

Network Working Group
Request for Comments: 1982
Updates: 1034, 1035
Category: Standards Track

R. Elz
University of Melbourne
R. Bush
RGnet, Inc.
August 1996

Арифметика порядковых номеров

Serial Number Arithmetic

Статус документа

Документ содержит спецификацию, предложенную в качестве стандарта для сообщества Internet, и служит приглашением к дискуссии в целях развития. Состояние стандартизации для документа вы можете проверить в Internet Official Protocol Standards (STD 1). Документ может распространяться свободно.

Аннотация

В этом документе рассматривается арифметика порядковых номеров, используемых в системе доменных имен (DNS¹). Система DNS в течение долгих лет использовала арифметику порядковых номеров, концепция которой не была определена в документах IETF, хотя эта арифметика достаточно понятна и широко используется. В данном документе дается недостающее определение. Документ служит обновлением для RFC1034 и RFC1035.

1. Введение

Поле порядкового номера в записи SOA определено в RFC1035 следующим образом

SERIAL – 32-битовое целое число без знака, задающее значение номера версии для оригинала зоны. При переносе зоны это значение сохраняется. Диапазон значений порядковых номеров является «циклическим²» и при сравнении значений следует использовать арифметику порядковых номеров.

RFC1034 использует такую же терминологию при определении процедур согласования для вторичного сервера зоны.

К сожалению термин «арифметика порядковых номеров» не определен ни в RFC1034 и RFC1035, ни в других упомянутых там документах.

Представляется, что в этой фразе имеется в виду арифметика, используемая для порядковых номеров TCP [RFC793], которая определена в документе [IEN-74].

К сожалению, арифметика, определенная [IEN-74], не соответствует требованиям DNS по причине отсутствия в этой арифметике операции сравнения.

Во избежание возникновения проблем в столь простой ситуации данный документ определяет операции, поддерживаемые для порядковых номеров. Это определение предназначено лишь для того, чтобы прояснить смысл RFC1034 и RFC1035 в надежде, что оно будет совместимо с существующими реализациями. Однако известно, что старые реализации трактуют порядковый номер как обычное целое число без знака без попыток реализовать тот или иной вариант «пространства порядковых номеров». В таких реализациях могут возникать проблемы при достижении максимального значения порядкового номера. Однако с этим уже ничего нельзя сделать, пока такие реализации не выйдут из употребления.

2. Арифметика порядковых номеров

Порядковые номера представляют собой конечное множество неотрицательных целых чисел, являющееся частью всего множества целых значений. Минимальное значение для каждого такого подмножества равно нулю, а максимальное на 1 меньше целой степени числа 2.

При рассмотрении порядковых номеров понятия минимального и максимального значения теряют смысл. Нет минимального номера и нет максимального, зато есть предыдущие и следующие номера.

Для такого использования порядковых номеров нужно задать размер пространства порядковых номеров. Этот размер, обозначаемый термином SERIAL_BITS, задается значением целочисленной степени числа 2, которое на 1 больше самого большого числа в пространстве порядковых номеров. Этот же параметр определяет число битов, которые нужны для представления всех возможных значений порядкового номера определенного типа. Операции над порядковыми номерами определены ниже.

3. Операции с порядковыми номерами

Для порядковых номеров определены лишь две операции – добавление положительного целого числа из ограниченного диапазона и сравнение двух порядковых номеров.

3.1. Добавление

Порядковые номера увеличиваются путем добавления положительного целого числа n , где значение n берется из диапазона $[0 .. (2^{SERIAL_BITS} - 1) - 1]$. Для порядкового номера s результат добавления s' определяется выражением

$$s' = (s + n) \text{ modulo } (2 \wedge SERIAL_BITS)$$

¹Domain Name System.

²В оригинале используется термин wrap.

где операции сложения и получения остатка от деления на модуль обеспечивают для нового порядкового номера значение в пределах определенного пространства порядковых номеров.

Добавление к порядковому номеру значений, не входящих в диапазон $[0 .. (2^{(\text{SERIAL_BITS} - 1)} - 1)]$, не определено.

3.2. Сравнение

Любые два порядковых номера s_1 и s_2 можно сравнить один с другим. Определение результата сравнения приводится ниже.

Рассмотрим сначала два целых числа i_1 и i_2 из неограниченного набора неотрицательных чисел, значения которых совпадают с s_1 и s_2 , соответственно. Сравним теперь числа i_1 и i_2 , используя обычную арифметику целых чисел.

Порядковые номера s_1 и s_2 равны между собой тогда и только тогда, когда $i_1 = i_2$. Во всех остальных случаях s_1 не равен s_2 .

Номер s_1 меньше номера s_2 тогда и только тогда, когда s_1 не равен s_2 и выполняется любое из приведенных ниже условий

$$(i_1 < i_2 \text{ and } i_2 - i_1 < 2^{(\text{SERIAL_BITS} - 1)})$$

или

$$(i_1 > i_2 \text{ and } i_1 - i_2 > 2^{(\text{SERIAL_BITS} - 1)})$$

Номер s_1 больше номера s_2 тогда и только тогда, когда s_1 не равен s_2 и выполняется любое из приведенных ниже условий

$$(i_1 < i_2 \text{ and } i_2 - i_1 > 2^{(\text{SERIAL_BITS} - 1)})$$

или

$$(i_1 > i_2 \text{ and } i_1 - i_2 < 2^{(\text{SERIAL_BITS} - 1)})$$

Отметим, что существуют пары номеров s_1 и s_2 , не равных друг другу, для которых приведенные выше правила не позволяют сказать, что s_1 больше или меньше, чем s_2 . Попытки использования операторов сравнения для таких пар приводят к неопределенному результату.

Причина неопределенности заключается в том, что любое простое определение операции сравнения, в соответствии с которым для такой пары (s_1 , s_2) можно сказать, что s_1 меньше s_2 , будет давать и обратный результат (s_2 меньше, чем s_1) при сравнении для пары (s_2 , s_1). Это означает, что выбор порядка операндов при сравнении может изменять результат, что и приводит к неопределенности

Можно определить операцию сравнения, не дающую неопределенных результатов для любой пары чисел, но такое определение приведет к неоправданному усложнению реализации и будет сложным для понимания. Кроме того, в результате подобного определения станут возможны случаи, когда

$$s_1 < s_2$$

но при этом

$$(s_1 + 1) > (s_2 + 1)$$

что вызывает интуитивное неприятие.

Реализации, в случае возникновения неопределенности, могут выбрать результат по своему усмотрению или возвращать флаг ошибки, а пользователям следует быть осторожными, чтобы не совершить в такой ситуации ошибку. В качестве меры предосторожности можно рекомендовать недопущение случаев возникновения таких пар.

Отношения «больше или равно» и «меньше или равно» логически вытекают из приведенных выше определений.

4. Следствия

Нужно отметить некоторые следствия из приведенных выше определений.

4.1. Следствие 1

Для любого порядкового номера s и любого целого числа n выполняется отношение

$$(s + n) \geq s.$$

Более того, равенство $(s + n) == s$ выполняется только в том случае, когда $n == 0$, а во всех остальных случаях $(s + n) > s$.

4.2. Следствие 2

Если s' является результатом прибавления ненулевого целого числа n к порядковому номеру s , а m – другое целое число из диапазона значений, которые можно прибавлять к порядковым номерам и s'' является результатом добавления m к порядковому номеру s' , то невозможно однозначно утверждать, что s'' больше (или меньше), чем s , хотя известно, что s'' не равен s .

4.3. Следствие 3

Если продолжит увеличение номера s'' из предыдущего следствия, нельзя будет предсказать результат сравнения результата с номером s .

4.4. Следствие 4

Если в следствии 2 добавление суммы $(n + m)$ к порядковому номеру s не будет создавать неопределенного результата, тогда на основании следствия 1 можно будет сказать, что s'' больше, чем s .

5. Примеры

5.1. Тривиальный случай

Простейший случай пространства порядковых номеров имеет $SERIAL_BITS == 2$. В этом случае пространство номеров будет включать значения 0, 1, 2 и 3 ($3 == 2^{SERIAL_BITS} - 1$).

В этом случае, максимальное число, которое можно добавлять к порядковому номеру, составляет $2^{(SERIAL_BITS - 1)} - 1$ или просто 1.

Тогда $0+1 == 1$, $1+1 == 2$, $2+1 == 3$, а $3+1 == 0$. Далее можно сказать, что $1 > 0$, $2 > 1$, $3 > 2$, а $0 > 3$. Не существует возможности однозначного сравнения номеров для пар (2, 0) и (3, 1).

5.2. Более сложный пример

Рассмотрим более сложный случай с $SERIAL_BITS == 8$. В этом случае пространство порядковых номеров будет включать целые числа 0, 1, 2, ..., 254, 255 ($255 == 2^{SERIAL_BITS} - 1$).

Для этого случая наибольшее значение, которое можно добавлять к порядковому номеру, составляет $2^{(SERIAL_BITS - 1)} - 1$ или 127.

Операции добавления дают вполне предсказуемые результаты – $255 + 1 == 0$, $100 + 100 == 200$, а $200 + 100 == 44$.

Операция сравнения более интересна – $1 > 0$, $44 > 0$, $100 > 0$, $100 > 44$, $200 > 100$, $255 > 200$, $0 > 255$, $100 > 255$, $0 > 200$, а $44 > 200$.

Отметим, что $100+100 > 100$, но $(100+100)+100 < 100$. Увеличение порядкового номера может привести к тому, что номер станет "меньше". Естественно, что при добавлении к номеру небольших значений, подобные ситуации будут происходить реже. Однако нужно помнить об этой особенности, чтобы избежать возможных ошибок или использовать ее для реального уменьшения порядкового номера.

Пары значения (0, 128), (1, 129), (2, 130) и т. д. до (127, 255) дают при сравнении неопределенный результат.

Можно (произвольно) определить, что $128 > 0$, $129 > 1$, $130 > 2$, ..., $255 > 127$, несколько изменив определение оператора сравнения, приведенное выше. Однако следует отметить, что такое изменение приведет к тому, что $255 > 127$, а $(255 + 1) < (127 + 1)$, поскольку $0 < 128$. Кроме того, такое определение усложняет реализацию.

6. Использование арифметики

Поскольку приведенное здесь определение арифметики порядковых номеров может оказаться полезным не только для DNS, на эту арифметику можно ссылаться как на «арифметику порядковых номеров RFC1982». Любые ссылки на использование этой арифметики должны принимать во внимание, что операции с порядковыми номерами должны соответствовать правилам, приведенным в параграфах 2 - 5.

7. Порядковый номер DNS SOA

Порядковый номер записей DNS SOA представляет собой номер, соответствующий приведенному выше определению с $SERIAL_BITS = 32$. таким образом, этот порядковый номер представляет собой неотрицательное целое число из диапазона [0 .. 4294967295] – 32-битовое целое число без знака.

Максимальное значение инкремента порядкового номера составляет $2^{31} - 1$.

Следует обратить внимание на ситуации, когда порядковый записи не обновляется достаточно долго (максимальный срок ограничен значением SOA.expire) и инкремент при следующем обновлении может превысить указанное выше значение. В этом случае некоторые вторичные серверы имен с устаревшими данными будут иметь порядковый номер "больше" обновленного значения для первичного сервера. В некоторых случаях может потребоваться даже уменьшение порядкового номера записи. Если это приходится делать, следует принять специальные меры для проверки того, что **все** вторичные серверы корректно восприняли порядковые номера записи основного сервера при ее последующих обновлениях.

Отметим, что во всех случаях увеличение порядкового номера должно трактоваться как начало новой последовательности увеличения номера, а также как продолжение всех предыдущих последовательностей увеличения номера, начатых в период, указанный SOA.expire.

Следует также обратить внимание на установку для порядкового номера нулевого значения. Хотя нуль не имеет особого значения для арифметики порядковых номеров и порядковых номерах записей DNS SOA, многие реализации DNS некорректно трактуют нулевое значение номера как специальный случай и от них можно ждать неожиданностей при установке нулевого значения порядкового номера записи DNS SOA.

8. Обновление документов

Использование арифметики порядковых номеров (sequence space arithmetic) RFC1034 и RFC1035 должно трактоваться в соответствии с определениями, приведенными в данном документе.

9. Вопросы безопасности

В этом документе не рассматриваются вопросы безопасности.

Не делается предположений, что та или иная часть данного документа вносит дополнительный вклад в проблемы безопасности, которые могут быть связаны с DNS, и не предпринималось каких-либо попыток изучения этого вопроса.

Литература

[RFC1034] Domain Names - Concepts and Facilities, P. Mockapetris, [STD 13](#), ISI, November 1987.

[RFC1035] Domain Names - Implementation and Specification, P. Mockapetris, [STD 13](#), ISI, November 1987

[RFC793] Transmission Control protocol, Information Sciences Institute, [STD 7](#), USC, September 1981

[IEN-74] Sequence Number Arithmetic, William W. Plummer, BB&N Inc, September 1978

Благодарности

Спасибо Робу Аустейну (Rob Austein) за предложения по разъяснению неопределенностей в операции сравнения и Михаэлю Паттону (Michael Patton) за его попытки найти другое решение для операции сравнения. Благодарим также членов рабочей группы IETF DNSIND 1995-6, в частности, Пола Мокпетриса (Paul Mockapetris).

Адреса авторов

Robert Elz

Computer Science

University of Melbourne

Parkville, Vic, 3052

Australia.

E-Mail: kre@munnari.OZ.AU

Randy Bush

RGnet, Inc.

10361 NE Sasquatch Lane

Bainbridge Island, Washington, 98110

United States.

E-Mail: randy@psg.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru