RFC 2611, BCP 33, June 1999.
- [17] Petke, R. and I. King, "Registration Procedures for URL Scheme Names", RFC 2717, BCP 35, November 1999.
- [18] Mealling, M. and R. Daniel, "The Naming Authority Pointer (NAPTR) DNS Resource Record", RFC 2915, August 2000.

## Приложение А. Псевдокод

В помощь разработчикам ниже приводится псевдокод клиентской программы, использующей записи NAPTR. Этот код не задает стандартного способа обработки записей NAPTR. Кроме того, как всякий псевдокод, он не может быть выполнен непосредственно и не исключена возможность наличия в нем логических ошибок. Мы вас предупредили.

```
//
// findResolver(URN)
// По данному значению URN найдем хост, который может преобразовать это значение.
//
findResolver(string URN) {
    // добавляем префикс к ".urn.arpa."
    sprintf(key, "%s.urn.arpa.", extractNS(URN));
    do {
        rewrite_flag = false;
        terminal = false;
        if (key has been seen) {
            завершаем цикл при обнаружении ошибки
        }
        добавляем ключ key к списку значений seen
        records = lookup(type=NAPTR, key); // получаем все записи NAPTR RR для ключа key

        отбрасываем все записи с неизвестным значением поля flags.
        сортируем записи NAPTR по значениям полей order preference
            (сначала по order, а потом по preference).
        n_naptrs = число записей NAPTR в отклике.
        curr_order = records[0].order;
        max_order = records[n_naptrs-1].order;

        // обработка текущего пакета записей NAPTR в соответствии со значением поля order.
        for (j=0; j < n_naptrs && records[j].order <= max_order; j++) {
            if (unknown_flag) // пропускаем запись и переходим к следующей
                continue;
            newkey = rewrite(URN, naptr[j].replacement, naptr[j].regex);
            if (!newkey) // переходим к следующей записи если перезаписи не происходит
                match continue;
            // мы не делаем перезаписи и усекаем max_order до текущего значения,
            // следовательно передача полномочий работает корректно
            max_order = naptr[j].order;
            // Если неизвестно, что делать с протоколом и службами
            // из записи NAPTR, пытаемся работать со следующей записью.
            if(!isKnownProto(naptr[j].services)) {
                continue;
            }
            if(!isKnownService(naptr[j].services)) {
                continue;
            }

            // К этому моменту перезапись успешно завершена и мы знаем, как сделать запрос к
            // известной службе преобразования. Перед следующим просмотром проверяются флаги
            // для определения дальнейших действий.
            // Примечание: эту часть можно переписать так, чтобы корректная запись помечалась
            // и процесс продолжался с целью поиска "лучшей" записи.
            // Однако этот код достаточно велик, а наше приложение является лишь иллюстрацией.
            if (strcasemp(flags, "S") || strcasemp(flags, "P") || strcasemp(flags, "A")) {
                terminal = true;
                services = naptr[j].services;
                addnl = все записи SRV и/или A, возвращенные как дополнение к naptr[j].
            }
            key = newkey;
        }
    }
}
```

```
rewriteflag = true;
break;
}
} while (rewriteflag && !terminal);

// Путь к преобразователю найден?
if (!rewrite_flag) {
    сообщить об ошибке;
    return NULL;
}

// Оставить дальнейшее другому протоколу?
if (strcasestr(flags, "P")) {
    вернуть ключ key, как хост для обращения;
}

// если нет, продолжаем
if (!addnl) { // В дополнительной информации нет записей SRV, ищем их
    srvs = lookup(type=SRV, key);
}

сортируем записи SRV по полям preference, weight, ...
for each (SRV record) { // в порядке значений preference
    пытаемся соединиться с srv[j].target, используя протокол и одну из служб
преобразования,
    в поле services последней записи NAPTR.
    if (successful)
        return (target, protocol, service);
    // Возможно, мы будем в реальности возвращать результат, но этот код предполагает,
    // что просто найден хороший хост для соединения с ним.
}
завершение с ошибкой "не удалось найти хост";
}
```

## Адрес автора

Michael Mealling

VeriSign

21345 Ridgetop Circle

Sterling, VA 20166

US

EMail: [michael@neonym.net](mailto:michael@neonym.net)

URI: <http://www.verisignlabs.com>

## Перевод на русский язык

Николай Малых

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

## Полное заявление авторских прав

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.