Network Working Group
Request for Comments: 3828
Category: Standards Track

L-A. Larzon
Lulea University of Technology
M. Degermark
S. Pink
The University of Arizona
L-E. Jonsson, Ed.
Ericsson
G. Fairhurst, Ed.
University of Aberdeen

July 2004

Облегченный протокол пользовательских дейтаграмм (UDP-Lite)

The Lightweight User Datagram Protocol (UDP-Lite)

Статус документа

Данный документ содержит стандарт протокола Internet, предложенного сообществу Internet, и является приглашением к дискуссии в целях развития этого протокола. Сведения о текущем состоянии стандартизации протокола вы найдёте в документе Internet Official Protocol Standards (STD 1). Документ можно распространять без ограничений.

Авторские права

Аннотация

Данный документ описывает протокол UDP-Lite[1], подобный протоколу UDP[2] (RFC 768), но может также обслуживать в склонных к ошибкам сетевых средах приложения, которые предпочитают получать частично подтверждённые дейтаграммы, но не отбрасывать пакеты при любой ошибке. Если это свойство не использовать, UDP-Lite семантически идентичен протоколу UDP.

Оглавление

1. Введение

Этот документ описывает новый транспортный протокол UDP-Lite, известный также, как UDPLite. Основой нового протокола служат три наблюдения.

Во первых, существует класс приложений, которые получат преимущества, если повреждённые данные будут доставляться, а не отбрасываться. Множество кодеков для голоса и видео относятся к этому классу (например, кодек речи AMR [RFC 3267], кодек Internet Low Bit Rate Codec [ILBRC], устойчивые к ошибкам видеокодеки H.263+ [ITU-H.263], H.264 [ITU-H.264; H.264], MPEG-4 [ISO-14496]). Эти кодеки могут разрабатываться так, что ошибки в данных для них будут предпочтительней отбрасывания пакетов целиком.

Во вторых, всем каналам, поддерживающим передачу трафика IP, следует использовать строгую проверку целостности на канальном уровне (например, CRC-32 [RFC 3819]) и эта проверка **должна** по умолчанию использоваться для трафика IP. Когда нижележащий канальный уровень поддерживает такую проверку, некоторые типы трафика (например, UDP-Lite) могут получить преимущества в результате смены поведения канального

[1]Lightweight User Datagram Protocol - облегченный протокол пользовательских дейтаграмм.
[2]User Datagram Protocol - протокол пользовательских дейтаграмм.

уровня, позволяющей по запросу пересылку частично повреждённых пакетов IP [RFC 3819]. Некоторые радиотехнологии (например, [3GPP]) поддерживают такое поведение при работе при работе в точках, где стоимость и задержки достаточно малы. Если склонные к ошибкам каналы знают о чувствительной к ошибкам части пакета, для физического соединения можно обеспечить дополнительную защиту за счёт снижения вероятности повреждения чувствительных к ошибкам байтов (например, использовать неоднородную упреждающую коррекцию ошибок FEC[1]).

В третьих, промежуточным уровням (т. е., IP и протокол транспортного уровня) не следует запрещать работу устойчивых к ошибкам приложений при наличии таких каналов. Протокол IP не создаёт проблем в этом отношении, поскольку заголовок IP включает контрольную сумму, покрывающую все данные пакета IP. Из транспортных протоколов общего назначения лучше всего подходит UDP, поскольку он не создаёт издержек, связанных с повтором передачи ошибочных пакетов, нарушением порядка доставки и коррекцией ошибок. Для IPv4 [RFC 791] контрольная сумма UDP покрывает пакет целиком или просто не используется. Для IPv6 [RFC 2460] контрольная сумма UDP является обязательной и её использование не может быть отключено. Заголовок IPv6 не включает контрольной суммы самого заголовка и считается необходимым всегда защищать адресную информацию IP с помощью обязательной контрольной суммы UDP.

Требуется транспортный протокол, который соответствует свойствам канального уровня и перечисленным выше требованиям [LDP99]. Механизм детектирования ошибок транспортного уровня должен обеспечивать защиту важной информации (такой, как заголовки) вкупе с игнорированием ошибок, с которыми лучше иметь дело приложениям. Набор октетов, проверяемых с помощью контрольной суммы, лучше всего задавать со стороны передающего приложения.

Протокол UDP-Lite обеспечивает поддержку контрольных сумм с необязательным частичным покрытием. При использовании этой опции пакет делится на две части - чувствительную к ошибкам (покрывается контрольной суммой) и нечувствительную к ним (не покрывается контрольной суммой). Ошибки в нечувствительной к ним части не будут приводить к отбрасыванию пакетов транспортным уровнем принимающего конечного хоста. Когда контрольная сумма покрывает весь пакет (это следует делать по умолчанию), UDP-Lite семантически идентичен протоколу UDP.

По сравнению с UDP, неполные контрольные суммы UDP-Lite обеспечивают дополнительную гибкость для приложений, которые хотят определить часть своих данных в пакетах, как нечувствительную к ошибкам.

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].

3. Описание протокола

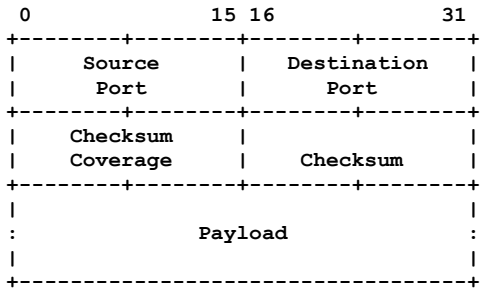

Рисунок 1. Формат заголовка UDP-Lite.

Заголовок пакетов UDP-Lite показан на рисунке 1. Его формат отличается от формата UDP - поле Length заменено полем Checksum Coverage. Эта замена допустима, поскольку информация о размере пакета UDP может быть обеспечена модулем IP так же, как это делается для TCP [RFC 793].

3.1. Поля

Поля Source Port и Destination Port определяются, как в спецификации UDP [RFC 768]. Протокол UDP-Lite использует тот же набор портов, который выделен агентством IANA для использования с UDP.

Поле Checksum Coverage указывает число октетов, начиная с первого октета заголовка UDP-Lite, используемых для расчёта контрольной суммы. Заголовок UDP-Lite **должен** всегда покрываться контрольной суммой. Вопреки этому требованию значение Checksum Coverage выражается числом октетов от начала заголовка UDP-Lite, как это делается для UDP. Нулевое значение Checksum Coverage показывает, что контрольная сумма вычисляется для всего пакета UDP-Lite. Это означает, что отличные от 0 значения поля Checksum Coverage **должны** быть не меньше 8[2]. Пакеты UDP-Lite со значением поля Checksum Coverage от 1 до 7 **должны** отбрасываться получателем. Безотносительно к значению Checksum Coverage поле Checksum **должно** учитывать псевдозаголовок, основанный на заголовке IP (см. ниже). Пакеты UDP-Lite со значением поля Checksum Coverage, превышающим размер IP, также **должны** отбрасываться.

Поле Checksum содержит 16-битовое поразрядное дополнение до 1 суммы поразрядных дополнений до 1 для информации псевдозаголовка, собранной из заголовка IP, числа октетов, заданного полем Checksum Coverage (начиная с первого октета в заголовке UDP-Lite), возможно дополненного октетом нулей в конце для выравнивания по 16-битовой границе [RFC 1071]. Перед расчётом контрольной суммы значение поля Checksum принимается нулевым. Если рассчитанная контрольная сумма равна 0, она передаётся как «все единицы» (эквивалентны в арифметике с дополнением до 1).

[1]Forward Error Correction.
[2]Размер заголовка UDP-Lite. *Прим. перев.*

Поскольку **недопустима** передача контрольной суммы, состоящей только из нулей, приложениям UDP-Lite, не желающим защищать свои данные с помощью контрольной суммы, следует использовать Checksum Coverage = 8. Это отличается от использования протокола UDP на основе IPv4 тем, что минимальная контрольная сумма UDP-Lite всегда покрывает заголовок UDP-Lite, который включает поле Checksum Coverage.

3.2. Псевдозаголовок

Протоколы UDP и UDP-Lite используют однотипный псевдозаголовок с уровня IP для расчёта контрольной суммы. Этот псевдозаголовок различается для протоколов IPv4 и IPv6. Псевдозаголовок UDP-Lite отличается от псевдозаголовка UDP тем, что поле Length берётся не из заголовка UDP-Lite, а из информации, предоставляемой модулем IP. Расчёт производится так же, как для протокола TCP [RFC 793], и предполагает, что поле Length псевдозаголовка включает заголовок UDP-Lite и все последующие октеты данных IP.

3.3. Интерфейс с приложением

Интерфейсу с прикладным уровнем следует разрешать такие же операции, как для протокола UDP. Кроме того, для передающего приложения следует обеспечивать способ передачи значения Checksum Coverage модулю UDP-Lite. Следует также обеспечивать способ передачи значения Checksum Coverage принимающему приложению или, по крайней мере, позволить принимающему приложению блокировать доставку пакетов, в которых покрытие для контрольной суммы меньше, чем значение, представленное этим приложением.

Рекомендуется по умолчанию протоколу UDP-Lite вести себя, подобно UDP, устанавливая значение поля Checksum Coverage соответствующим размеру пакета UDP-Lite и проверяя весь пакет. Приложениям, желающим определить часть своих данных, как нечувствительные к битовым ошибкам (например, для устойчивых к ошибкам кодеков RTP [RFC 3550]), следует делать это путём явного системного вызова на стороне отправителя. Приложениям, желающим получать данные с неполным покрытием для контрольной суммы, следует информировать принимающую систему с помощью явного системного вызова.

Характеристики каналов, формирующих путь через Internet, могут существенно различаться. Следовательно, трудно делать какие-либо предположения об уровне или характере ошибок, которые могут приводить к повреждению нечувствительной части данных пакетов UDP-Lite. Приложениям, использующим UDP-Lite, не следует делать каких-либо предположений о корректности принятых данных за пределами области, указанной полем Checksum Coverage, и следует, при необходимости пользоваться своими средствами контроля ошибок.

3.4. Интерфейс с IP

Как для UDP, модуль IP должен обеспечивать псевдозаголовок модулю протокола UDP-Lite (его называют также модулем UDPLite). Псевдозаголовок UDP-Lite содержит поля адресов IP и поле протокола из заголовка IP, а также размер данных IP, который определяется на основе поля Length в заголовке IP.

Передающему модулю IP **недопустимо** дополнять данные IP октетами заполнения, поскольку размер данных UDP-Lite, доставляемых принимающему приложению, определяется размером данных IP.

3.5. Джамбограммы

Поле Checksum Coverage имеет размер 16 битов и может представлять значения Checksum Coverage вплоть до 65535 октетов. Это позволяет использовать любое покрытие для контрольной суммы пакетов IP, если они не относятся к числу Jumbogram. Для Jumbogram контрольная сумма будет покрывать все данные (Checksum Coverage = 0) или не более 65535 начальных октетов пакета UDP-Lite.

4. Нижележащий уровень

Поскольку UDP-Lite может доставлять пакеты с повреждёнными данными приложениям, которые пожелали такие пакеты получать, кадры, содержащие пакеты UDP-Lite, не требуется отбрасывать на нижележащих уровнях при обнаружении ошибок в нечувствительной части. Для каналов, поддерживающих частичное детектирование ошибок, поле Checksum Coverage в заголовке UDP-Lite **может** использоваться в качестве рекомендации где не следует проверять наличие ошибок. Нижележащие уровни **должны** использовать строгий механизм детектирования ошибок [RFC 3819] для обнаружения по крайней мере ошибок в чувствительной части и отбрасывания повреждённых пакетов. Чувствительная часть включает октеты, начиная с первого октета заголовка IP и заканчивая последним октетом, указанным полем Checksum Coverage. Чувствительная часть будет, таким образом, трактоваться в точности так же, как для пакета UDP.

Канальные уровни, не поддерживающие частичного детектирования ошибок, подходящего для UDP-Lite, как описано выше, **должны** детектировать ошибки во всем пакете UDP-Lite и отбрасывать повреждённые пакеты [RFC 3819]. Весь пакет UDP-Lite в этом случае трактуется в точности как пакет UDP.

Следует отметить, что протокол UDP-Lite будет отличаться для приложений только в том случае, когда частичное детектирование ошибок, основанное на неполных контрольных суммах UDP-Lite, реализовано также на канальном уровне, как сказано выше. Использование частичного детектирования ошибок на канальном уровне будет давать эффект только при работе через склонные к ошибкам каналы.

5. Совместимость с UDP

Протоколы UDP и UDP-Lite похожи синтаксически и семантически. Приложения, разработанные для UDP могут, следовательно, использовать взамен протокол UDP-Lite и по умолчанию будут получать полное покрытие для контрольной суммы. Сходство протоколов также упрощает реализацию UDP-Lite, поскольку требуется внести сравнительно небольшие изменения в существующие реализации UDP.

Протоколу UDP-Lite был выделен отдельный идентификатор протокола IP – 136 (UDPLite), что позволяет получателю отличить протокол UDP от протокола UDP-Lite. Конечный хост-адресат, не поддерживающий UDP-Lite, будет в общем случае возвращать сообщение ICMP Protocol Unreachable или ICMPv6 Payload Type Unknown (в зависимости от типа

протокола IP). Этот простой метод детектирования не поддерживающих UDP-Lite систем является главным преимуществом использования отдельного идентификатора протокола.

В оставшейся части этой главы даётся обоснование выделению отдельного идентификатора протокола IP для UDP-Lite, вместо использования имеющегося идентификатора для UDP.

Не существует известных проблем взаимодействия между UDP и UDP-Lite при использовании обоими протоколами одного идентификатора IP. В частности, не возникает ситуаций, когда пакет, способный вызвать проблемы, будет доставлен не подозревающему об этом приложению - данные UDP-Lite с частичным покрытием для контрольной суммы не могут быть доставлены приложениям UDP, а пакеты UDP, не полностью заполняющие поле данных IP, не могут быть доставлены приложениям, использующим UDP-Lite.

Однако при использовании протоколами UDP и UDP-Lite одного идентификатора, если реализация UDP-Lite будет передавать пакеты UDP-Lite с неполным покрытием для контрольной суммы реализации UDP, последняя будет отбрасывать пакеты без уведомления, поскольку несоответствие псевдозаголовков приведёт к отказу при проверке контрольной суммы UDP. Ни одно из приложений не получит уведомления о сложившейся ситуации. Решениями этой проблемы могут служить:

1) явная сигнализация на прикладном уровне через основное соединение (пока не используется опция частичного покрытия для контрольной суммы), позволяющая отправителю узнать, поддерживает ли получатель протокол UDP-Lite;

2) использование отдельной сигнализации (например, H.323, SIP или RTCP) для передачи информации о поддержке получателем протокола UDP-Lite.

Поскольку протоколу UDP-Lite присвоен свой идентификатор протокола IP, не возникает необходимости рассмотрения возможности доставки пакета UDP-Lite в ничего не подозревающий порт UDP.

6. Вопросы безопасности

Вопросы безопасности UDP-Lite связаны с взаимодействием данного протокола с механизмами аутентификации и шифрования. При использовании опции неполного покрытия для контрольных сумм UDP-Lite нечувствительная к ошибкам часть пакета может быть изменена в процессе доставки. Это вступает в противоречие с идеей, лежащей в основе большинства механизмов аутентификации - аутентификация считается успешной, если пакет не изменён в процессе передачи. Пока не будет разработан и развернут механизм аутентификации, способный работать только с чувствительной частью пакета, аутентификация всегда будет приводить к отказу для пакетов UDP-Lite с повреждениями в нечувствительной к ошибкам части.

Проверка целостности IPsec (Encapsulation Security Protocol, ESP [RFC 2406] или Authentication Header, AH [RFC 2402]) применяется (как минимум) ко всей области данных пакета IP. Повреждение любого бита в защищённой области будет приводить к тому, что уровень IP на приёмной стороне будет отбрасывать все повреждённые пакеты UDP-Lite.

При использовании IPsec с ESP для шифрования данных канал не может идентифицировать транспортный протокол пересылаемых пакетов путём просмотра области данных IP. В таких случаях канал **должен** обеспечивать стандартную проверку целостности для всего пакета IP. В этом случае протокол UDP-Lite не даёт никаких преимуществ.

Для передачи данных может использоваться шифрование (например, на прикладном или транспортном уровне). В этом случае при повреждении небольшого числа битов пакета механизм дешифровки обычно приводит к распространению ошибки и пакет становится непригодным для использования. Подобное поведение характерно для многих современных механизмов шифрования. Существуют потоковые шифры, которые не приводят к распространению ошибок при дешифровке. Отметим, что отказ от проверки целостности может при некоторых обстоятельствах создавать риск потери конфиденциальности [Bellovin98]. Точность потоковых шифров обусловлена использованием собственных challenge [BB01]. В частности, атакующий может внести предсказуемые изменения в зашифрованный текст даже без его расшифровки.

7. Взаимодействие с IANA

Новый номер протокола IP (1360 был выделен для протокола UDP-Lite. С этим идентификатором связано также имя UDPLite. Такое имя обеспечивает совместимость с широким спектром платформ, поскольку на некоторых платформах символ «-» не может быть частью имени протокольного объекта.

8. Литература

8.1. Нормативные документы

[RFC-768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC-791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC-793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC-1071] Braden, R., Borman, D. and C. Partridge, "Computing the Internet Checksum", RFC 1071, September 1988.

[RFC-2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC-2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

8.2. Дополнительная литература

[Bellovin98] Bellovin, S.M., "Cryptography and the Internet", in Proceedings of CRYPTO '98, August 1988.

[BB01] Bellovin, S. and M. Blaze, "Cryptographic Modes of Operation for the Internet", Second NIST Workshop on Modes of Operation, August 2001.

[3GPP] "Technical Specification Group Services and System Aspects; Quality of Service (QoS) concept and architecture", TS 23.107 V5.9.0, Technical Specification 3rd Generation Partnership Project, June 2003.

[H.264] Hannuksela, M.M., Stockhammer, T., Westerlund, M. and D. Singer, "RTP payload Format for H.264 Video", Internet Draft, Work in Progress[1], March 2003.

[ILBRC] S.V. Andersen, et. al., "Internet Low Bit Rate Codec", Work in Progress[2], March 2003.

[ISO-14496] ISO/IEC International Standard 1446 (MPEG-4), "Information Technology Coding of Audio-Visual Objects", January 2000.

[ITU-H.263] "Video Coding for Low Bit Rate Communication," ITU-T Recommendation H.263, January 1998.

[ITU-H.264] "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification", ITU-T Recommendation H.264, May 2003.

[RFC-3819] Karn, Ed., P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J. And L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.

[RFC-3550] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

[RFC-2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[RFC-2406] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[RFC-3267] Sjoberg, J., Westerlund, M., Lakeaniemi, A. and Q. Xie, "Real-Time Transport Protocol (RTP) Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", RFC 3267, June 2002.

[LDP99] Larzon, L-A., Degermark, M. and S. Pink, "UDP Lite for Real-Time Multimedia Applications", Proceedings of the IEEE International Conference of Communications (ICC), 1999.

9. Благодарности

Спасибо Ghyslain Pelletier за важные технические и редакторские комментарии. Благодарим также Steven Bellovin, Elisabetta Carrara и Mats Naslund за обзор главы, посвященной безопасности, и Peter Eriksson за просмотр документа с точки зрения языка, значительно улучшивший понимание документа.

10. Адреса авторов

Lars-Ake Larzon
Department of CS & EE
Lulea University of Technology
S-971 87 Lulea, Sweden
EMail: lln@csee.ltu.se

Mikael Degermark
Department of Computer Science
The University of Arizona
P.O. Box 210077
Tucson, AZ 85721-0077, USA
EMail: micke@cs.arizona.edu

Stephen Pink
The University of Arizona
P.O. Box 210077
Tucson, AZ 85721-0077, USA
EMail: steve@cs.arizona.edu

Lars-Erik Jonsson
Ericsson AB
Box 920
S-971 28 Lulea, Sweden
EMail: lars-erik.jonsson@ericsson.com

Godred Fairhurst
Department of Engineering
University of Aberdeen
Aberdeen, AB24 3UE, UK
EMail: gorry@erg.abdn.ac.uk

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

[1]Работа опубликована в RFC 3984. *Прим. перев.*
[2]Работа опубликована в RFC 3951. *Прим. перев.*

11. Полное заявление авторских прав

Подтверждение

Финансирование функций RFC Editor обеспечивается Internet Society.