

Network Working Group
Request for Comments: 3443
Updates: 3032
Category: Standards Track

P. Agarwal
Brocade
B. Akyol
Cisco Systems
January 2003

Обработка TTL в сетях MPLS

Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks

Статус документа

В этом документе содержится спецификация протокола, предложенного сообществу Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации протокола вы можете узнать из документа Internet Official Protocol Standards (STD 1). Документ может распространяться без ограничений.

Авторские права

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

Тезисы

Этот документ описывает обработку полей TTL¹ в иерархических сетях MPLS². Основным мотивом послужила потребность формализовать прозрачных для TTL режим работы путей с коммутацией по меткам MPLS. Документ обновляет RFC 3032 «MPLS Label Stack Encoding». Описана обработка TTL в иерархических туннелях с режимами Pipe и Uniform с примерами для случаев push и pop. Документ также служит дополнением к RFC 3270 «MPLS Support of Differentiated Services» и объединяет введенную в этом документе терминологию с обработкой TTL в иерархических сетях MPLS.

Используемые соглашения

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 должны интерпретироваться в соответствии с [RFC-2119].

1. Введение и мотивы

Этот документ описывает обработку полей TTL в иерархических сетях MPLS. Мы надеемся, что этот документ детализирует вопросы, рассмотренные в [MPLS-ARCH, MPLS-ENCAPS], а представленные здесь методы дополняют [MPLS-DS].

В частности, вводится новый режим обработки (его называют Pipe Model) для поддержки практики настройки MPLS LSP так, что проходящие через LSP пакеты видят туннель, как один интервал пересылки (hop), независимо от числа промежуточных маршрутизаторов LSR³. Модель Pipe (труба) для TTL в настоящее время применяется во многих сетях и реализуется операторами сетей в форме настраиваемой опции для оборудования нескольких производителей.

Этот документ формализует обработку TTL в сетях MPLS и связывает ее с терминологией, введенной в [MPLS-DS].

2. Обработка TTL в сетях MPLS

2.1. Отличия от RFC 3032 [MPLS-ENCAPS]

- [MPLS-ENCAPS] охватывает только однородную модель (Uniform Model), **не** затрагивая модели Pipe и Short Pipe. Данный документ рассматривает обе эти модели и для полноты включает также модель Uniform.
- [MPLS-ENCAPS] не охватывает иерархические LSP, которые рассмотрены здесь.
- [MPLS-ENCAPS] не определяет обработку TTL в присутствии PHP⁴, что рассмотрено здесь.

2.2. Терминология и основы

Как определено в [MPLS-ENCAPS], пакеты MPLS используют заголовок-прокладку (shim) MPLS, который содержит перечисленную ниже информацию о пакете..

- Метка MPLS (20 битов).
- TTL (8 битов).
- Дно (Bottom) стека (1 бит).
- Экспериментальные биты (3 бита).

¹Time To Live — время жизни.

²Multi-Protocol Label Switching — многопротокольная коммутация по меткам.

³Label switch router — маршрутизатор с коммутацией по меткам.

⁴Penultimate Hop Popping - «выплывание» (выталкивание) на предпоследнем этапе.

Экспериментальные биты были переопределены в [MPLS-DS] для индикации поведения в плане планирования и формовки (shaping), связанных с пакетом MPLS.

В [MPLS-DS] определены две модели (режима) для туннелей MPLS - Pipe и Uniform. В режиме Pipe сеть MPLS действует подобно устройству (каналу) при прохождении пакетов MPLS через сеть и для узлов, расположенных вне туннеля видны только вход и выход LSP. Сокращенная вариация этого режима (Short Pipe) также определена в [MPLS-DS] с целью различения вариантов пересылки на выходе и трактовок качества обслуживания (QoS). С другой стороны, модель Uniform делает видимыми все узлы, через которые проходит LSP, для расположенных вне туннеля узлов. Здесь модели Pipe и Uniform расширяются за счет включения обработки TTL. Кроме того, в документе описана обработка TTL при выполнении PHP. Подробное описание моделей Pipe и Uniform приведено в [MPLS-DS].

Обработку TTL в сетях MPLS можно разбить на два логических блока: (i) определение TTL на входе для учета любого выхода из туннеля в результате операций MPLS Pop и (ii) обработка (возможно) раскрываемых пакетов и TTL на выходе.

Отметим также, что сигнализация типа LSP (Pipe, Short Pipe или Uniform) выходит за рамки документа и не рассматривается в текущих версиях протоколов распространения меток (например, LDP [MPLS-LDP] и RSVP-TE [MPLS-RSVP]). В настоящее время LSP настраиваются операторами вручную с помощью команд или интерфейса управления сетью.

2.3. Новые термины

iTTL

Значение TTL для использования в качестве TTL на входе. Проверки iTTL не выполняется.

oTTL

Это значение служит в качестве TTL на выходе TTL (исключения указаны в параграфе 3.5). Если явно не указано иное, это значение всегда составляет (iTTL - 1).

oTTL Check

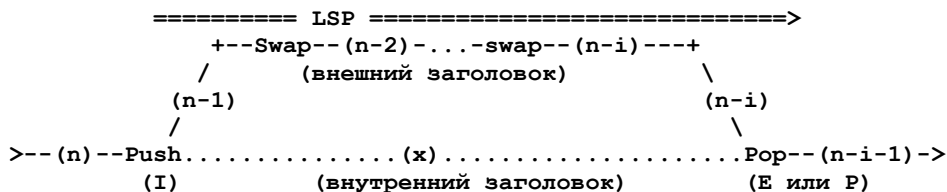
Проверка того, что oTTL > 0. Если эта проверка не проходит, пакет не будет пересылаться. Отметим, что проверка oTTL выполняется лишь в тех случаях, когда в качестве TTL на выходе (IP или MPLS) установлено значения oTTL (исключения указаны в параграфе 3.5).

3. Обработка TTL в разных моделях

В этом разделе описана обработка TTL для LSP, соответствующих каждой из 3 моделей (Uniform, Short Pipe, Pipe) при наличии и отсутствии PHP (когда это применимо).

3.1. Обработка TTL в Uniform LSP (с PHP и без него)

(согласуется с [MPLS-ENCAPS]):

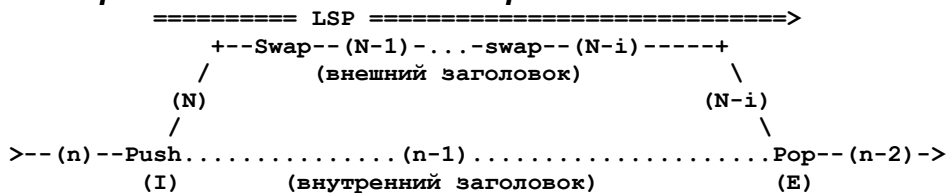


- (n) представляет значение TTL в соответствующем заголовке
- (x) представляет не имеющую значения информацию TTL
- (I) представляет входной узел LSP
- (P) представляет предпоследний узел LSP
- (E) представляет выходной узел LSP

Рисунок показывает обработку TTL для MPLS LSP в режиме Uniform. Отметим, что внутреннее и внешнее значения TTL в пакетах синхронизируются на входе и выходе туннеля.

3.2. Обработка TTL в Short Pipe LSP

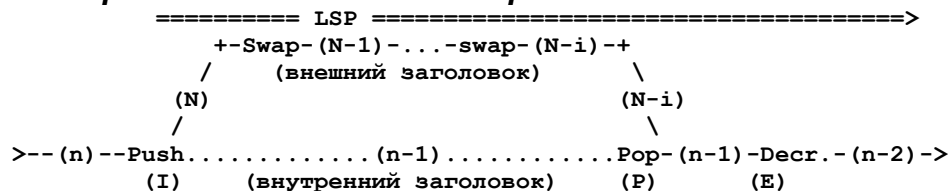
3.2.1. Обработка TTL для Short Pipe LSP без PHP



- (N) представляет значение TTL (может быть не связано с n)
- (n) представляет туннелированное значение TTL в инкапсулированном заголовке
- (I) представляет входной узел LSP
- (E) представляет выходной узел LSP

Модель Short Pipe была введена в [MPLS-DS]. В модели Short Pipe трактовка пересылки на выходном LSR базируется на туннелированном (а не инкапсулирующем) пакете.

3.2.2. Обработка TTL для Short Pipe LSP с PHP

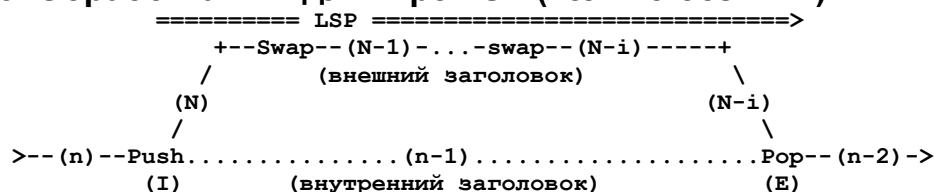


- (N) представляет значение TTL (может быть не связано с n)
- (n) представляет туннелированное значение TTL в инкапсулированном заголовке
- (I) представляет входной узел LSP
- (P) представляет предпоследний узел LSP
- (E) представляет выходной узел LSP

Поскольку метка уже была вытолкнута (popped) предпоследним узлом LSP, выходной узел LSP просто уменьшает TTL в заголовке.

Отметим также, что в конце LSP в режиме Short Pipe значение TTL в туннелируемом пакете уменьшается на 2 (независимо от PHP).

3.3. Обработка TTL для Pipe LSP (только без PHP)



- (N) представляет значение TTL (может быть не связано с n)
- (n) представляет туннелированное значение TTL в инкапсулированном заголовке
- (I) представляет входной узел LSP
- (E) представляет выходной узел LSP

В части TTL трактовка Pipe LSP идентична Short Pipe без PHP.

3.4. Определение TTL на входе (iTTL)

Если входящий пакет относится к протоколу IP, iTTL будет совпадать со значением TTL во входящем пакете IP.

Если входящий пакет относится к MPLS и выполняется процедура Push/Swap/PHP, iTTL будет совпадать со значением TTL верхней входящей метки.

Если входящий пакет относится к MPLS и выполняется процедура Pop (выход из туннеля), значение iTTL основывается на типе туннеля (Pipe или Uniform) в LSP, где происходит выталкивание. Если выталкиваемая метка относится к Pipe LSP, iTTL будет совпадать со значением поля TTL в заголовке, показываемом после выталкивания метки (отметим, что в контексте данного документа может показываться заголовок IP или метка MPLS). Если выталкиваемая метка относится к Uniform LSP, iTTL будет совпадать с TTL вытолкнутой метки. Если последовательно выполняется множество операций выталкивания (Pop), указанная выше процедура будет повторяться с одним исключением — значение iTTL, рассчитанное для предыдущего выталкивания, служит в качестве TTL последующей выталкиваемой метки (т. е. значение TTL, содержащееся в последующей метке, игнорируется и заменяется значением iTTL, полученным для предыдущего выталкивания).

3.5. Определение TTL на выходе и обработка пакетов

После завершения расчета iTTL выполняется проверка oTTL. При успешном завершении проверки oTTL рассчитывается выходное значение TTL пакетов (с метками и без меток) и заголовки пакетов обновляются, как описано ниже.

Если пакет маршрутизировался, как пакет IP, в качестве значения TTL в заголовке пакета IP устанавливается oTTL (iTTL - 1). Значения TTL для всех втолкнутых (pushed) меток определяются в соответствии с параграфом 3.6.

Для пакетов, маршрутизируемых как MPLS, имеется 4 варианта, перечисленных ниже.

- 1) Только обмен (Swap-only) — маршрутизируемая метка меняется местами (swapped) с другой меткой, а в поле TTL выходной метки устанавливается значение oTTL.
- 2) Обмен с последующим вталкиванием (Swap followed by a Push) — выполняется обмен метками как в п. 1). Значения TTL всех вталкиваемых меток определяются в соответствии с параграфом 3.6.
- 3) Выталкивание на предпоследнем этапе (PHP) — маршрутизируемая метка выталкивается (popped). Проверку oTTL следует проводить независимо от использования oTTL для обновления поля TTL в выходном заголовке. Если вытолкнутая (PHPed) метка относилась к Short Pipe LSP, поле TTL показанного PHP заголовка не проверяется и не обновляется. Если вытолкнутая метка относилась к Uniform LSP, в поле TTL показанного PHP заголовка устанавливается значение oTTL. Значения TTL дополнительных меток определяются в соответствии с параграфом 3.6
- 4) Выталкивание (Pop) — операция выталкивания происходит до маршрутизации и поэтому здесь не рассматривается.

3.6. Обработка на входе в туннель (Push)

Для каждой вталкиваемой (pushed) метки в режиме Uniform значение TTL копируется из метки/IP-пакета, расположенного непосредственно ниже.

Для каждой вталкиваемой метки в режиме Pipe или Short Pipe в поле TTL устанавливается значение, заданное оператором. Во многих реализациях по умолчанию установлено значение 255.

3.7. Замечания по реализации

- 1) Хотя iTTL при обновлении или определении oTTL может инкрементироваться больше, чем на 1, эту возможность следует использовать лишь для «компенсации» узлов, не способных уменьшать значения TTL.
- 2) При уменьшении iTTL должны быть приняты меры предотвращения отрицательных значений.
- 3) В режиме Short Pipe с включенным PHP значение TTL в туннелированных пакетах не изменяется после операции PHP.

4. Заключение

В этом документе описана возможная обработка полей TTL в сетях MPLS. Разъяснены разные методы, которые могут применяться при наличии иерархических туннелей, а также завершено объединение моделей Pipe и Uniform с обработкой TTL.

5. Вопросы безопасности

Этот документ не добавляет вопросов безопасности к отмеченным в [MPLS-ENCAPS, MPLS-DS]. В частности, документ не определяет новых протоколов или расширений существующих и не создает проблем безопасности для имеющихся протоколов. Авторы надеются, что прояснение обработки TTL в сетях MPLS обеспечит преимущества для сетевых операторов и их пользователей за счет упрощения поиска неполадок.

6. Литература

6.1. Нормативные документы

- [RFC-2119] Bradner, S. "Key words for use in RFC's to Indicate Requirement Levels", BCP 14, [RFC 2119](#), March 1997.
- [MPLS-ARCH] Rosen, E., Viswanathan, A. and R. Callon, "Multiprotocol Label Switching Architecture", [RFC 3031](#), January 2001.
- [MPLS-ENCAPS] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T. and A. Conta, "MPLS Label Stack Encoding", [RFC 3032](#), January 2001.
- [MPLS-DS] Le Faucheur, F., Wu, L., Davie, B., Davari, S., Vaananen, P., Krishnan, R., Cheval, P. and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, May 2002.

6.2. Дополнительная литература

- [MPLS-LDP] Andersson, L., Doolan, P., Feldman, N., Fredette, A. and B. Thomas, "LDP Specification", RFC 3036, January 2001.
- [MPLS-RSVP] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V. and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

7. Благодарности

Авторы благодарят членов рабочей группы MPLS за их отзывы. Отдельная благодарность Shahram Davari и Loa Andersson за внимательное рецензирование документа и комментарии.

8. Адреса авторов

Puneet Agarwal

Brocade Communications Systems, Inc.

1745 Technology Drive

San Jose, CA 95110

EMail: puneet@acm.org

Bora Akyol

Cisco Systems

170 W. Tasman Drive

San Jose, CA 95134

EMail: bora@cisco.com

Перевод на русский язык

Николай Малых

9. Полное заявление авторских прав

Copyright (C) The Internet Society (2003). 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.