

Network Working Group  
Request for Comments: 3540  
Category: Experimental

N. Spring  
D. Wetherall  
D. Ely  
University of Washington  
June 2003

## Устойчивый механизм сигнализации насыщения с помощью ECN-nonce

Robust Explicit Congestion Notification (ECN)  
Signaling with Nonces

### Статус документа

Этот документ определяет экспериментальный протокол (Experimental Protocol) для сообщества Internet. Документ не содержит спецификаций каких-либо стандартов и служит приглашением к дискуссии в целях развития протокола. Допускается свободное распространение документа.

### Авторские права

Copyright (C) The Internet Society (2003). All Rights Reserved.

### Тезисы

В это документе описано необязательное расширение ECN-nonce, обеспечивающее для ECN<sup>1</sup> защиту от случайного или преднамеренного сокрытия пакетов, помеченных отправителем TCP. Расширение повышает устойчивость работы систем контроля насыщения, предотвращая возможность использования ECN для неправомерного захвата полосы сетевых соединений. ECN-nonce использует два кода ECT<sup>2</sup> в поле ECN заголовков IP и требует наличия флага в заголовке TCP. Расширение эффективно работает как на маршрутизаторах, так и на хостах.

## 1. Введение

### Назначение

Данная спецификация описывает необязательное расширение ECN [RFC3168], повышающее устойчивость к случайному или преднамеренному сокрытию помеченных ECN пакетов. Это расширение не будет развертываться повсеместно. Одной из целей публикации данного RFC со статусом Experimental является осторожное развертывание и использование протокола до начала его стандартизации. Другой целью публикации является обеспечение разработчикам межсетевых экранов времени для признания и восприятия модели, представленной с помощью nonce<sup>3</sup>. Рабочая группа Transport Area заново представит эту спецификацию в качестве IETF Proposed Standard (проект стандарта) в будущем после обретения необходимого опыта.

Для корректной работы ECN нужна кооперация получателя (для передачи отправителю сигнала о насыщении - CE<sup>4</sup>) и отправителя, однако протокол не обеспечивает механизма для такой кооперации. Это ведет к тому, что неаккуратно или некорректно настроенные получатели всегда будут сбрасывать ECN-Echo и просто не будут сообщать отправителю о возникновении перегрузки в сети. Такое поведение дает получателю преимущества в скорости по сравнению с корректно настроенными соединениями. Более того, любое устройство на пути (транслятор адресов NAT, межсетевой экран, формирователь полосы QoS и т. п.) может безнаказанно удалить индикаторы перегрузки.

Такое поведение может создавать угрозу работе системы контроля насыщения в Internet. С учетом роли системы контроля насыщения разумно разработать сигнальный механизм ECN, который будет обеспечивать устойчивость к максимально возможному числу угроз. ECN-nonce обеспечивает простой и эффективный механизм предотвращения недопустимого использования ECN.

ECN-nonce позволяет отправителю проверить корректность поведения получателя ECN и не вносит искажений, которые могли бы привести к сокрытию помеченных (или отброшенных) пакетов на сигнальном пути. Механизм ECN-nonce защищает как от ошибок в реализациях, так и от злонамеренного использования. Этот механизм:

- с высокой вероятностью детектирует некорректно работающих получателей и не оказывает влияния на добросовестных получателей;
- не меняет других аспектов использования ECN и не снижает преимуществ, обеспечиваемых ECN для корректно работающих получателей;
- практически не добавляет служебной информации (один флаг в заголовке TCP) и не требует значительных ресурсов для обработки;

<sup>1</sup>Explicit Congestion Notification – явное уведомление о насыщении.

<sup>2</sup>ECN-Capable Transport – поддерживающий ECN транспорт.

<sup>3</sup>Мне не удалось найти выражения в русском языке, подходящего для перевода этого термина в контексте данного документа. Трактовку английского слова nonce можно посмотреть на сайте Wikipedia – <http://en.wikipedia.org/wiki/Nonce>.  
*Прим. перев.*

<sup>4</sup>Congestion Experienced – перегрузка в сети.

- прост и по имеющимся данным не подвержен другим атакам.

Отметим также, что использование ECN-ponce дает два дополнительных преимущества, даже если использовать только маршрутизаторы drop-tail<sup>1</sup>. Во-первых, отбрасывание пакетов не может быть скрыто от отправителя. Во-вторых, он позволяет предотвратить излишне оптимистичные подтверждения [Savage], когда сегменты TCP подтверждаются еще до их получения. Эти преимущества также служат повышению устойчивости системы контроля насыщения к атакам. Детального рассмотрения преимуществ в данном документе не приводится.

Остальная часть документа содержит описание механизма ECN-ponce в виде обзора с последующим детальным описанием поведения отправителей и получателей.

Ключевые слова: **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с [RFC2119].

## 2. Обзор

Работа ECN-ponce строится на базе существующих сигнальных механизмов ECN-Echo и CWR<sup>2</sup>. Предполагается знакомство читателя с ECN [ECN]. Для простоты ECN-ponce описывается только для одного направления, хотя этот механизм работает параллельно в обоих направлениях.

Протокол ECN для TCP сохраняется неизменным за исключением добавления нового поля в заголовок TCP. В соответствии с [RFC3168] в поле ECN заголовка IP исходящих пакетов устанавливается код ECT(0) или ECT(1). Перегруженные маршрутизаторы изменяют значение этого поля на CE. Когда получатель TCP видит код CE, он устанавливает флаг ECE<sup>3</sup> в последующих подтверждениях, пока не получит флаг CWR. Этот флаг (CWR) устанавливается отправителем в новых данных, как реакция отправителя на перегрузку.

ECN-ponce позволяет получателю уведомить отправителя, что подтверждаемый сегмент отправителя был принят без маркеров перегрузки. Случайные однобитовые значения (ponce) помещаются в 2-битовый код ECT. Однобитовая сумма этих значений возвращается в виде флага в заголовке TCP, называемого битом NS<sup>4</sup>. Маркировка пакетов удаляет значения ponce в коде ECT, поскольку CE переписывает оба бита ECN в заголовках IP. Так как для расчета суммы требуется каждое значение ponce, корректная сумма предполагает получение непомянутого пакета. В результате не только предотвращается сокрытие маркировки получателями, но и возможность снятия маркеров на промежуточных маршрутизаторах без предсказания исходных значений ponce.

Отправитель может проверить возвращенную получателем сумму значений ponce, чтобы убедиться в том, что индикация перегрузки путем маркировки (или отбрасывания) пакетов не была скрыта. Поскольку сумма ponce выражается в виде одного бита, отправитель имеет 50% вероятность обнаружения получателей, скрывающих наличие индикатора перегрузки. Поскольку каждое подтверждение представляет собой независимую попытку такого определения, обман разоблачается очень быстро, если сигналы о насыщении повторялись.

В следующих параграфах приводится более детальное описание протокола ECN-ponce.

В каждом подтверждении содержится сумма значений ponce, которая представляет собой 1-битовый результат сложения (четность или исключающее ИЛИ) значений ponce для подтверждаемого диапазона байтов. Сумма используется по той причине, что не каждый пакет подтверждается индивидуально и гарантия доставки подтверждений не обеспечивается. Если не пользоваться суммой, можно будут просто возвращать значение ponce из пакета без маркировки для уведомления отправителя о том, что пакет был доставлен без маркировки. Однако в силу того, что для таких подтверждений доставка не гарантируется, отправитель не сможет отличить потерю подтверждения ACK от сокрытия маркированных пакетов. Использование суммы ponce не позволяет получателю скрывать получение пакетов с маркерами путем простого отказа от их подтверждения. Поскольку значения ponce и сумма этих значений являются однобитовыми, предсказание суммы не проще, чем предсказание отдельных значений. Расчет суммы значений ponce показан на рисунке 1.

Отправитель	Получатель
	исходная сумма = 1
-- 1:4 ECT(0)	--> NS = 1 + 0(1:4) = 1(:4)
<- ACK 4, NS=1	---
-- 4:8 ECT(1)	--> NS = 1(:4) + 1(4:8) = 0(:8)
<- ACK 8, NS=0	---
-- 8:12 ECT(1)	-> NS = 0(:8) + 1(8:12) = 1(:12)
<- ACK 12, NS=1	--
-- 12:16 ECT(1)	-> NS = 1(:12) + 1(12:16) = 0(:16)
<- ACK 16, NS=0	--

Рисунок 1: Расчет суммы значений ponce (NS) на приемной стороне.

Отправитель	Получатель
	исходная сумма = 1
-- 1:4 ECT(0)	-> NS = 1 + 0(1:4) = 1(:4)
<- ACK 4, NS=1	--
-- 4:8 ECT(1)	-> CE -> NS = 1(:4) + ?(4:8) = 1(:8)
<- ACK 8, ECE NS=1	--
-- 8:12 ECT(1), CWR	-> NS = 1(:8) + 1(8:12) = 0(:12)
<- ACK 12, NS=0	--
-- 12:16 ECT(1)	-> NS = 0(:12) + 1(12:16) = 1(:16)
<- ACK 16, NS=1	--

Рисунок 2: Расчет NS на приемной стороне при маркировке пакета (4:8). Получатель может рассчитать некорректное значение суммы в результате утери ponce при маркировке пакета.

После того, как в сети возникло насыщение и пакеты были помечены или потеряны, требуется ресинхронизация сумм ponce между отправителем и получателем. При маркировке пакетов значения ponce сбрасываются и сумма ponce на приемной стороне не будет совпадать с суммой на стороне отправителя. После потери значений ponce различие в суммах у отправителя и получателя будет сохраняться до следующей потери. Это означает, что можно ресинхронизировать отправителя и получателя после перегрузки путем установки на передающей стороне значения суммы, принятой от получателя. Поскольку индикацию перегрузки не требуется

<sup>1</sup>«Обрубание хвостов» - вариант управления очередями в маршрутизаторах. Прим. перев.

<sup>2</sup>Congestion Window Reduced – уменьшение размера окна насыщения.

<sup>3</sup>ECN-Echo.

<sup>4</sup>Nonce sum – сумма значений ponce.

делать чаще, чем один раз за период кругового обхода, отправитель прекращает проверку суммы при получении сигнала CWR и устанавливает для своей суммы поспе значение, принятое от получателя с подтверждением новых данных. Преимуществом такого подхода является то, что получатель явно не вовлечен в процесс ресинхронизации. Иллюстрация процесса показана на рисунке 2. Отметим, что сумма поспе, возвращенная в ACK 12 (NS=0), отличается от суммы в предыдущем примере (NS=1) и сохраняет отличие для ACK 16 на рисунке 2.

Нам нужно еще согласовать значения поспе, передаваемые в пакетах и подтверждающие диапазон принятых байтов. Байтовые границы для подтверждений не обязательно соответствуют границам при передаче, а при повторе пакеты могут передаваться с иными границами. Первый вопрос об установке поспе при подтверждении части сегмента рассматривается в параграфе 6.1. Второй вопрос о значении поспе, передаваемом при повторе мелкого сегмента в большом, имеет простой ответ - ECN отключается при повторе, поэтому поспе не передается. Поскольку повторы передачи связаны с фактами насыщения, проверка поспе отключается, пока не будет подтверждения CWR и перегрузка не прекратится.

В следующих параграфах подробно рассматривается поведение отправителей, маршрутизаторов и получателей, начиная с передачи данных отправителем. Далее рассматривается сигнальный цикл ECN и в заключение описан процесс обработки подтверждения отправителем.

### 3. Поведение отправителя (передача)

Поведением отправителя управляют CWR и ECN-Echo, как это делалось раньше. Кроме того, нужно включать значения поспе в передаваемые пакеты и проверять корректность сумм поспе в принятых подтверждениях. Данный параграф описывает процесс передачи.

Для включения однобитового значения поспе в каждый совместимый с ECN пакет IP отправитель использует два кода ECT - ECT(0) представляет поспе = 0, а ECT(1) - поспе = 1. Как и для ECN повтор передачи не совместим с использованием ECN, поэтому пакет не включает значения поспе.

Отправитель поддерживает отображение порядкового номера для окончания каждого пакета в сумму поспе (а не значение поспе, включенное в исходную передачу), ожидаемую в подтверждении, соответствующем этому порядковому номеру.

### 4. Поведение маршрутизатора

Маршрутизаторы ведут себя в соответствии с [RFC3168]. При маркировке пакетов для индикации перегрузки исходное значение поспе в ECT(0) или ECT(1) теряется. Ни получатель, ни любой другой узел не могут снять маркировку пакета без предсказания исходного значения поспе.

### 5. Поведение получателя (прием и передача)

Получатель ECN-поспе поддерживает сумму значений поспе, вычисляемую по мере доставки пакетов, и возвращает текущее значение суммы поспе в каждом подтверждении. В остальном поведение получателя не отличается от [RFC3168]. Возврат суммы поспе не требуется, но рекомендуется, поскольку отправителям разрешено прекращать передачу ECN-пакетов получателям, которые не поддерживают ECN-поспе.

По мере того, как пакеты удаляются из очереди доставленных с нарушением порядка пакетов для подтверждения, значения поспе восстанавливаются из заголовков IP. Полученное из заголовка значение поспе прибавляется к текущему значению суммы поспе, как подтверждение порядкового номера для недавнего пакета.

В случае маркировки пакетов одно или множество значений поспе будет неизвестно получателю. В этом случае отсутствующие значения поспе игнорируются при расчете суммы (или в расчет включается нулевое значение поспе) и устанавливается флаг ECN-Echo для индикации насыщения отправителю.

Возврат суммы поспе, соответствующей данному подтверждению, прост. Сумма передается, как однобитовый флаг NS<sup>1</sup> в заголовке TCP. Этот бит размещается рядом с битами CWR и ECN-Echo, занимая позицию 7 в байте 13 заголовка TCP, как показано на рисунке 3.

Рисунок 3: Новое определение байтов 13 и 14 заголовка TCP

Начальное значение суммы поспе равно 1 и это значение включается в пакеты SYN/ACK и ACK трехэтапного согласования TCP. Это позволяет удаленной точке определить факт поддержки поспе, но не является согласованием и получателю SYN/ACK не требуется проверять наличие флага NS для принятия решения об установке NS в последующем пакете ACK.

### 6. Поведение отправителя (прием)

Этот параграф дополняет описание поведения отправителя, рассматривая этап проверки полученных значений суммы поспе.

Сумма поспе проверяется при получении подтверждения доставки новых данных за исключением периодов восстановления после перегрузки, когда дополнительные сигналы ECN-Echo будут игнорироваться. Проверка заключается в сравнении корректной суммы поспе, хранящейся в буфере, со значением из подтверждения с учетом рассмотренных ниже корректировок.

Если флаг ECN-Echo не установлен, это свидетельствует о том, что получатель не принимал пакетов с маркерами и, следовательно, может рассчитать и вернуть корректную сумму поспе. Для сокрытия маркеров получатель должен угадать сумму значений поспе, которые не были получены, поскольку, по крайней мере, один пакет был промаркирован

<sup>1</sup>Nonce Sum

и значение `nonce` было удалено. Так как `nonce` может с равной вероятностью принимать значения 0 или 1, сумма этих значений также может с равной вероятностью быть 0 или 1. Иными словами, вероятность угадывания составляет 50%. Благодаря тому, что каждое новое подтверждение является независимой попыткой угадать значение, отправитель может обнаружить подмену после небольшого числа удачных обманов.

Если флаг `ECN-Echo` установлен, это говорит о том, что получатель сигнализирует о перегрузки сети и сумму `nonce` проверять не нужно. Окно насыщения будет уменьшено наполовину, в следующем передаваемом пакете данных будет установлен флаг `CWR` и флаг `ECN-Echo` будет сброшен после получения сигнала `CWR`, как описано в [RFC3168]. В течение этого процесса восстановления сумма `nonce` может быть некорректной, поскольку одно или несколько значений `nonce` не будут получены. Это не имеет значения на этапе восстановления, так как TCP активизирует механизмы контроля насыщения не более одного раза за период  $RTT^1$ , независимо от числа потерянных за это время пакетов.

## 6.1. Ресинхронизация после потери или маркировки

После восстановления требуется заново синхронизировать суммы `nonce` для отправителя и получателя, чтобы была возможность дальнейшей проверки подтверждений. Когда сумма на стороне получателя некорректна, эта некорректность будет сохраняться до прекращения потери пакетов. Это позволяет воспользоваться простым механизмом ресинхронизации, когда отправитель сбрасывает свою сумму `nonce` в то значение, которое получатель указал в подтверждении данных, полученных после снижения размера окна насыщения. При обработке явных сигналов о насыщении это будет первое подтверждение без флага `ECN-Echo` (подтверждение пакета, содержащего флаг `CWR`).

Отправитель	Получатель
	исходная сумма = 1
-- 1:4 ECT(0)	-> NS = 1 + 0(1:4) = 1(:4)
<- ACK 4, NS=1	--
-- 4:8 ECT(1)	-> LOST
-- 8:12 ECT(1)	-> расчет суммы nonce откладывается до восстановления порядка доставки
<- ACK 4, NS=0	--
-- 12:16 ECT(1)	-> расчет суммы nonce отложен
<- ACK 4, NS=0	--
-- 4:8 retransmit	-> NS = 1(:4) + ?(4:8) + 1(8:12) + 1(12:16) = 1(:16)
<- ACK 16, NS=1	--
-- 16:20 ECT(1) CWR	->
<- ACK 20, NS=0	-- NS = 1(:16) + 1(16:20) = 0(:20)

Рисунок 4: Расчет сумм `nonce` на стороне получателя в случае потери пакетов и ресинхронизация после потери. Сумма `nonce` не изменяется, пока не будет кумулятивного подтверждения.

На практике ресинхронизация может быть обеспечена за счет сохранения бита, который имеет значение 1, если хранящееся у отправителя ожидаемое значение суммы отличается от суммы, полученной в подтверждении `CWR` и 0 в остальных случаях. Этот бит смещения синхронизации может использоваться при сравнении ожидаемой суммы с принятым значением суммы `nonce`.

Отправителю следует игнорировать значения суммы `nonce`, возвращаемые в подтверждениях с флагом `ECN-echo`.

Когда подтверждение относится лишь к части сегмента (например, в результате ресегментации TCP на промежуточном узле вместо использования фрагментации пакетов IP), отправителю следует принять сумму `nonce`, ожидаемую для следующей границы сегмента. Иными словами, подтверждения, относящиеся лишь к части исходного сегмента, будут включать ожидаемую сумму `nonce` только тогда, когда сегмент подтвержден полностью.

Следует отметить, что при использовании `ECN` отправитель может по той или иной причине не указывать для некоторых пакетов поддержку `ECN`. Отправитель `ECN-nonce` должен после передачи таких пакетов выполнять ресинхронизацию путем передачи `CWR` с первым пакетом данных после не поддерживающих `ECN` пакетов. Отправитель теряет защиту для всех неподтвержденных пакетов, пока не произойдет ресинхронизация.

## 6.2. Поведение отправителя при получении некорректной суммы `Nonce`

Реакция отправителя на получение некорректной суммы `nonce` является вопросом политики. Эти действия не связаны с механизмом проверки корректности и не обязательно одинаковы для всех отправителей. Более того, проверка корректности полученных сумм `nonce` не является обязательной и может быть отключена.

Если получатель никогда не передает отличных от нуля сумм `nonce`, отправитель может предположить, что получатель не понимает `nonce` и ограничить скорость соединения, помещая его в очередь с низким приоритетом или прекращая установку `ECT` в передаваемых сегментах.

Если принятая сумма `nonce` была установлена в предыдущем подтверждении, отправитель может предположить, что устройство в сети конфликтует с корректной сигнализацией между поддерживающими `ECN-nonce` конечными точками. Минимальный отклик на получение некорректной суммы `nonce` совпадает с откликом на получение `ECE`. Однако для компенсации скрытых сигналов о насыщении отправитель может снизить размер окна насыщения для одного сегмента и прекратить установку `ECT` в исходящих сегментах. Некорректная сумма `nonce` является признаком некорректного поведения или наличия ошибок между парой поддерживающих `ECN-nonce` конечных точек.

### 6.2.1. Использование `ECN-nonce` для защиты от некорректного поведения

Механизм `ECN-nonce` может обеспечивать дополнительные средства повышения устойчивости, сверх проверки передачи сигналов о маркировке пакетов. Этот механизм также препятствует сокрытию фактов отбрасывания пакетов от их отправителя (поскольку значения `nonce` из таких пакетов теряются). Отбрасывание пакетов потенциально возможно в дефектных реализациях TCP, во время некоторых атак и даже при использовании гипотетического ускорителя TCP. Такой ускоритель может играть на том, что он способен выполнить процедуру `fast start` для быстрого резервирования полосы и повторить это для другого соединения или обеспечить гарантии доставки на прикладных уровнях. Если нужно обеспечить устойчивость к такому поведению, не следует отключать `ECN-nonce`. Вместо этого нужно уменьшить на 1 размер окна насыщения или, используя очередь с низким приоритетом, наказать дефектную реализацию, продолжая проверки.

<sup>1</sup>Время кругового обхода. Прим. перев.

ECN-nonce может также детектировать некорректное поведение Eifel [Eifel] – недавно предложенного механизма устранения неопределенности повтора передачи для повышения производительности TCP. Некорректность поведения получателя может выражаться в том, что заявляется только прием исходных передач<sup>1</sup>, для того, чтобы заставить отправителя отказаться от выполненных действий по предотвращению перегрузки. Поскольку повторы передаются без ECT (и, следовательно, без nonce), возврат корректной суммы nonce подтверждает, что были получены только исходные передачи.

## 7. Взаимодействие с другими протоколами

### 7.1. Path MTU Discovery

Как описано в RFC3168, при использовании ECN рекомендуется устанавливать флаг запрета фрагментирования DF. Получатели, которые принимают непомятые фрагменты, могут восстановить исходное значение nonce для сокрытия маркированных фрагментов. ECN-nonce не может обеспечить защиту от некорректного поведения получателей, скрывающих маркированные фрагменты, поэтому часть защитных возможностей теряется в ситуациях, когда запрещено использование Path MTU Discovery.

В ответ на малые значения MTU для пути отправитель будет повторять передачу в виде мелких кадров взамен одного большого. Поскольку эти более мелкие пакеты передаются повторно, они будут несовместимы с ECN и не будут включать nonce. Отправителю следует выполнять ресинхронизацию при первом заново<sup>2</sup> переданном пакете.

### 7.2. SACK

Селективные подтверждения позволяют получателю подтверждать доставку с нарушением порядка в целях оптимизации. Не требуется изменять опцию селективного подтверждения, чтобы использовать суммы nonce, вычисленные для диапазона подтверждаемых байтов, поскольку SACK не может использоваться получателем для сокрытия сигналов насыщения. Сумма nonce соответствует только данным, подтверждаемым кумулятивным пакетом ACK.

### 7.3. IPv6

Заголовок IPv4 защищен контрольной суммой, но этого нет в IPv6, что повышает вероятность пропуска битовых ошибок в заголовках IPv6. Битовые ошибки, нарушающие целостность полей уведомления о перегрузке, могут приводить к получению некорректных значений nonce и возврату некорректных сумм nonce.

## 8. Вопросы безопасности

Для создания случайных однобитовых значений nonce не требуется криптографического качества генерации случайных чисел. Повышение качества генерации случайных чисел будет вести к потере производительности. В силу сказанного, последовательности случайных значений nonce не следует использовать для каких-либо иных целей.

Псевдослучайную последовательность битов не следует генерировать путем линейного сдвига регистра [Schneier] или с использованием подобных схем, которые позволяют любому, кто увидел несколько предыдущих битов, предсказать функцию генерации и последующие результаты.

Хотя ECN-nonce защищает от сокрытия сигналов насыщения и чересчур оптимистичных подтверждений, этот механизм не обеспечивает дополнительной защиты целостности соединений.

## 9. Согласование с IANA

Флаг NS (Nonce Sum) передается в резервном бите заголовка TCP, который должен быть выделен для этих целей. Данный документ описывает использование бита 7, смежного другим битам заголовка, используемым для ECN.

Код для флага NS в заголовке TCP задается стандартизацией (Standards Action) данного RFC, как требует RFC 2780. Агентство IANA добавило следующее определение в реестр флагов "TCP Header Flags":

RFC 3540 определяет бит 7 из зарезервированного поля для использования в качестве Nonce Sum, следующим образом:

```

    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     | N | C | E | U | A | P | R | S | F | |
| Header Length | Reserved | S | W | C | R | C | S | S | Y | I |
|                                     | R | E | G | K | H | T | N | N |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Флаги заголовков TCP

Бит	Имя	Документ
7	NS (Nonce Sum)	[RFC 3540]

## 10. Заключение

ECN-nonce представляет собой простую модификацию сигнального механизма ECN, которая повышает устойчивость ECN за счет предотвращения сокрытия получателями маркированных (или отброшенных) пакетов. Назначение этой работы заключается в повышении уровня устойчивости системы контроля насыщения в Internet. Модификация сохраняет характер и простоту существующей сигнализации ECN. Она практична для развертывания в сети Internet. Механизм использует коды ECT(0) и ECT(1), а также один флаг в заголовке TCP (так же, как CWR и ECN-Echo) и имеет простые правила обработки.

<sup>1</sup>А не повторов. Прим. перев.

<sup>2</sup>Не повторно. Прим. перев.

## 11. Литература

- [ECN] "The ECN Web Page", URL "<http://www.icir.org/floyd/ecn.html>".
- [RFC3168] Ramakrishnan, K., Floyd, S. and D. Black, "The addition of explicit congestion notification (ECN) to IP", [RFC 3168](#), September 2001.
- [Eifel] R. Ludwig and R. Katz. The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions. Computer Communications Review, January, 2000.
- [B97] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), March 1997.
- [Savage] S. Savage, N. Cardwell, D. Wetherall, T. Anderson. TCP congestion control with a misbehaving receiver. SIGCOMM CCR, October 1999.
- [Schneier] Bruce Schneier. Applied Cryptography, 2nd ed., 1996

## 12. Благодарности

Этот документ основан на результатах исследований Stefan Savage, David Ely, David Wetherall, Tom Anderson и Neil Spring. Мы весьма признательны Sally Floyd за его отклики и помощь.

## 13. Адреса авторов

**Neil Spring**

E-Mail: [nspring@cs.washington.edu](mailto:nspring@cs.washington.edu)

**David Wetherall**

Department of Computer Science and Engineering, Box 352350

University of Washington

Seattle WA 98195-2350

E-Mail: [djw@cs.washington.edu](mailto:djw@cs.washington.edu)

**David Ely**

Computer Science and Engineering, 352350

University of Washington

Seattle, WA 98195-2350

E-Mail: [ely@cs.washington.edu](mailto:ely@cs.washington.edu)

### Перевод на русский язык

Николай Малых

[nmalykh@gmail.com](mailto:nmalykh@gmail.com)

## 14 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 обеспечивается Internet Society.