



eucariot 1 октября 2015 в 11:33

Сети для Самых Маленьких. Микровыпуск №5. FAQ по сетевым технологиям

Системное администрирование, Сетевые технологии

Пока весь мир с замиранием ждёт 11-го выпуска СДСМ, посвящённого MPLS BGP L3VPN, я решил сделать вольный перевод неплохой статьи Джереми Стреча с Packetlife.net.

Это подборка небольших FAQ для новичков.

#На каком уровне OSI работает протокол Ч?

#Какая разница между маршрутизатором и многоуровневым коммутатором?

#Какая разница между forwarding и control planes?

#Какая разница между MTU и MSS?

#Какая разница между интерфейсами VLAN и BVI?

#Как работает туннельный интерфейс?

#Что означают четыре типа адресов в NAT?

#Могу ли я использовать адрес сети и широковещательный адрес в NAT-пуле?

#Почему нам нужны IP-адреса? Разве нам не хватит MAC-адресации для всего?

#Позволяет ли QoS расширить пропускную способность?

#На каком уровне OSI работает протокол Ч?

Первая вещь, с которой сталкивается любой, кто изучает сети — это модель OSI (Open Systems Interconnection). Это семиуровневая эталонная модель, официально определённая в IOS/IEC 7498-1. Вы встретите её в любой когда либо напечатанной учебной литературе. Это совершенно обычное дело — ссылаться на OSI при обсуждении взаимодействия между протоколами. Так, например, TCP — это протокол четвёртого уровня, и он сидит на шее IP — протоколе третьего уровня.

Но что это значит на самом деле? Кто решает какому уровню принадлежит протокол? Модель OSI была задумана ещё в 70-е годы, как часть семейства протоколов OSI, которая на полном серьёзе позиционировалась как соперник стеку TCP/IP (спойлер: TCP/IP таки выиграл). Если исключить горстку выживших (наверняка, вы слышали про протокол динамической маршрутизации IS-IS), то протоколы OSI сейчас фактически не используются. Однако эталонная модель OSI, описывающая, как они должны были взаимодействовать, живее всех живых. Что, впрочем, заставляет нас привязывать протоколы одного семейства к уровням, определённым для другого.

По большей части всё работает прекрасно: TCP и UDP едут верхом на IP, который в свою очередь передвигается на Ethernet, PPP или чём бы там ни было другом. Но сорокалетняя модель не всегда может удовлетворить нужды современных протоколов. Возьмём для примера MPLS. Часто его относят к уровню 2,5, потому что он работает поверх канального, но ниже сетевого, не осуществляя при этом ни формирования фреймов ни сквозную адресацию (в отличие от IP-адресов, метки MPLS меняются на каждом узле по мере продвижения пакета к точке назначения). Разумеется, добавление нового уровня между двумя другими разрушает стандартную модель.

Строго говоря, ни один протокол из стека TCP/IP не закреплён официально за каким-либо уровнем OSI именно по той причине, что это разные семейства. Яблоки и апельсины. Эталонная модель — это эталон (*Прим. переводчика: всё-таки русское название немного не соответствует Reference Model, эталон предполагает свою идеальность и стремление ему соответствовать*). OSI помогает иллюстрировать зависимость одних протоколов от других, и кто кем погоняет, но она не может диктовать, как им функционировать.

Но если вдруг кто-то спросит, отвечайте, что MPLS — это протокол третьего уровня.

#Какая разница между маршрутизатором и многоуровневым коммутатором?

В стародавние времена, маршрутизаторы служили для того чтобы передавать пакеты на основе IP-адресов и предоставляли широкий диапазон интерфейсов: Ethernet, E1, Serial, OC-3 итд. В то же время коммутатор передавал пакеты (кадры, прим. для лиги зануд), основываясь на MAC-адресах, и имели только порты Ethernet.

Но в начале 2000-х нашему чёткому пониманию этой разницы пришёл конец — вырисовывались две важные тенденции. Во-первых, появились многоуровневые коммутаторы, которые не просто получили право передавать пакеты, основываясь на IP-

адресах, но и участвовать в протоколах динамической маршрутизации, как самые настоящие маршрутизаторы. Во-вторых, операторы начали необратимый процесс миграции с технологий с коммутацией каналов на современный Ethernet, предоставляющий высокие скорости за низкую плату. Сегодня совершенно в порядке вещей, если маршрутизатор имеет только Ethernet-интерфейсы, как будто бы он коммутатор.

Где лежит граница между маршрутизатором и многоуровневым коммутатором? Существует ли ещё эта граница? Фактическая разница между ними сводится к следующим нескольким пунктам:

- **Плотность портов.** Коммутаторы уровня Enterprise обычно несут на борту 24 или 48 портов. Иногда они могут стекироваться для ещё большего расширения. Основная цель: засунуть настолько много интерфейсов, насколько позволяет передняя панель. Маршрутизатор напротив обычно имеет намного меньше интерфейсов, возможно, разнесённых по разным сменным платам. (Прим. переводчика: если речь идёт о оборудовании операторского класса, то плотность портов на линейных платах маршрутизатора вполне сравнится с коммутаторскими).
- **Скорость.** Коммутаторы созданы для того, чтобы молотить трафик. Сейчас даже скромные офисные коммутаторы зачастую предоставляют пропускную способность на скорости линии. Это достигается за счёт того, что обработка трафика происходит на аппаратных чипсетах без участия CPU. (Прим. переводчика: следует и тут заметить, что и маршрутизаторы сейчас преимущественно используют для передачи трафика FPGA и ASIC и в своей пропускной способности не уступают коммутаторам).
- **Интеллект.** Ключевое же различие, которое может вынудить вас выбрать маршрутизатор — интеллектуальная начинка. Маршрутизатор предоставляет такие функции, как NAT, DPI, Stateful файрвол, шифрование итд — всё это, как правило, не поддерживается коммутатором.

Как бы то ни было, современный мир жидется на оборудовании, изготовленном под конкретные нужды. Однако, если заглянуть в завтра с виртуальными эплайнсами, NFV и SDN, мы приходим к тому, что одна и та же коробочка может выполнять совершенно разные роли в зависимости от своего положения в сети.

#Какая разница между forwarding и control planes?

Для новичков это, несомненно, источник путаницы.

Forwarding plane часто называют Data Plane, а по-русски самый удачный вариант — плоскость коммутации. Её задача — доставить пакет из пункта А в пункт Б. **Плоскость коммутации коммутирует.**

Control plane — плоскость управления — обслуживает функции предписывающие, как должна работать плоскость коммутации. **Плоскость управления управляет.**

Вот например, у вас есть маршрутизатор с OSPF. Он обменивается маршрутной информацией с соседними маршрутизаторами OSPF, составляет граф всей сети и вычисляет маршруты. Когда таблица маршрутизации (RIB) построена, маршрутизатор устанавливает лучший маршрут до каждой известной точки назначения в таблицу коммутации (FIB). Это функции control plane.

Когда тот же маршрутизатор получает IP-пакет, он ищет адрес назначения в своей таблице коммутации, чтобы определить интерфейс, в который пакет нужно отправить. Далее пакет передаётся в буфер выходного интерфейса и затем в кабель. Это функции forwarding plane.

Чувствуете различие? Плоскость коммутации отвечает за приём и передачу пакетов, в то время как плоскость управления — за то, как именно принимается решение о передаче пакета.

Плоскость коммутации реализована, как правило, в железе, иными словами выполняется специальными чипсетами (например, Network Processor обращается к TCAM, чтобы быстро извлечь выходной интерфейс из FIB), не требуя обращения к CPU.

Плоскость управления же работает на CPU и в обычной памяти, что очень похоже на работу персонального компьютера. Дело в том, что уровень управления выполняет очень сложные функции, которые с одной стороны не нужны в реальном времени, а с другой их проблематично реализовать в железе. Например, совершенно не важна задержка в несколько миллисекунд, когда маршрутизатор устанавливает маршрут в таблицу коммутации, в то время как для уровня коммутации это может быть серьёзной деградацией производительности.

#Какая разница между MTU и MSS?

Maximum transmission unit (MTU) говорит о максимальном объёме данных, который может нести один пакет. Обычно мы говорим о MTU в отношении Ethernet (хотя другие протоколы, конечно, тоже имеют свои MTU). MTU по умолчанию на большинстве платформ — 1500 байтов. Это означает, что узел может передать кадр, несущий 1500 байтов полезной нагрузки.

Сюда не включены 14 байтов заголовка Ethernet (18 в случае 802.1q) и 4 байта поля FSC. Итоговый же размер кадра 1518 байтов (1522 в случае 802.1q). Многие узлы сейчас поддерживают джамбофреймы (jumbo), для этого стандартный MTU увеличивается до 9000+ байтов.

Maximum segment size (MSS) — это величина характерная для TCP, которая показывает максимальную полезную TCP нагрузку в пакете, фактически это MTU для TCP. TCP MSS вычисляется, исходя из значения Ethernet MTU (а, может, и не Ethernet) на интерфейсе. Поскольку TCP должен втиснуться в кадр Ethernet, MSS должен быть меньше, чем MTU. В идеале MSS должен быть максимально возможным: MTU-размер заголовка IP-размер заголовка TCP.

Предположим MTU 1500 байтов, вычитаем из него 20 байтов IPv4 адреса и ещё 20 байтов TCP и получаем MSS 1460 байтов. IPv6 с его удлинённым заголовком оставит для MSS всего 1440 байтов.

TCP MSS определяется один раз в ходе установления соединения. Каждый узел включает свой MSS в опции TCP в первый пакет (тот, что с флагом SYN), и оба узла выбирают наименьшее значение из двух как MSS сессии. Однажды установленный MSS уже не меняется в течение жизни сессии.

#Какая разница между интерфейсами VLAN и BVI?

VLAN-интерфейс, известный также как SVI (Switch Virtual Interface) или RVI (Routed VLAN Interface) — это виртуальный интерфейс на многоуровневом коммутаторе. Он обеспечивает маршрутизацию и часто служит шлюзом по умолчанию для локального сегмента сети. VLAN-интерфейс обычно ведёт себя и настраивается как физический интерфейс маршрутизатора: на него можно назначить IP, он участвует в VRRP, может иметь ACL итд. Вы можете представить себе, что это физический интерфейс внутри коммутатора, а можете, наоборот, вообразить, что это маршрутизирующий интерфейс вне коммутатора, на котором terminates данный VLAN.

Bridge group Virtual Interface (BVI) служит похожим целям, но существует на маршрутизаторе, на котором нет концепции VLAN, потому что всего его порты обычно работают на L3 (*Прим. переводчика: на маршрутизаторах концепция VLAN вполне может присутствовать*). Bridge group заставляет два или более портов работать на L2, разделяя между ними широковещательный домен. BVI связывает интерфейсы в Bridge Group и служит виртуальным L3-интерфейсом для всех сегментов, подключённых к нему. Когда маршрутизатор работает одновременно на L2 и L3, его называют Integrated Routing and Bridging (IRB).

В то время, как VLAN-интерфейс — жизненная необходимость многоуровневого коммутатора, IRB — нишевая вещь, которая может использоваться, например, на точках доступа WiFi.

#Как работает туннельный интерфейс?

Многие люди испытывают трудности с пониманием концепции туннельных интерфейсов (*Прим. переводчика: действительно?*). Туннелирование — это просто инкапсуляция одних пакетов внутри других при передаче их между двумя точками. Туннельный интерфейс используется для достижения такой инкапсуляции для маршрутизируемых VPN, которые позволяют защититься и абстрагироваться от топологии нижележащей сети. Существует много методов инкапсуляции, включающие IPsec, GRE, MPLS итд.

Несмотря на то, что туннельный интерфейс имеет виртуальную природу, ведёт себя он как и любой другой, когда дело доходит до маршрутизации, с той лишь разницей, что когда пакет выходит через туннельный интерфейс, он упаковывается в новый пакет, для которого снова принимается решение о маршрутизации. Новый беременный пакет отправляется в среду и достигает в конечном счёте точки назначения. На другом конце туннеля внешние заголовки снимаются, и на свет выходит оригинальный пакет, над которым снова принимается решение о маршрутизации.

#Что означают четыре типа адресов в NAT?

Существует четыре класса IP-адресов в контексте NAT:

- Inside global
- Inside local
- Outside local
- Outside global

К сожалению, эти термины редко объясняются в документации достаточно понятно.

Каждый из них описывает два атрибута: местоположение (**location**) и точка зрения (**perspective**).

Местоположение сообщает о каком узле идёт речь. Внутри сети (до NAT) — Inside; во внешней сети (после NAT) — Outside.

Точка зрения сообщает о том, откуда мы смотрим на этот узел. Изнутри нашей сети — Local; из внешней сети — Global.

Возьмём для примера случай, когда вы с компьютера с приватным адресом 192.168.0.10 хотите зайти по telnet на адрес в Интернете 94.142.241.111. Из пула NAT вам выделен IP-адрес 192.0.2.10.

Вот так будет выглядеть таблица трансляций:

```
R2# show ip nat translations
Pro Inside global      Inside local          Outside local          Outside global
tcp 192.0.2.10:32978   192.168.0.10:32978   94.142.241.111:23     94.142.241.111:23
```

Разберёмся?

Inside Global — как внутренний узел выглядит извне. Сервер в Интернете действительно видит адрес из вашего пула NAT.

Inside Local — как внутренний адрес выглядит изнутри — приватный адрес компьютера

Outside Local — как внешний адрес выглядит изнутри — видим его публичный адрес и порт 23.

Outside Global — тут должно быть то, как выглядит внешний адрес извне, но ваш NAT таких трансляций не умеет, поэтому адрес совпадает с Outside Local.

#Могу ли я использовать адрес сети и широковещательный адрес в NAT-пуле?

Да.

Во-первых, в контексте пула NAT вообще нет понятий маски адрес сети и широковещательный адрес.

Далее прим. переводчика.

Во-вторых адрес сети и широковещательный адрес определяются маской подсети — без неё они теряют смысл. Поэтому считать ли адрес 192.168.0.255 широковещательным адресом, а 192.168.1.0 адресом сети зависит целиком и полностью от маски: для /23 ответ нет, для /24 и более ответ да, а для /32 снова нет.

Поэтому адрес 192.168.0.255 вы можете не только указать в пуле, но даже настроить на интерфейсе с маской /23.

#Почему нам нужны IP-адреса? Разве нам не хватит MAC-адресации для всего?

Когда новичок начинает изучение MAC-адресов, он видит, что они должны быть уникальными глобально. И возникает закономерный вопрос, почему бы не использовать MAC-адреса для сквозной адресации через весь Интернет, не прибегая вообще к IP? Однако существует несколько достаточно весомых причин привлечь IP.

Во-первых, не все сети имеют MAC-адресацию. Вообще такой тип свойственен только семейству 802. Очень легко забыть об этом в мире, где практически всё — Ethernet или его вариации (например IEEE 802.11 WiFi). Но во времена юности Ethernet несколько десятилетий назад буйствовало беззаконие в сфере протоколов: Token Ring, Ethernet, Frame Relay, ATM боролись за место в маршрутизаторе. И обеспечить взаимодействие узлов из Token Ring с узлами из ATM посредством MAC-адресов было проблематично — нужен был протокол сетевого уровня.

Во-вторых, IP-адреса мобильны — они могут назначаться администраторами или даже выдаваться автоматически, в то время, как MAC-адреса вшиты в сетевой адаптер на веки вечные. Технически MAC-адрес, конечно, тоже можно поменять, но это не предполагалось изначально и сейчас нет никаких средств для удобного управления ими.

Но самая главная причина третья — IP масштабируем и может связывать огромные сети, а Ethernet — удел небольших сегментов. Пространство IP-адресов иерархично, MAC-адресов — плоско. 254 узла одной локальной сети могут быть агрегированы в одну подсеть /24. 8 подсетей /24 могут быть агрегированы в одну /21. Это возможно, потому что блоки адресов обычно располагаются рядом в Интернете. Всё, о чём нужно заботиться в этом случае маршрутизатору — как добраться до подсети.

MAC-адреса же каждый сам по себе, так как назначаются псевдослучайным образом на производстве, и два адреса, различающихся только в последнем бите, могут оказаться в диаметральных концах планеты. Если вдруг кому-то взбредёт в голову использовать MAC-адреса для сквозной адресации в Интернете, он столкнётся с тем, что маршрутизаторам будет нужно знать адрес каждого отдельно взятого узла в глобальной сети. Здравствуй, интернет вещей.

Далее прим. переводчика.

Освещённый в оригинальной статье вопрос на самом деле простой — одного отсутствия масштабирования достаточно для того, чтобы отказаться от этой идеи.

Гораздо интереснее обратный вопрос: Почему нам нужны MAC-адреса? Разве нам не хватит IP-адресации для всего? Тут всё не так однозначно. Почему бы действительно в современном мире, где скоро название стека можно менять на TCP/IP/Ethernet, не отказаться совсем от адресации на L2 и позволить узлам в сегменте взаимодействовать по IP?

ARP больше не нужен — пакет коммутируется по IP (кстати, уже сейчас существуют коммутаторы, которые действительно могут производить IP Learning вместо MAC Learning). Широковещание доступно так же через адрес 255.255.255.255.

При этом, я не предлагаю отказаться от Ethernet или L2 совсем, нет — этот уровень абстракции необходим — сетевой не должен работать напрямую с физическим, заниматься фреймингом, проверкой целостности итд; мы просто убираем адресацию из L2.

Сложность начинается на самом деле при передаче пакета из одной подсети в другую через череду маршрутизаторов. Тут даёт о себе знать широковещательная природа Ethernet. В заголовке IP, адрес назначения фиксирован и не меняется по мере продвижения пакета. Поэтому встаёт вопрос, как правильно переслать пакет между маршрутизаторами. Сейчас как раз для этого используются MAC-адреса Next-Hop. Дело в том, что за Ethernet-интерфейсом маршрутизатора может быть не один соседний маршрутизатор, а два, три, десяток, и здесь придётся добавлять ещё какой-то идентификатор Next-hop.

В реальном мире в 99,9% мы используем P2P линии между маршрутизаторами и тут нет необходимости в добавлении адреса Next-hop в пакет — больше ведь и слать некому — просто отправляем кадр в кабель. Тут можно вспомнить PPP, где хоть формально поле «адрес» и есть, но оно фактически не используется.

Но концепция Ethernet, который изначально планировался только для локальных сегментов с пользовательскими машинами, не предусматривает сценарий P2P отдельно.

В итоге адресацию с уровня Ethernet мы не можем убрать. Однако тут до сих пор остаётся вопрос — зачем MAC-адреса, ведь в заголовке Ethernet мы могли бы указывать IP-адрес Next-Hop, который менялся бы также на каждом узле.

В целом это верно, но такой подход ломает идеологию стека протоколов, предполагающую независимость уровней друг от друга. Сейчас, например, легко можно выкинуть Ethernet и вместо него использовать xDSL или PON или, прости Лейбниц, Frame Relay — сложности лишь административные и финансовые. Также, поверх Ethernet технически вы можете пустить собственный сетевой протокол IPЧ — и это всё будет работать с минимальными изменениями (добавить новый Ethertype).

Замечу, что этот вопрос нельзя обсуждать в отрыве от исторического и административного контекста. Даже если мы возьмём на себя смелость предположить, что мы нашли идеальное сочетание идеальных протоколов IP+Ethernet, и ближайшие 300 лет нам не грозят глобальные изменения, нужно помнить, что 20 лет назад мир был другим, как мы уже говорили выше, и Ethernet был лишь одним из. Мы не могли так жёстко связывать сетевой и канальный уровни. А теперь сети, которые уже работают, и нам для этого как правило не нужно прилагать титанических усилий, никто не будет переделывать просто потому, что кажется избыточным одновременное использование IP и MAC-адресации.

Кстати, возможно, вы будете несколько удивлены, но часть описанных идей войдут в нашу с вами жизнь в лице IPv6 с его концепцией Link-Local адресов.

#Позволяет ли QoS расширить пропускную способность?

Среди новичков иногда существует заблуждение, что QoS — это волшебная технология, позволяющая пропихнуть через линию больше пакетов, чем она может. Это не так. К примеру, если ваш интернет-канал ограничен 10Мб/с, вы никогда не сможете отправить в него больше. Задача QoS — отдавать предпочтения одним типам трафика над другими. Таким образом во время перегрузки на линии (при попытке отправить больше 10 Мб/с) менее важный трафик будет отброшен в пользу свободной передачи более приоритетного.

Обычно QoS применяется для того, чтобы защитить трафик реального времени, такой как голос или видеоконференции от трафика, терпимого к задержкам и потерям — WEB, почта, FTP, Torrent итд. Кроме того QoS поможет избежать оккупирования всей полосы передач большого объёма трафика, типа резервного копирования серверов.

Рассмотрим ситуацию, когда у вас есть офис, подключенный через два канала Е1 с общей пропускной способностью 4Мб/с. По

этой линии передаётся и голос и данные. Чтобы во время перегрузки голосовой трафик не испытывал деградацию, с помощью QoS можно выделить гарантированную полосу для него. Оставшаяся часть будет доступна для данных. Однако если после этого трафик с данными заметно ухудшится, то QoS уже не поможет — в этом случае придётся расширять канал.

Переводчик позволил себе некоторые вольности в русскоязычных терминах, которые позволят, как ему кажется, лучше понять смысл.

Теги: сети для самых маленьких, FAQ, сетевые технологии, IP, NAT, SVI, BVI, OSI, forwarding, routing

Хабы: Системное администрирование, Сетевые технологии

↑ +32 ↓ 690 87k 7 Поделиться



Марат @eucariot
Пользователь

ПОХОЖИЕ ПУБЛИКАЦИИ

22 декабря 2017 в 12:58

Сети для самых маленьких. Часть четырнадцатая. Путь пакета

↑ +28 52,1k 413 17

1 августа 2016 в 11:07

Сети Для Самых Маленьких. Микровыпуск №6. MPLS L3VPN и доступ в Интернет

↑ +29 26,8k 290 5

18 июля 2012 в 12:21

Сети для самых маленьких. Часть пятая. ACL и NAT

↑ +89 338k 1629 48

ЗАКАЗЫ

Оптимизация mysql и запросов к ней со стороны Битрикс сайта
20 000 ₽ за проект • 3 отклика • 15 просмотров

Настроить Commers на Друпал 7, выборочный вывод товаров и стили
3 000 ₽ за проект • 3 отклика • 9 просмотров

Сверстать макет на Vue.js
50 000 ₽ за проект • 6 откликов • 30 просмотров

Разработать приложение для Android
15 000 ₽ за проект • 2 отклика • 19 просмотров

Нарисовать иллюстрации
8 000 ₽ за проект • 7 откликов • 31 просмотр

Больше заказов на Хабр Фрилансе






Комментарии 7



pavelsh 2 октября 2015 в 04:01

↑ 0 ↓

Да-да-да! Мы очень ждем выпуск с MPLS BGP L3VPN

 **eucariot** 2 октября 2015 в 20:06    

 0 

В этом году будет. Надеюсь, даже в ноябре.

 **nazgul** 2 октября 2015 в 07:06  

 0 

Я уже было подумал, что серия умерла. Рад слышать, что работа на следующей частью идёт. Очень жду:)

 **Rel1cto**  2 октября 2015 в 09:49  






 +1 

" Модель OSI была задумана ещё в 70-е годы, как часть семейства протоколов OSI, которая на полном серьёзе позиционировалась как соперник стеку TCP/IP (спойлер: TCP/IP таки выиграл)."

Если я ничего не путаю, де-юре конкурс выиграли таки стек и модель OSI. После чего разработка семейства OSI была поставлена под жёсткий контроль государства, завязла в бюрократии и не смогла развивать протокол со скоростью, требуемой рынком. А TCP/IP никто не ограничивал, поэтому нужные функции реализовывались быстро, и он стал стандартом де-факто. Поэтому мы и имеем такую странную ситуацию — семейство протоколов используется одно, а модель — от другого семейства.

«Каждый узел включает свой MSS в опции TCP в первый пакет (тот, что с флагом SYN), и оба узла выбирают наименьшее значение из двух как MSS сессии.»

Что такое MSS сессии? Насколько мне известно, тут механизм похож на hold-timer в EIGRP. То есть каждый узел говорит, с каким максимальным MSS лично он бы хотел принимать данные. Но они ничего не согласовывают и ни о чём не договариваются — каждый может выбрать свой собственный MSS для передачи (равный минимальному значению между полученным от соседа пожеланием и собственным MTU на исходящем интерфейсе) в рамках этой сессии.

 **askbow** 6 ноября 2015 в 18:49    

 0 

>модель OSI...






Вроде дело было так (если имеющиеся источники не врут): процедура создания OSI изначально была очень бюрократизирована и была «открыта» (т.е. кто угодно мог высказывать мнение — а сколько людей (заинтересованных организаций), столько и мнений), потому медленна;

TCP/IP же был, в некотором смысле, под диктатурой сравнительно небольшой группы инженеров (получавших гранты государства через бюджет ARPA) — у них дело шло быстрее (ну и плюс наличие реализации в коде и железе ещё до стандартизации).

 **Javian** 6 октября 2015 в 08:49  

 0 

Ремарка: «Сети для Самых Маленьких» должны быть иллюстрированы :) Как любая другая публикация для Самых Маленьких.

 **eucariot** 6 октября 2015 в 09:07    

 0 

Следуя традиции, пришлось бы и видео записать :)

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ

Сутки

Неделя

Месяц

Как платить программистам

Не говорите «I feel myself», и другие правила английского языка, которые вгоняют в ступор

Инсайды от сотрудника Facebook: как попасть на стажировку, получить оффер и все о работе в компании

Умные пайетки

Визуальные и приборные правила полетов

Ваш аккаунт	Разделы	Информация	Услуги
Войти	Публикации	Устройство сайта	Реклама
Регистрация	Новости	Для авторов	Тарифы
	Хабы	Для компаний	Контент
	Компании	Документы	Семинары
	Пользователи	Соглашение	Мегaproекты
	Песочница	Конфиденциальность	

Если нашли опечатку в посте, выделите ее и нажмите Ctrl+Enter, чтобы сообщить автору.