IPv4r2
расширение протокола IPv4
до версии (ревизии) два,
описание протокола от 12 декабря 2015 года.

Полянин М.А.
12 декабря 2015.

Базовое расширение IPv4 до IPv4r2.

Базовое расширение IPv4 до IPv4r2 заключается в добавлении к IPv4 новых опций заголовка IPv4 для поддержки IPv4r2 расширенной адресации и IPv4r2 индексов локальных сокетов шлюза.

Описание IPv4r2 расширенной адресации и IPv4r2 индексов шлюза приведено далее, но для понимания терминов "множество сетей IPv4r2" и "глобальная адресация IPv4r2", о которых говорится в описании опций IPv4r2, необходимо сразу сказать, что:

- Любая сеть IPv4r2 имеет, в терминах IPv4 адресации, многоуровневую иерархическую структуру адреса, т.е. выглядит как несколько вложенных IPv4 адресов. Формат URL "протокол://a1.b1.c1.d1/a2.b2.c2.d2/..." описывает адресацию в сети IPv4r2. При этом IPv4 адрес a1.b1.c1.d1, который указан в IPv4 заголовке это, в терминах IPv4r2 адресации, адрес в корневой сети IPv4 (уровень корневой сети IPv4).
- Каждый адрес в корневой сети IPv4 определяет точку входа в IPv4r2 сеть, таким образом, сетей IPv4r2 получается множество и формат IPv4 адресов внутренних уровней в этих сетях зависит от структуры конкретной IPv4r2 сети. Если корневая сеть IPv4 для адреса a1.b1.c1.d1 является глобальной, то и вся эта сеть IPv4r2 может быть адресована глобально, на уровне интернет (это и есть та глобальная адресация, про которую говорится в описании опций IPv4r2).

Расширенная IPv4r2 адресация подразделяется на несколько типов:

- основная IPv4r2 адресация
  - базовая
  - обобщенная
- дополнительная IPv4r2 адресация сетей пользователя

Основная базовая IPv4r2 адресация.

Для добавления свойств IPv4r2 к заголовку IPv4 используется опция 0x88 (136) заголовка IPv4 специального формата. В

терминах IPv4r2 опция 0x88 называется op1. По правилам опций заголовка IPv4, для любой опций IPv4 после октета (в этом тексте октет эквивалентен байту) тип (для op1=0x88) идет октет длина, хранящий размер опции, размер опций включает в себя и первые два байта опции (включает байты тип 0x88 и длина).

Базовые варианты кодирования новых IPv4r2 опций:

- <op1>[8]<длина=8>[8]<флаги>[3]<адрес>[45]
  IPv4r2 асимметричная адресация 45 бит
- <op1>[8]<длина=12>[8]<флаги>[3]<адрес>[45]<индекс>[32]
  IPv4r2 асимметричная адресация 45 бит
  IPv4r2 индекс локального сокета шлюза 32 бита
- <op1>[8]<длина=16>[8]<адрес источника>[48]<адрес назначения>[48]<индекс>[16]
  IPv4r2 адресация 48 бит
  IPv4r2 индекс локального сокета шлюза 16 бит

Для IPv4r2 асимметричной адресации 45 бит, трех- битовое поле флагов обозначает:

- 010b
  расширение адреса источника
- 100b
  расширение адреса назначения

Описание базовых вариантов кодирования IPv4r2 опций.

"IPv4r2 асимметричная адресация 45 бит" (т.е. расширение адреса на 45 бит, общий размер адреса 77 бит) это основной вид глобальной адресации пользовательских серверов в сетях IPv4r2 специального типа, когда IPv4r2 источник подключен к сети IPv4r2 через IPv4 шлюз провайдера (источник использует расширение адреса назначения).

"IPv4r2 асимметричная адресация 45 бит с индексом шлюза" автоматически используется при доставке пакетов в формате "IPv4r2 асимметричная адресация 45 бит" на участке сети между шлюзом провайдера и пользовательским сервером, в том случае, когда IPv4 адрес шлюза провайдера перегружен локальными подключениями от клиентов провайдера. Практически размер IPv4r2 заголовка от этого возрастает на 4 байта.

"IPv4r2 адресация 48 бит" (т.е. расширение адреса на 48 бит, общий размер адреса 80 бит) это расширенный вид 45 битной основной глобальной адресации в сетях IPv4r2 специального типа, когда оба IPv4r2 адреса уникальны на глобальном уровне, поэтому IPv4 шлюз провайдера не обязателен, но может быть использован. Для преобразования адреса 45 бит в адрес 48 бит и обратно используются специальные правила, которые рассмотрены далее в разделе "полное описание IPv4r2".

Рекомендуемое значение глобальной уникальной адресации для сетей IPv4r2 - 77 бит.

Общий формат кодирования IPv4r2 опций.

Тип расширения IPv4r2, который задается опцией IPv4r2 op1=136, определяется размером опции, а также дополнительными флагами в области данных опции, флаги применяются тогда, когда одному размеру опции соответствует несколько разных расширений.

Список дополнительных кодировок для IPv4r2 опции op1:

- <op1>[8]<длина=4>[8]<номер потока>[16]
  IPv4 зарезервировано для номера потока
- <op1>[8]<длина=6>[8]<индекс>[32]
  IPv4r2 индекс локального сокета шлюза 32 бит
- <op1>[8]<длина=10>[8]<флаги>[3]<адрес>[45]<индекс>[16]
  IPv4r2 асимметричная адресация 45 бит
  IPv4r2 индекс локального сокета шлюза 16 бит
- <op1>[8]<длина=14>[8]<адрес источника>[48]<адрес назначения>[48]
  IPv4r2 адресация 48 бит
- <op1>[8]<длина=18>[8]<адрес источника>[48]<адрес назначения>[48]<индекс>[32]
  IPv4r2 адресация 48 бит
  IPv4r2 индекс локального сокета шлюза 32 бит

Для IPv4r2 асимметричной адресации 45 бит, трех- битовое поле флагов дополнительно обозначает:

- 001b
  индекс локального сокета шлюза

Для дополнительных IPv4r2 расширений, когда одному размеру опции op1 соответствует несколько разных расширений, но флаги задать нельзя, используется опция IPv4r2 op2 (числовое значение еще не зафиксировано, предположительно 0x8A).

Список базовых кодировок для IPv4r2 опции op2:

- <op2>[8]<длина=4>[8]<индекс>[16]
  IPv4r2 индекс локального сокета шлюза 16 бит
- <op2>[8]<длина=12>[8]<адрес источника>[40]<адрес назначения>[40]
  IPv4r2 короткая адресация 40 бит
- <op2>[8]<длина=16>[8]<адрес источника>[40]<адрес назначения>[40]<индекс>[32]
  IPv4r2 короткая адресация 40 бит
  IPv4r2 индекс локального сокета шлюза 32 бит

IPv4r2 адресация 40 бит (т.е. расширение адреса на 40 бит, общий размер адреса 72 бит) это короткий вид 45 битной основной адресации в сетях IPv4r2 специального типа, когда оба IPv4r2 адреса уникальны на глобальном уровне. Практически это позволяет уменьшить размер IPv4r2 заголовка на 4 байта по сравнению с такой же адресацией 48 бит, когда IPv4 шлюз провайдера не используется и если структура этой сети IPv4r2 допускает 72 битную глобальную уникальную адресацию, эта возможность зависит от структуры конкретной сети IPv4r2. Для преобразования адреса 45 бит в адрес 40 бит и обратно используются специальные правила, которые рассмотрены далее в разделе "полное описание IPv4r2".

Список дополнительных кодировок для IPv4r2 опции op2:

- <op2>[8]<длина=14>[8]<адрес источника>[40]<адрес назначения>[40]<индекс>[16]
  IPv4r2 короткая адресация 40 бит
  IPv4r2 индекс локального сокета шлюза 16 бит

Основная обобщенная IPv4r2 адресация.

Обобщенная IPv4r2 адресация это (по сравнению с базовой IPv4r2 адресацией) альтернативный и менее эффективный способ указания адресов в сетях IPv4r2.

Для дополнительных IPv4r2 расширений, когда одному размеру опций op1, op2 соответствует несколько разных расширений, но флаги задать нельзя, используется опция IPv4r2 op3 (числовое значение еще не зафиксировано, предположительно 0x8B).

Список основных адресных кодировок для обобщенной IPv4r2 адресации:

- <op1>[8]<длина=3+>[8]<формат опции=1>[1]<флаг адреса>[1]<младший байт адреса>[6]<адрес>[0+]
  длина больше равна 3 и всегда нечетная
  IPv4r2 асимметричная обобщенная адресация 6+ бит
- <op2>[8]<длина=7+>[8]<формат опции=1>[1]<флаг адреса>[1]<младший байт адреса>[6]<адрес>[0+]<индекс>[32]
  длина больше равна 7 и всегда нечетная
  IPv4r2 асимметричная обобщенная адресация 6+ бит
  IPv4r2 индекс локального сокета шлюза 32 бит
- <op3>[8]<длина=4+>[8]<формат опции=1>[1]<размер индекса>[2]<размер адреса источника>[5]<адрес источника>[0+]<адрес назначения>[0+]<индекс>[0+]
  длина больше равна 4
  IPv4r2 обобщенная адресация
  размер адреса источника в байтах 0..31
  размер индекса локального сокета шлюза в байтах 0..3
  размер адреса назначения = длина-3-"размер адреса источника"-"размер индекса"

Формат флага адреса для асимметричной обобщенной адресации:

- флаг адреса
  - 0b - расширение адреса источника
    1b - расширение адреса назначения

IPv4r2 асимметричная адресация 6+ бит это обобщенная глобальная IPv4r2 адресация при работе через IPv4 шлюз провайдера. Есть варианты кодирования с индексом локального сокета и без такого индекса.

Распределение байтов поля адреса обобщенной IPv4r2 адресации по уровням IPv4r2 адресации имеет сложную структуру и отличается от распределения битов поля адреса в основных режимах IPv4r2 адресации (в режимах IPv4r2 адресации фиксированного 40, 45 и 48 бит размера). Обобщенная IPv4r2 адресация допустима, но практического смысла не имеет. Подробнее смотри в разделе "полное описание IPv4r2".

Список дополнительных адресных кодировок для обобщенной IPv4r2 адресации:
- <op2>[8]<длина=5+>[8]<формат опции=0>[1]<флаг адреса>[1]<младший байт адреса>[6]<адрес>[0+]<индекс>[16]
  длина больше равна 5 и всегда нечетная
  IPv4r2 асимметричная обобщенная адресация 6+ бит
  IPv4r2 индекс локального сокета шлюза 16 бит

Дополнительная IPv4r2 адресация сети пользователя.

Дополнительная IPv4r2 адресация пользовательской сети позволяет на один централизованно выделенный пользователю IPv4r2 адрес оперативно размещать дополнительные ресурсы пользователя, глобально адресуемые и уникальные путем привязки к этому выделенному IPv4r2 адресу.

Дополнительная адресация пользовательской сети это отдельное адресное пространство, которое не может быть адресовано через основную или обобщенную адресацию, именно поэтому в нем можно выделять ресурсы самому владельцу IPv4r2 адреса.

Список кодировок IPv4r2 адресации пользовательской сети:
- <op3>[8]<длина=3+>[8]<формат опции=000>[3]<флаг адреса>[1]<старший байт адреса>[4]<адрес>[0+]
  длина больше равна 3
  IPv4r2 асимметричная адресация расширения пользовательской сети
  формат байтов поля адреса: от старшего байта к младшему
- <op3>[8]<длина=4+>[8]<формат опции=001>[1]<число байт адреса источника>[5]<байты источника>[0+]<байты назначения>[0+]
  длина больше равна 4
  IPv4r2 адресация расширения пользовательской сети
  число байт назначения = длина-3-"число байт источника"

формат байтов поля адреса: от старшего байта к младшему

Формат флага адреса для IPv4r2 адресации пользовательской сети:

- флаг адреса
  - 0b - расширение адреса источника
    1b - расширение адреса назначения

Подсети IPv4r2.

В IPv4r2 сетях, подсети вида "IPv4r2 адрес нулевого хоста подсети"/:"число бит маски подсети", могут формироваться без выделения IPv4r2 адресов на широковещательный адрес и маршрут к сети, для этого вместо выделенного IPv4r2 адреса применяются IPv4r2 опции, которые кодируют:

- число бит маски (или обратной маски) такой подсети для IPv4r2 адресата;
- флаг широковещательный запрос для IPv4r2 адреса назначения.

Список кодировок для IPv4r2 подсетей:

- <op3>[8]<длина=4..5>[8]<формат опции=01>[2]<флаги маски>[6]<маска источника>[0+]<маска назначения>[0+]
  длина от 4 до 5
  размер поля маска источника и маска назначения один байт, поле содержит число бит маски подсети вида /:"число бит маски подсети"
  - флаги маски сети
    000001b - есть поле число бит маски сети источника, хранящее маску
    000010b - есть поле число бит маски сети источника, хранящее обратную маску
    000100b - есть поле число бит маски сети назначения, хранящее маску
    001000b - есть поле число бит маски сети назначения, хранящее обратную маску
    010000b - это IPv4r2 ответ от службы сети источника
    100000b - это широковещательный IPv4r2 запрос к сети назначения

Если установлен флаг "широковещательный IPv4r2 запрос к сети назначения", то IPv4r2 адрес в поле назначения хранит не адрес хоста, а после применения маски, этот адрес указывает на сеть назначения.

Если установлен флаг "IPv4r2 ответ от службы сети источника", то IPv4r2 адрес в поле источника хранит не адрес хоста, а после применения маски, этот адрес указывает на сеть источника.

Поле маски включает в себя и пространство дополнительной IPv4r2 адресации пользовательской сети - основное и пользовательское адресные пространство для маски подсети объединяется, основное адресное пространство для маски подсети

идет первым.

Вспомогательные IPv4r2 опции.

Список вспомогательных IPv4r2 опций:

- <op2>[8]<длина=6>[8]<отметка времени>[32]
  Для установления времени жизни IPv4r2 пакета, источник IPv4r2 может отмечать свои пакеты отметкой времени синхронизированного со своим корневым a1.b1.c1.d1 (UTC время в секундах). Для этого IPv4r2 определяет механизм опроса источника времени в корневой IPv4 сети:
  - корневой узел a1.b1.c1.d1 сети IPv4r2 обязан реализовать ответы о времени на запрос от любого другого корневого IPv4 не реже раза в минуту после первого ответа для каждого такого корневого адреса;
  - IPv4r2 получатель обращается за временем a1.b1.c1.d1 источника к своему корневому a1.b1.c1.d1.

Комбинации новых IPv4r2 опций.

Опции op1, op2, op3 и флаги могут комбинироваться по их смыслу, при этом нельзя:

- аддитивно наращивать расширение одного и того же адреса, несколько раз применяя опции такого расширения;
- аддитивно наращивать индекс шлюза с помощью полей плюс флагов в одной опции или с помощью нескольких опций;
- комбинировать обобщенную и основную адресацию IPv4r2 для одного и того же адреса;

Например, можно комбинировать <op1>[8]<длина=14>[8] и <op1>[8]<длина=6>[8] для 32 индексации локальных сокетов шлюза и 48 битного расширения адреса, тем увеличивая IPv4r2 заголовок до 20 байт, заместо более компактного 16 байтового формата <op1>[8]<длина=16>[8] с 16 битной индексацией локальных сокетов шлюза, если 16 бит индексов сокета шлюза не хватает.

С точки зрения эффективности маршрутизации, лучше задавать расширение адресов и индексы шлюза только одной опцией, идущей в заголовке первой. Опция дополнительной IPv4r2 адресации пользовательской сети (рекомендуется задавать второй) и опция IPv4r2 подсетей (рекомендуется задавать третьей) обычно анализируются только конечными IPv4r2 адресатами, промежуточные IPv4r2 маршрутизаторы их игнорируют.

IPv4r2 пакеты с ошибочными адресными опциями не могут быть доставлены и будут отброшены IPv4r2 маршрутизаторами и адресатами.

Причины, лежащие в основе выбора свойств протокола IPv4r2.

Введение.

Перейти с IPv4 на IPv6 это все равно что поменять уровень и частоту напряжения в сети, после того как это повсеместно

стало 220 Вольт и 50 Гц, зачем это нужно затевать пользователю? Чтобы у него все отключилось?

В общем случае, если спросить "модернизировать ли старые проблемы или выбрать новый формат", то правильный ответ будет "выбрать новый формат". Но в конкретном частном случае надо смотреть, что же именно модернизируется.

Протокол IPv6, каким бы он ни был хорошим, а он на самом деле не так и хорош, как может показаться, имеет одну проблему - тотальную несовместимость с сетями IPv4, с прежними программами и оборудованием для IPv4. Недопустимо вносить такую несовместимость (на уровне недоставки пакетов и невозможности адресации) без наличия веских причин, при которых продолжать нормальную работу в текущих условиях просто невозможно.

Говоря о тотальной несовместимости, мы имеем ввиду не только физическую возможность или невозможность доставки нового IPv6 пакета по сетям IPv4, но и сложность работ по модификации прежних IPv4 программ с открытым кодом на стороне пользователя, которые надо выполнить при введении нового IPv6 протокола.

Какие же цели достигаются при модификации IPv4 до IPv4r2 вместо полной его замены на IPv6?

Модифицируя IPv4 мы сохраняем полную совместимость с аппаратным и программным оборудованием, действовавшим до этого 20 лет, сохраняем так, что нет нужды его физически переделывать или сложность работ по модификации прежних IPv4 программ с открытым кодом на стороне пользователя минимальна.

Метод модификации IPv4 делает модифицированный протокол IPv4r2 "внутренне несовершенным", но это проблема от того, что исходный IPv4 изначально был внутренне несовершенным, чтобы его могли использовать на глобальном уровне (от осины не родятся апельсины) и новая версия IPv6, предложенная "олигархами", будет на глобальном уровне столь же убога как и IPv4, и эти проблемы IPv6 так же предсказуемы, как были предсказуемы проблемы IPv4.

В отличие от каких-то иных вещей, для такого протокола, как межсетевой протокол (протокол интернета) важно, чтобы он был принят всеми участниками, это вроде как признание золота в качестве ценности. Лучший способ этого добиться для IPv4r2 - реализовать такой IPv4r2 протокол на на тех компьютерных системах, которые используются пользователем так, чтобы пользователь мог его задействовать, если это ему интересно.

Политизированность и рыночность технического вопроса.

Я вовсе не политизирую технический вопрос, но действия разных "консорциумов и комитетов по стандартизации" за последние десятки лет, среди них такие действия как принятие форматов ATA, CD/DVD, USB, PCI, EFI твердо указывают на то, что за этими стандартами стоят вовсе не технические причины, а стандарты принимаются так, чтобы из них можно было бы извлекать прибыль, чтобы введением стандартов создавалась бы определенная проблема, ключи в решении которой были бы у авторов стандарта.

Права разработчиков конечно должны защищаться, но не ущерб же соединению между собой компонентов предназначенных для общей работы, т.е. контроль за соблюдением прав тех, кто вложил деньги в разработку стандарта, должен лежать на уровне производства и допуска лицензионного товара к продаже, а не в деятельности самих приборов. Разного рода соглашения между производителями могут легко нарушаться на уровне незаконной продажи и серых товаров.

В общем вопрос прав на интерфейсы не простой. Те производители, кто выполняет серьезную работу по формированию интерфейсов и рынка устройств с таким интерфейсом не хотят, чтобы другие производители просто пришли и стали делать совместимые копии. А пользователь не хочет, чтобы интерфейсы разных производителей были бы несовместимы между собой. Погоня производителя за защитой его прав приводит к серьезным техническим проблемам в работе устройств. Это в целом проблема некачественного общественного устройства, противоречивого и конфликтного, где извлечение прибыли как угодно и откуда угодно является единственной целью и смыслом деятельности каждого.

В общем те производители, кто не оплатил процесс создания общего интерфейса и формирования рынка потребителей такого интерфейса должны заплатить именно за это при использовании такого стандарта в своих изделиях. Оплата за общий стандарт не может быть использована для извлечения вечной прибыли для создателя стандарта или для конкурентной борьбы.

В целях борьбы с монополизмом производитель должен быть не вправе закрывать стандарты, не вправе лицензировать принципы, но в праве требовать от других участников рынка оплатить все его расходы по созданию этого стандарта, по формированию рынка для этого интерфейса или этого принципа, включая все его риски по провалу, которые у него были при создании такого рынка и он нес их в одиночку. Другими словами, быть вторым производителем, ждущим как пойдет дело у первого, не должно быть выгодно. Также первый производитель может иметь определенное время на торговое преимущество на новом рынке, но не более чем на 50% по объему продаж, чтобы быть первым было выгодно с точки зрения реализации продукции.

В итоге, с такими лицензиями все плохо. При социализме таких проблем нет.

Технические задачи IPv4r2.

IPv4 это такой протокол сетей, который позволяет адресовать компьютеры в сети. В глобальной сети интернет, имея только 32 бита на такой сетевой адрес, IPv4 исчерпал свои возможности по такой сетевой адресации. Также IPv4, инкапсулируя в себя фрагментацию, подсчет контрольных сумм и даже маршрутизацию, нарушает принцип разделения задач по уровням сетевого взаимодействия, подобным уровням сетевой модели OSI.

Задачей протокола IPv4r2, как сетевого протокола, является доставка пакетов, путем:

- адресации конечной и начальной точек сетевого пути;
- инкапсуляции доставляемых данных;

в этом смысле фрагментация и доставка фрагментов, подсчет контрольной суммы и маршрутизация всех пакетов не является задачей IPv4r2.

Для реализации каждой из таких функций для IPv4r2 протокола должен использоваться внешний специализированный протокол-обертка, сложность которого зависит от сложности решения этих задач на данном участке пути при доставке IPv4r2 пакета и в который пакет IPv4r2 должен быть вложен как обычные данные. Так что на простых путях доставки и в малых сетях такие внешние протоколы-обертки могут не понадобиться совсем.

Тем не менее, IPv4r2 позволяет прямо общаться с традиционными IPv4 сетями, допуская подсчет контрольной суммы, фрагментацию и доставку фрагментов, маршрутизацию по правилам IPv4 сети.

Таким образом для обеспечения этих противоречивых требований IPv4 и IPv4r2, протокол IPv4r2 может работать в двух основных режимах:

- в режиме совместимости с IPv4
  в котором обеспечивается только расширенная адресация IPv4r2 при полной совместимости с IPv4;
- в полном, основном режиме
  в котором обеспечивается полная оптимальная функциональность IPv4r2.

Для упрощения на стороне пользователя управления этими свойствами IPv4r2 для разных IP адресов разных типов, адреса традиционного IPv4 динамически могут отображаться на локальные адреса машины так, что при обращении в эту локальную IPv4 подсеть, протокол IPv4r2 работает в режиме совместимости с IPv4 и не задействует свойства IPv4r2, несовместимые с IPv4.

Предполагается, что IPv4r2 будет использоваться:

- как межсетевой протокол глобальной сети с малыми пакетами и большими IPv4r2 адресами;
- как сетевой протокол локальной сети с большими IPv4r2 пакетами и малыми IPv4 адресами;
- возможны обратные комбинации в случае специальных сетей.

Сложность модификации IPv4 устройства для обеспечения работы IPv4r2.

Работая в основном режиме, протокол IPv4r2 не является 100% совместимым с IPv4, поэтому могут возникать задачи модификации старых IPv4 устройств для поддержки ими работы IPv4r2 протокола в основном режиме.

Модификация модификации рознь. Рассматривая изменения в сетевых пользователях IPv4, которые нужно в них внести для обеспечения их совместимости с IPv4r2, мы будем изучать вопрос "какие изменения надо внести в программу обслуживания IPv4 с открытым исходным кодом для обеспечения совместимости с IPv4r2, надо ли при этом:

- менять ядро ОС, глобально и тяжело адаптируя новое ядро под имеющуюся аппаратуру;
- тяжело добавлять новые сетевые службы от нового ядра на старое ядро;
- легко добавлять новые фильтры на старые сетевые службы;

- легко менять константы и способы обработки полей опций в старых сетевых службах;
- выполнять т.п. простые работы;

оценивая все эти работы по сложности в модификации и в последующей отладке".

Например, протокол IPv4r2 имеет резервное значение 0xFFFF в поле контрольной суммы IPv4 заголовка. Для исправления программ, обслуживающих оборудование поддерживающее исходный IPv4, с целью внесения совместимости по интерпретации IPv4r2 поля контрольной суммы, изменения будут откровенно косметическими (по сравнению с ситуацией внедрения IPv6), даже "решение в лоб" путем:

- фильтрации всех входящих IPv4 пакетов с полем контрольной суммы 0xFFFF на предмет подсчета контрольной суммы;
- фильтрации всех исходящих IPv4 пакетов установкой поля контрольной суммы в значение 0xFFFF;

без учета эффективности, равной занесению правильных значений в момент создания IPv4r2 заголовка.

Так вот, в этом примере изменения для программ с открытым исходным кодом, связанные с использованием резервного значения 0xFFFF в поле контрольной суммы, будут очень простыми, для модификации не потребуется смена ядра, переделка всего стека протоколов и не нужны т.п. глобальные обновления.

Полная модификация IPv4 в IPv4r2 потребует создавать полностью новый IPv4r2 стек, как и для IPv6, но реализация функциональности IPv4r2 настолько схожа с IPv4r2, что реально новый IPv4r2 стек это будет копия функций старого IPv4 стека, часть IPv4 функций которого будут слегка, довольно незначительно, модифицированы.

Подсчет контрольной суммы IPv4r2.

IPv4r2 создан в предположении, что логические протоколы вообще не должны обременять себя контролем целостности при передаче данных (за исключением поддержки логических состояний соединения), хотя бы потому, что они не знают какова среда для передачи данных и какие сбойные факторы в этой среде действуют, таким образом физически передавая IPv4r2 пакеты вы должны обертывать его в такую контрольную обертку, сложность которой зависит от проблем с передачей данных в конкретном канале.

В этом плане подсчет контрольной суммы IPv4 это операция достаточно трудоемкая, достаточно ненадежная и в целом архаичная.

IPv4 реализует идею перенести все вычислительные нагрузки по контролю целостности пакетов на источник и адресат пакета, чтобы освободить от этого промежуточные узлы, но такая идея не может сработать в большой сети, состоящей из сегментов с разным качеством и надежностью передачи информации. Тем более что при маршрутизации пакета IPv4 эту контрольную сумму приходится корректировать и в общем говоря пересчитывать.

Заместо такого переноса вычислений, IPv4 фактически работает в предположении, что все участки сети, по которым будут

предаваться данные, должны быть не хуже, чем такие участки, при передаче по которым контроль целостности можно обеспечить подсчетом контрольной суммы как в IPv4. В результате для хороших участков сети это избыточно, а для плохих - недостаточно.

Контроль целостности при передаче данных это задача того уровня сетевого взаимодействия, на котором данные передаются физически. Пакеты IP должны получаться и отправляться на нижележащие сетевые уровни также, как происходит чтение секторов с контроллера диска (каким именно диском управляет контроллер пользователю неизвестно), при этом пользователь, получив сектор от контроллера диска, никакой контрольной суммы не вычисляет.

Например, подсчет контрольных сумм применяется для ПЗУ персонального компьютера и происходит разово, например при начальной загрузке, чтобы убедится что данные для канала связи, сохраненные в ПЗУ, целостные. Это необходимо потому что само ПЗУ исторически было таким, что при хранении данных могло иногда их терять. Если же источник предоставляет для передачи по каналу связи заведомо целостный пакет IPv4, а это требование всегда выполнено, если ОЗУ компьютера исправно, то контрольную сумму в приемнике вычислять нет нужды, качество передачи должно быть обеспечено каналом связи.

С точки зрения уровней сетевого взаимодействия должен быть отдельный от IP протокол, который позволяет накладывать разные алгоритмы контроля целостности при передаче IP пакетов на участке сети, например для выполнения подсчета контрольной суммы IPv4 этот протокол может иметь такой заголовок:
<протокол контроля=1>[8]<версия=1>[8]<контрольная сумма>[16]<размер пакета>[32]<любые данные>[размер пакета/8]

Поэтому IPv4r2 работая в основном режиме позволяет не использовать подсчет контрольной суммы, если это сделано на нижележащих уровнях.

Арифметика дополнения до единицы в IPv4.

Во многих доступных людям ПК встретить "арифметику дополнения до единицы" (в наших терминах "арифметику обратных кодов") не просто. Поэтому вопрос "что это такое и зачем она нужна" сразу возникает.

Можно сказать, что если суммировать так, что возникающий от двоичного суммирования перенос затем прибавляется в младший разряд суммы, то вся сумма из-за переполнения деградировать в ноль никогда не сможет.

- Второй раз перенос при суммировании результата с битом переноса не возникает. Если самое большое для данного числа бит число, например FF для 8 бит сложить с самим собой (с другим самым большим числом), это все равно что умножить FF на два или сдвинуть FF влево на один разряд и получить 1FE, в результате самый правый бит даже у самой большой суммы всегда чистый, и если перенос прибавить к младшему разряду, то второй раз переполнение возникнуть не сможет.

- Раз битов хватило для представления суммы самых больших чисел и их переноса, то если складывать числа меньше самых

больших, то второй перенос тем более не может возникнуть. Это можно увидеть, заметив что в числе меньшем FF есть хоть один нулевой бит, а значит на этом нуле при суммировании единицы процесс второго переноса и остановится.

Сумма при таком суммировании конечно тоже деградирует, но в некоторое ненулевое значение. Это позволяет резервировать значения суммы 0 для каких-то еще целей, никаких иных особых преимуществ сложение с собственным флагом переноса, помещенным в младший разряд слагаемого, кажется не сулит.

Зато это сулит проблемы от того, что нам надо не только складывать, но и вычитать, чтобы вычислять небольшие изменения в контрольной сумме от поля TTL и т.п., не пересчитывая всю сумму полностью.

Суммируя с переносом мы попадаем в арифметику обратных, а не дополнительных кодов, поэтому чтобы вычислить разность, надо представлять вычитаемое как отрицательное число в обратном коде (командой NOT) с последующим сложением этого отрицательного числа в обратном коде с переносом в младший разряд, это автоматически компенсирует возможное добавление переноса к младшему разряду в предыдущих суммированиях.

Арифметика обратных кодов получается, если условно взять старший бит за знак, а инверсией бит положительные числа менять на отрицательные и наоборот т.е. разбить беззнаковые числа на диапазоны

- 01-7F
- 00
- FF
- FE-80

Инверсией бит мы добиваемся, что сложив положительное и отрицательное числа равные по модулю мы получаем -0 (FF)
В обратном коде основной ноль это -0 (FF), а двоичный ноль +0 (00), получается только при вычитании числа -0 и при константном задании +0 в качестве аргумента.

По кодам видно, что в арифметике обратных кодов прибавление переноса в младший разряд иногда получается когда хотя бы одно из чисел, которое мы суммируем, является в этом коде отрицательным (у одного включен бит в старшем разряде) и всегда получается если складываем два отрицательных числа (у обоих включен бит в старшем разряде).

Код называется обратный, потому что ряду положительных 00, 01, 02 чисел через +1 соответствует ряд отрицательных FF, FE, FD через -1, т.е. числа в отображении как бы выставлены в обратном порядке возрастания, поэтому при такой сортировке и положительные и отрицательные числа отстоят друг от друга на +1, а также оказываются битовой инверсией по отношению друг к другу

- 00 - FF
- 01 - FE
- ...

- 7F - 80

В обратном коде отрицательные и положительные числа тоже получаются путем перехода на +1 и -1, но можно видеть, что в обратном коде отрицательные числа сдвинуты относительно двоичного 0 (+0) на -1, поэтому арифметика привычного дополнительного кода как N повторов +1 и -1 для них невозможна и после арифметических операций с отрицательными числами, в зависимости от результата операции, может быть необходима коррекция на +1.

Например, складывая в обратном коде -1 и -1, смещенные в численном виде на (-1), имеем
- -1(-1) + -1(-1) = -2(-2) = -4
- реальный результат -2(-1) = -3
- надо прибавить +1

- 1(+0) + -2(-1) = -1(-1)= -2
- реальный результат -1(-1) = -2
- не надо прибавлять +1

- 2(+0) + -1(-1) = 1(-1)= 0
- реальный результат 1(+0) = 1
- надо прибавить +1

Для простоты реализации обратного кода прибавление +1 берется от флага переноса всегда, если он установился после суммирования, даже если с точки зрения арифметики есть переполнение. По этому правилу суммируя в обратном коде (с учетом переноса) и в случае появления переполнения для отрицательных чисел мы добиваемся симметрии для последующего добавление обратного положительного числа и устранения переполнения. Это качество используется и в расчете модификации контрольной суммы IPv4 путем вычитания старого и добавления нового числа, при этом как раз важно, чтобы операции сложения и вычитания одного и того же числа были бы симметричны по отношению к переполнению.

Фрагментация IPv4r2.

Если попытаться обойтись без фрагментации IPv4 пакетов, то при обмене данными с помощью IPv4 источник должен ограничить размер пакета самым узким участком всей сети, посылая по широким участкам сети множество мелких пакетов.

Протокол IPv4 предполагает, что фрагментация пакетов это вещь достаточно универсальная, известная на стороне приемника и передатчика, так что при передаче IPv4 пакетов, также как для подсчета контрольной суммы, применяются средства самого IPv4 для обеспечения такой фрагментации.

Значит качество обеспечения универсальной фрагментации в протоколе IPv4 казалось бы могло быть лучше, чем универсальная защита целостности заголовка IPv4 путем подсчета контрольной суммы, но в реальности это не так. Потому что фрагментация силами IPv4 не только усложняет заголовок IPv4 пакета, но также усложняет саму маршрутизацию, которая вынуждена работать с IPv4 фрагментами.

Какие же проблемы при маршрутизации IPv4 фрагментов? Заметим, что по сути при фрагментации на протокол IPv4 возложена задача протокола TCP по передаче и сборке фрагментов, при том что фрагменты при IPv4 фрагментации передаются по правилам UDP. В результате огромный IPv4 исходный пакет может быть утрачен при утрате единственного фрагмента этого пакета, при том что сеть, передающая эти фрагменты, выполнила 99,9% процентов всей работы по передаче этого IPv4 пакета.

Таким образом, опять, как и при подсчете контрольной суммы, оказывается что маршрутизация фрагментов зависит от проблем на конкретных участках сети, о которых IPv4 адресаты ничего не знают и протокол IPv4 не может предоставить средства для правильной фрагментации.

Поэтому IPv4r2 работая в основном режиме предполагает, что фрагментацией, пусть и универсальным образом, должен заниматься отдельный протокол на проблемных участках канала передачи данных, аналогично протоколу для контроля целостности данных, внутрь которого будет вложен IPv4r2 пакет.

Перечислим пару основных путей обеспечения фрагментации IPv4r2, поскольку это в общем менее известно, чем разные способы помехо- защищенного кодирования:

- фрагментация правилами UDP с маршрутизацией (используется в IPv4, применяется для надежных сетей);
- фрагментация правилами UDP без маршрутизации на пути точка-точка (применяется на надежных участках сети);
- фрагментация правилами TCP с маршрутизацией фрагментов (применяется в обычных сетях).

Интерес представляет именно третий вариант, когда узел, вынужденный выполнить маршрутизацию, просто использует TCP протокол, внутри которого вложен IPv4r2 пакет, устанавливая канал связи до того маршрутизатора, который сможет передавать исходный IPv4r2 пакет, в лучшем случае этот маршрутизатор знает о размере своей сети со этим MTU и пробросит исходный IPv4r2 пакет от одного конца своей сети до другого в виде фрагментов, в худшем случае, конечной точкой маршрута станет сам получатель IPv4r2 пакета и начиная с узкого места в сети в адрес получателя IPv4r2 пакета пойдут фрагменты исходного пакета, упакованные в TCP сегменты.

Естественно, что размер IPv4r2 пакета, который может быть передан промежуточным маршрутизатором хотя бы в виде фрагментов, не может быть каким угодно большим. В общем случае IPv4r2 источник должен заранее знать IPv4r2 пакеты какого размера могут быть доставлены до конкретного IPv4r2 назначения по конкретному маршруту.

Для решения вопроса гарантированной, в смысле подходящего размера IPv4r2 пакета, доставки IPv4r2 пакета есть два

способа:

- задание максимально размера IPv4r2 пакета пригодного для передачи по любым IPv4r2 сетям (межсетевая работа в интернете);
- поиск максимального размера IPv4r2 пакета для доставки по заданным путям до заданного адресата (работа в специальных и локальных сетях).

Поскольку IPv4r2 это межсетевой протокол, работа в специальных и локальных сетях для него не основная и предполагаемый практический способ установления максимального размера пакета в таких специальных и локальных IPv4r2 сетях, это задание максимального размера IPv4r2 пакета администратором этой сети в ручную.

MTU 1152 в сетях IPv4r2 для межсетевой работы.

Для протокола IPv4 для межсетевой работы установлено требование гарантированной передачи на каждом участке сети без фрагментации IPv4 пакета максимального размера 576 байт (MTU 576), состоящего из:

- 512 байт полезных данных в пакете;
- 64 байт вспомогательных данных управления передачей, включающих в себя заголовки IP/UDP/TCP;

т.е. в IPv4 маршрутизаторах для каждого IPv4 пакета 64 байта максимум отводится на хранение управляющей информации и 512 байт максимум отводится на хранение данных (если 64 байт для хранения управляющей информации не хватает, то что не поместилось в 64 занимает место в блоке из 512 байт, затрудняя обмен данными блоками по 512 байт), в IPv4 на 8 порций пакета идет 1 порция управляющей информации (всего 9 порций по 64 байта), т.е. 1/9=11% управляющей информации.

Пакеты IPv4 большего размера могут потребовать фрагментации и попадают в разряд работы в специальных и локальных IPv4 сетях.

IPv4r2 оперирует б`ольшими адресами, которые не всегда помещаются в 64 байта вместе с опциями и прочей управляющей информацией, поэтому размер передаваемых в IPv4r2 пакете данных имеет все шансы стать менее 512 байт, чтобы передаваться по сетям IPv4 без фрагментации. Поэтому, а также чтобы сохранить и для IPv4r2 соотношение в 11% управляющей информации, максимальный размер пакета IPv4r2, который должен гарантированно передаваться в сетях IPv4r2 без фрагментации, удваивается, т.е. это максимум 1152 байта в пакете (MTU 1152):

- 1024 байт на полезные данные;
- 128 байт на управляющую информацию.

IPv4r2 работая в основном режиме для межсетевой работы использует MTU 1152. Пакеты IPv4r2 большего размера могут потребовать фрагментации и попадают в разряд работы в специальных и локальных IPv4r2 сетях.

Заголовок IPv4r2 с максимальным размером 80 байт.

Расширение адресации IPv4r2 в среднем требует дополнительно 20 байт для заголовка IPv4r2. Чтобы вписаться в 11% на управляющую информацию и в 128 байт, оставив 48 байт на заголовок UDP/TCP, в основном режиме работы IPv4r2 существует расширение размера заголовка IPv4r2 на 20 байт, так что максимальный размер IPv4r2 заголовка станет равен 80 байтам и при этом функциональность IPv4r2 заголовка по опциям будет не хуже чем для IPv4.

Включение режима максимальный размер 80 байт для заголовка IPv4r2, автоматически включает режим максимальный размер 48 байт для:

- заголовка IPv4 протоколов UDP/TCP в стеке протоколов IPv4r2;
- заголовка IPv4r2 протоколов UDPr2/TCPr2.

Превышение размера IPv4r2 заголовка при использовании IPv4 опций контролируется также как и для IPv4, что дает возможность передавать 1024 байт полезных данных без риска IPv4r2 и UDP/TCP заголовкам мешать этой области. При включении опций IPv4r2 для увеличения размера IPv4r2 заголовка на произвольную величину, контроль расхода 128 байт на управляющую информацию лежит на пользователе.

Расширение размера IPv4r2 заголовка до 80 байт предназначено:

- для межсетевой работы IPv4r2 (при работе в интернет), когда имеет значение MTU 1152;
- чтобы давать больше гибкости в использовании опций IPv4.

Расширение размера заголовка IPv4r2 до 80 байт не предназначено:

- для замены сложных протоколов маршрутизации или отладки сети опциями заголовка IPv4, эти опции IPv4 имеют ограниченное применение для особенных случаев.

Если надо доставлять IPv4r2 заголовок по специальному маршруту или заниматься отладкой сети, то IPv4r2 заголовок должен быть обернут в транспортный или отладочный протокол для такой маршрутизации или отладки, при этом исходный IPv4r2 заголовок, содержащий только конечные адреса маршрута, становится протоколом более высокого уровня.

Фрагментация IPv4r2 пакета флагами IPv4.

Пакеты IPv4r2 с IPv4 флагом запрета фрагментации не должны использовать поля идентификатор и смещение, устанавливая их в 0, это технические поля которые нужны только при фрагментации и которые имеют значение только между теми хостами, которые передают между собой фрагменты. Незнакомые друг с другом конечные IPv4 адресаты точно не могут знать о каких то магических значениях поля идентификатор, а метки потоков и дополнительная адресация задаются отдельными опциями IPv4r2 или протоколами более высокого уровня.

IPv4 флаг запрет фрагментации должен устанавливаться по умолчанию для пакетов размером меньше максимально допустимого MTU, т.к. нет никаких причин для протокола более высокого уровня отсылать в неизвестный адрес пакеты большие, чем максимально допустимый для отсылки без фрагментации, т.к. какой бы размер фрагмента IPv4r2 не был взят, данные в протоколе более высокого уровня все равно придется фрагментировать по некоему размеру блока и нет никаких причин, кроме попыток уменьшить нагрузку на сеть устранением лишних заголовков, чтобы не взять этот размер блока сразу корректным для IPv4r2.

Сетевые протоколы IPv4r2 очень высокого уровня должны работать с блоками полезных данных не более 512 байт, тогда они не будут в нормальном случае генерировать фрагментацию. Поскольку протоколы могут быть вложенными и в UDP/TCP, то IPv4r2 с MTU равным 1152 позволяет оборачивать 512 байт реальных данных довольно большое количество раз.

Для работы именно в сетях IPv4r2 приложение не перегружающее данные заголовками может ориентироваться на блоки полезных данных по 1024 байта с целью уменьшения затрат сети на служебные данные, но работа без задержек в физических сетях с общей средой передачи и множеством пользователей все равно возможна только с малыми пакетами полезных данных по 512 байт.

Отсылать пакеты IPv4r2 больших размеров можно только в локальных сетях, в специальных глобальных сетях и в локальных соединениях точка-точка, для локальных связей точка-точка IP протокол явно не нужен и используется там только для уменьшения числа используемых протоколов и для однообразия работы со всеми машинами в любых сетях.

Асимметричная адресация IPv4r2 (77 бит).

Главной проблемой исчерпания IPv4 адресов для пользователя (и главной основой для возникновения торговли IPv4 адресами и некоторыми видами хостинга) является то, что пользователи не могут выставлять в сеть серверные ресурсы, поскольку другие люди эти ресурсы не могут адресовать. Сами пользователи могут легко выходить на "крупные" сервера, которые имеют IPv4 адреса, через шлюзы провайдера. Ситуация в чем то аналогичная ADSL, по асимметрии.

Отметим, что работа через шлюз заставляет провайдера создавать вычислительные ресурсы маршрутизатора, которые занимаются трансляцией IPv4 адресов, но опять же, проблема тут только в протоколе IPv4, который всеми был в свое время принят, вот за это и расплата. Теперь от IPv4 взять и просто отказаться уже нельзя.

По большому счету вопрос трансляции IPv4 адресов можно считать у провайдера уже решенным - оборудование для трансляции IPv4 адресов уже установлено. Также вычислительная нагрузка на шлюзы провайдера:

- по маршрутизации IP пакетов без их модификации;
- по физической ретрансляции пакетов в сетях одного типа;
- по необходимости в передаче пакетов между разными физическими типами сетей;

в любом случае останется, даже при IPv6.

Отметим, что некоторым пользователям вообще не понадобится реальный IPv4r2 адрес для предоставления доступа к своим компьютерам из сети, потому что у них нет серверных ресурсов.

Пока провайдеры имеют ресурсы обеспечивать клиента динамическими IPv4 адресами, нам выгодно рассмотреть асимметричную схему адресации IPv4r2 сетей, потому что:

- размер заголовка IPv4 ограничен и много- битовая адресация уменьшает функциональность IPv4 сетей;
- много- битовая адресация всегда ведет к росту накладных расходов при передаче полезных данных малыми пакетами;
- нам для установления связи надо уметь адресовать только сервера в глобальных IPv4r2 сетях.

Равноправное общение двух IPv4r2 адресов при асимметричной адресации происходит по "кольцевой" схеме с участием IPv4 шлюза провайдера:
- исходящие двух- или однонаправленные соединения с первого IPv4r2 адреса идут через динамический IPv4 его (первого) провайдера на полный IPv4r2 адрес второго IPv4r2;
- при необходимости установить ответное исходящее соединение, второй IPv4r2 адрес через динамический IPv4 уже своего (второго) провайдера обращается на полный IPv4r2 адрес первого IPv4r2.

Расширенная адресация IPv4r2 для шлюзов (индексы локальных сокетов).

Протоколы UDP/TCP имеют собственную, независимую от IP, адресацию, которая называется портами (адрес каждого порта это число, как адрес байта памяти, например, порт 100). По сути порт UDP/TCP это аналог аппаратного порта ввода-вывода для архитектур компьютеров времени создания UDP/TCP.

В порт ввода-вывода можно писать данные или читать из него, а то что будет при этом происходить зависит от того, как эти порты соединены физическими проводами снаружи компьютера между собой или с другими компьютерами и прочими внешними устройствами.

Порты UDP/TCP делают то же самое, если их заместо физических проводов программно соединить с другими портами. Когда такое соединение сделано, и по протоколу UDP/TCP предаются данные, можно говорить о канале передачи данных типа точка-точка.

Чтобы соединить между собой порты UDP/TCP на разных компьютерах, каждый такой компьютер должен иметь свой сетевой адрес, например, IPv4, NetBIOS и т.д. Один или более адресов от одного или более протокола, нужных совместно для осуществления успешной связи, в данном случае это "сетевой адрес:порт UDP/TCP", называется сокетом. Пусть такие соединяемые компьютеры имеют IPv4 адреса, тогда сокет это пара адресов "IPv4:порт".

Каждый UDP/TCP канал передачи данных, имеющий тип точка-точка, можно описать парой сокетов на концах такого соединения:

источник и назначение.

При трансляции IPv4 адресов шлюзом провайдера, каждому реальному IPv4 адресу и порту шлюза ставится в соответствие локальный IPv4 адрес и порт источника и IPv4 адрес и порт назначения.

источник- локальный IPv4: локальный порт
шлюз провайдера- реальный IPv4: реальный порт
назначение- назначение IPv4: назначение порт

Т.е. для компьютера назначения все выглядит так, словно с ним работает шлюз от имени своего реального IPv4 адреса и порта, хотя на самом деле шлюз отправляет все приходящие внешние данные на локальный IPv4 адрес и порт источника и в обратную сторону.

В терминах сокетов, каждому сокету шлюза ставится в соответствие UDP/TCP канал передачи данных
 сокет шлюза:
  UDP/TCP канал передачи данных:
   сокет источника
   сокет назначения

Чтобы шлюз мог отправить обратный IPv4 пакет от "назначение IPv4: назначение порт", который приходит на "реальный IPv4: реальный порт", в правильный "локальный IPv4: локальный порт", шлюз должен понять кому в локальной сети шлюза на деле адресован пакет, приходящий на "реальный IPv4: реальный порт" шлюза. Сокет шлюза должен определить какой UDP/TCP канал передачи данных используется этим IPv4 пакетом, пришедшим от сокета назначения.

Шлюз в малой локальной сети.

а) Это можно решить так, как делается в малой локальной сети:
каждому "локальный IPv4: локальный порт" шлюз динамически выделяет отдельный реальный порт шлюза по запросу от источника, т.е. с каждым "реальный IPv4: реальный порт" шлюза связан только один:
"локальный IPv4: локальный порт"

Этим на локальной половине исходного канала "сокет источника - сокет назначения" создается уникальная связь "сокет источника - порт шлюза", так что внешние сокеты не могут адресовать, т.е. физически не могут указать какие-либо иные порты локального источника, кроме того что связан с этим сокетом источника. Сокет же источника может адресовать любой сокет назначения через этот порт шлюза.

Тогда если ответные данные пришли на этот "реальный IPv4: реальный порт" от любого внешнего адреса, они могут быть

правильно отправлены на "локальный IPv4: локальный порт". И наоборот, данные пришедшие от "локальный IPv4: локальный порт" могут быть правильно отправлены на любой внешний сокет.

Проблема тут в том, что адресация UDP/TCP портов всего лишь 16 битная. Для одной машины этого может и хватает, но вот для шлюза провайдера, обслуживающего всех его клиентов, этого явно мало, поскольку клиентов больше, чем 60 тысяч.

На самом деле даже один клиент может открыть десятки, сотни исходящих портов одновременно и для каждого из них шлюз провайдера должен выделить отдельный уникальный порт шлюза, т.е. одним IPv4 адресом провайдера такого шлюза реально сможет пользоваться только около тысячи обычных клиентов одновременно, что хватает для малых локальных сетей, не не подходит для провайдера.

б) Таким образом, для обеспечения лучшей работы IPv4 шлюзов, в том числе защитных экранов для локальных сетей, нужно расширение адресации сокетов, а поскольку сетевой шлюз предполагает единственный сетевой адрес (IPv4 ли это адрес или IPv6 адрес - это неважно), то расширять можно только адреса портов.

Есть два пути расширения адресации портов:
- расширять адрес порта UDP/TCP, превратив их из 16 битных в 24 или 32 битные;
- добавить индексы локального сокета шлюза для сетевого адреса IPv4, специально чтобы поддержать шлюзы.

Индексы локального сокета шлюза.

Добавить индексы шлюза для сетевого адреса IPv4 выгодно потому, что тогда обычный шлюз сможет работать только на сетевом уровне IPv4, независимо от типа протокола UDP/TCP или иного протокола, что и сделано в протоколе IPv4r2. Поэтому получается IPv4r2 сокет такого вида: "сетевой адрес IPv4r2:индекс локального сокета шлюза".

Теперь IPv4r2 шлюз каждому клиенту с сокетом "локальный IPv4: локальный индекс шлюза" ставит в соответствие сокет шлюза "реальный IPv4: реальный индекс шлюза", независимо от того какой протокол и какие порты используются внутри IPv4r2 пакета.

Для того чтобы это сработало и шлюз и назначение должны понимать что такое "индекс локального сокета шлюза". Если IPv4r2 сервер назначения не понимает "индекс локального сокета шлюза", то шлюз должен работать с таким назначением по старой схеме, с анализом портов UDP/TCP, поэтому если шлюз не анализирует протоколы UDP/TCP, то может отвергать пакеты с индексом 0 приходящие от IPv4r2 назначения на IPv4 шлюза, поскольку такой шлюз никогда не назначает своих локальных клиентов на индекс 0 (под таким индексом работает любой клиент и сам шлюз как клиент, не перенаправляя данные от своего локального клиента).

Индекс сокета шлюза не нужен для адресации пользователем и не входит в URL запроса к серверу, он создается динамически

на шлюзах и используется только IPv4r2 сервером и IPv4r2 шлюзом. Теоретически локальный компьютер-источник может (предполагаемо в URL IPv4r2 его можно указать так "протокол://адрес:UDP/TCP порт:IP индекс"), но должен воздерживаться от того, чтобы использовать индексы сокета шлюза (всегда используя индекс сокета 0). Перегруженный шлюз вправе отбросить пакеты от клиента с индексом локального сокета не равным 0, если по мнению шлюза этот клиент сам не может быть шлюзом.

Шлюз работающий на сетевом уровне:

- может назначать индексы сокета шлюза своим клиентам динамически при каждой новой авторизации пользователя или первом после тайм- аута исходящем от этого клиента IPv4 пакете (при этом на практике для достаточно непрерывно посылающего пакеты клиента будет создан постоянный на каждый непрерывный сеанс авторизации IPv4r2 канал, с постоянным индексом сокета шлюза связанным с локальным IPv4 этого клиента);
- может разделять один индекс сокета шлюза между несколькими клиентами, отсылая входящие на этот индекс сокета пакеты всем клиентам, для которых этот индекс сокета назначен;
- может проверять, что адрес назначения тот, который был до истечения тайм- аута указан в исходящих запросах клиента.

Индекс сокета шлюза, работающего на сетевом уровне, это просто расширение IPv4 адресации, которое позволяет подключать больше компьютеров локальной сети к одному IPv4 адресу, но это такое малое расширение IPv4 адресации, которое в общем требует меньше места в IPv4 заголовке для хранения расширения, например 16 битные индексы займут в IPv4 заголовке минимум 2 и максимум 4 байта и то не на всех участках сетевого маршрута. Размер индекса произвольный, разумные значения: 16, 32, 48 бит на индекс, что потребует в IPv4 заголовке максимум 8 байт.

Также индекс сокета позволяет шлюзу работать в режиме IPv4 моста, позволяя через индекс сокета адресовать все UDP/TCP порты локального компьютера связанного с этим сокетом.

Индекс сокета шлюза, работающего на уровне UDP/TCP по схеме (а) шлюза в малой локальной сети, эквивалентен расширению адресации UDP/TCP портов. Поскольку это расширение адресации UDP/TCP будет прозрачно для UDP/TCP пользователей, не затрагивает локальные машины и динамическое (зависит от загрузки шлюза), это более гибкая система расширения адресации UDP/TCP, если шлюз необходим, поскольку шлюзы служат не только для выхода в корневую IPv4 сеть из локальных сетей, но и для защитной фильтрации трафика, не допуская:
- входящие соединения на компьютеры локальной сети;
- входящие соединения с недопустимых адресов на компьютеры локальной сети;
- входящие пакеты с иных адресов назначения, не указанных в исходящем запросе UDP/TCP от компьютера локальной сети;
- и т.д.

Для сервера реализующего UDP/TCP в стеке протоколов IPv4r2, сокет IPv4r2 имеет три поля:

- сетевой сокет

    - сетевой адрес
    - индекс локального сокета шлюза
- порт UDP/TCP

Адресация подсетей IPv4r2.

В наследство от IPv4 протоколу IPv4r2 достались:

- подсети вида "IPv4 адрес нулевого хоста подсети"/"число бит маски подсети", так что в нулевых битах маски сети у IPv4 адреса могут быть любые комбинации битов, определяющие уникальные адреса хостов этой подсети, а применение маски сети к любому IPv4 адресу подсети дает "IPv4 адрес нулевого хоста подсети";
- широковещательные IP запросы к такой подсети на выделенный IPv4 адрес, у которого все биты в поле хоста подсети установлены в 1 (это битовая инверсия маски подсети);
- назначение сетевому интерфейсу работы быть шлюзом до хостов этой подсети, путем указания выделенного IPv4 адреса, у которого все биты в поле хоста подсети установлены в 0 (это маска подсети примененная к IPv4 адресу любого хоста подсети).

Выделение нескольких IPv4 адресов на эти служебные цели создает некоторые трудности, как при формировании малых IPv4 подсетей и IPv4 подсетей сложной структуры, так и с уникальностью адресации хостов в подсетях.

Поэтому в сетях IPv4r2 подсети могут формироваться иным путем, для этого вместо выделенного адреса применяются IPv4r2 опции, которые кодируют:

- число бит маски подсети для IPv4r2 адресата;
- широковещательный запрос для IPv4r2 адреса назначения;

а для обозначения назначения сетевому интерфейсу быть шлюзом до хостов подсети используется запись "IPv4r2 адрес нулевого хоста подсети"/:"число бит маски подсети".

Таким образом, в сетях IPv4r2, нулевой и последний IPv4r2 адрес такой подсети может быть использован как адрес хоста. Также один IPv4r2 адрес хоста может принадлежать одновременно разным IPv4r2 подсетям, но этот IPv4r2 адрес в обычном случае должен принадлежать только одному интерфейсу, т.е. два разных интерфейса не могут отличаться только маской подсети.

Из соображений эффективности кодирования маска сети может задаваться не как число единиц от начала, а как число нулей от конца - обратная маска, обозначение: /:-"число бит обратной маски подсети".

В сетях IPv4r2 указание маски подсети в заголовке IPv4r2 обычно требуется только при отправке широковещательных IPv4r2 запросов в эту подсеть.

IPv4r2 адреса в IPv4 опциях маршрутизации и отладки.

Ряд опций IPv4 для маршрутизации и отладки предполагают хранение в IPv4 заголовке IPv4 адресов. Поскольку адреса IPv4r2 много- уровневые и обычно занимают 9 или 10 байт, то записи маршрутов и адресов в сети в виде IPv4r2 адресов это довольно нерациональная операция с учетом MTU 1152.

Если запись адресов такого рода (опциями IPv4) нужна, то все маршрутизаторы IPv4r2, которые отмечаются в IPv4r2 заголовке, должны иметь адреса или в корневой IPv4 сети, или в локальной IPv4 сети для каждого предыдущего IPv4r2 маршрутизатора (для фиксированной маршрутизации это условие выполнить легче всего), поэтому эти IPv4 опции не подвергаются в IPv4r2 модификации - они хранят локальные адреса в IPv4 сетях.

Из назначения "межсетевого протокола" следует, что корневая сеть IPv4 должна прямо адресовать только шлюзы корневой сети, а адресация машин или подсетей, скрытых за этими шлюзами, должна достигаться внутренними протоколами, которые способны сформировать многоуровневый в терминах IPv4 адрес.

В версии "межсетевого протокола" IPv4r2, этот многоуровневый IPv4r2 адрес машин и подсетей хранится в целом аналогично опциям маршрутизации IPv4 заголовка, так что протоколы верхнего уровня часто смогут могут работать в IPv4r2 сетях также как и в корневой сети IPv4.

Если нужна более сложная, глобальная маршрутизация или маршрутизация с указанием адресов IPv4r2, нужно использовать специальный протокол маршрутизации заместо IPv4 опций внутри IPv4r2 заголовка.

Эффективность использования IPv4r2 адресов в IPv4 опциях.

Для определения IPv4r2 адресов, маршрутизатор в сетях IPv4r2 должен проводить просмотр и анализ списка IPv4r2 опций, что в принципе дает значительное пенальти в виде потери производительности таких систем по сравнению с системами, где адрес указан в фиксированном поле.

Для решения этой проблемы и для улучшения внутренней структуры IPv4r2 заголовка, при использовании IPv4r2 опций расширения адреса, IPv4r2 адресаты по возможности помещают адресные IPv4r2 опции основной адресации (базовой или обобщенной) по фиксированному смещению 20 относительно начала заголовка, т.е. расположение больших адресов в этом формате IPv4r2 заголовка становится фиксированным, при этом сам заголовок остается полностью совместимым с IPv4.

При использовании основных схем кодирования IPv4r2 адресации, расширение адресов и индексы шлюза кодируются единственной опцией opX, что дополнительно удовлетворяет этим требованиям и если сравнивать такой IPv4r2 пакет с идеальным IPv4 пакетом без опций, то сканирование единственной опции не сильно отличается от фиксированной позиции в заголовке.

Заметим, что если опции присутствуют в IPv4r2 заголовке, то эти опции в общем все равно придется сканировать (ну, может

быть не в момент определения IPv4r2 адресов), так что запись адреса в опции это не такая и большая утрата производительности на самом деле.

Также если информация, которая записана в опциях, существенна для маршрутизации, то по большому счету это все равно, записана ли эта информация в виде списка опций или по фиксированным смещениям, особенно если запись по фиксированным смещениям в принципе невозможна.

При трансляции IPv4r2 пакетов по сетям IPv4, порядок опций заголовка в промежуточных IPv4 маршрутизаторах может меняться, поэтому каждый IPv4r2 маршрутизатор, обнаружив неправильный порядок IPv4r2 опций, может восстановить эффективное расположение опций при трансляции IPv4r2 пакета в следующую подсеть, таким образом дополнительное падение производительности из-за перестановки опций будет только при трансляции IPv4r2 пакетов через "плохие" с точки зрения IPv4r2 сетей маршрутизаторы IPv4.

Алгоритм поиска расширения адреса в опциях может быть таким:

- просмотреть список опций входящего IPv4r2 пакета и в зависимости от целей маршрутизатора:
  - построить IPv4r2 псевдо- заголовок из важных полей (адресных например);
  - восстановить правильный порядок опций в ретранслируемом IPv4r2 пакете, если этот порядок был нарушен.

В каких-то случаях маршрутизатор, проверив во входящем IPv4r2 пакете опции, найдя правильный порядок опций и обнаружив по смещению 20 единственную базовую кодировку IPv4r2 (все остальные опции могут быть для этого маршрутизатора не обслуживаемыми), может отправить такой IPv4r2 пакет на обработку своему высокопроизводительному коду, который будет считывать базовую адресацию непосредственно из IPv4r2 заголовка. Этот случай в целом мало чем отличается от фиксированных смещений IPv6.

Опция дополнительной IPv4r2 адресации пользовательской сети должна быть задана второй, сразу после задания основной адресации (базовой или обобщенной). Опция IPv4r2 описания подсетей должна быть задана третьей, сразу после адресации пользовательской сети. Эти опции могут не использоваться промежуточными IPv4r2 маршрутизаторами, их анализируют только конечные IPv4r2 адресаты.

При восстановлении порядка IPv4r2 опций в промежуточных маршрутизаторах, дополнительные IPv4r2 опции, анализ значений которых требует сложной интерпретации их битовых полей, могут просто помещаться после опций основной адресации, декодирование которых простое, в начало IPv4r2 заголовка.

Полное описание расширения IPv4 до IPv4r2, тонкая настройка IPv4r2.

Адресация IPv4r2.

-) Многоуровневый режим IPv4r2 адресации.

По отношению к 128 битным адресам IPv6, 128 битный адрес IPv4r2 имеет четырех- уровневую IPv4 иерархию:

a1.b1.c1.d1/a2.b2.c2.d2/a3.b3.c3.d3/a4.b4.c4.d4

Все основные режимы IPv4r2 адресации (77 бит, 80 бит и 72 бита) отображаются на 128 битный адрес четырех- уровневой иерархии IPv4 адресов по десяти- байтовой схеме 4/2/3/1 так:

a1.b1.c1.d1/c2.d2/b3.c3.d3/d4

специальным образом заполняя эти байты (см. описание основной IPv4r2 адресации далее).

А обобщенное IPv4r2 расширение адресации отображается на 128 битный адрес четырех- уровневой иерархии IPv4 адресов иначе:

- биты поля расширения адреса от старшего бита к младшему биту образуют байты;
- байты располагаются в 128 битном IPv4r2 адресе по уровням, полностью заполняют один уровень, без пропусков, только потом переходят к следующему уровню;
- первыми в поле расширения адреса идут байты старших уровней, старшинство убывает с возрастанием номеров уровня (второй, третий, четвертый);
- в пределах одного уровня байты идут от младшего к старшему: dX, cX, bX, aX;
- для асимметричной адресации 70 это выглядит так
  a1.b1.c1.d1/a2.b2.c2.d2/a3.b3.c3.d3/d4
  d4[6] = поле <младший байт>[6],
  <d2>[8]<c2>[8]<b2>[8]<a2>[8]<d3>[8]<c3>[8]<b3>[8]<a3>[8] = поле <адрес>[64]
- для асимметричной адресации 38 это выглядит так
  a1.b1.c1.d1/a2.b2.c2.d2/d3
  d3[6] = поле <младший байт>[6],
  <d2>[8]<c2>[8]<b2>[8]<a2>[8] = поле <адрес>[32]

В асимметричном режиме обобщенном IPv4r2 расширении адресации, поле младший байт адреса (d2 для 6 бит), если не равно нулю, то выравнивается по границе _младшего_ бита, заполняя старшие биты неполно адресуемого младшего байта нулями. Если поле младший байт адреса равно нулю, то оно считается не используемым и не входит в число байт длины адреса (см. ниже сравнение адресов).

-) Сравнение IPv4r2 адресов.

Адреса IPv4r2 в основном и обобщенном режимах IPv4r2 адресации принадлежат одному и тому же основному IPv4r2 адресному пространству. Адреса в этом основном IPv4r2 адресном пространстве имеют длину, выражаемую в целых байтах. При сравнении таких адресов равными могут быть только адреса с равной длиной, независимо от того, как совпадают их битовые поля на участках общей длины для адресов разного размера. Это дополнительно увеличивает адресное пространство IPv4r2, которое независимо для IPv4r2 адреса каждой длины.

Адреса IPv4r2 в режиме дополнительной IPv4r2 адресации сети пользователя образуют отдельное от основного IPv4r2 адресное пространство, которое также имеет длину адреса. Сравнение дополнительного IPv4r2 адреса производится по тем же принципам как и основного, после того, как совпал основной адрес.

Адреса в основных режимах IPv4r2 адресации имеют реальную длину 9 или 10 байт, но отображаются на 16 байтовое (128 битное) четырех- уровневое представление, а в режиме обобщенного расширения адресации, адреса непрерывно отображаются до того уровня, до которого адрес заполняется байтами адреса, дополняя старшие байты IPv4 адреса последнего уровня нулями (в примере 0.0.0.d2). Другими словами, гранулярность (минимальное приращение) длины обобщенного адреса равна четырем байтам, независимо от того сколько байт хранится в поле адреса заголовка. При упаковке обобщенного адреса в IPv4r2 заголовок, три старшие байта последнего уровня могут быть отброшены, если равны нулю.

Поэтому чтобы выразить базовые адреса с помощью обобщенной адресации, размер обобщенного адреса должен быть равен 13 байтам (102 бит), а не 9 или 10 байт. Для основной IPv4r2 адресации нет проблемы вычисления и сравнения длины адреса, это всегда адрес 16 байт (128 бит) длиной в виде 9 или 10 реально используемых байт.

Таким образом, обобщенное расширение адресации позволяет обращаться в пределах 128 бит к глобальным адресам, которые недостижимы с помощью основных режимов адресации, но практического смысла в такой адресации нет, поскольку базовая адресация итак уже очень велика.

-) Асимметричный режим IPv4r2 адресации 77 бит (основной режим IPv4r2 адресации).

Этот 77 битный адрес отображается на 128 битный адрес четырехуровневой иерархии IPv4 адресов:

a1.b1.c1.d1/a2.b2.c2.d2/a3.b3.c3.d3/a4.b4.c4.d4
как
a1.b1.c1.d1/c2.d2/b3.c3.d3/d4

Биты поля <расширения адреса>[45] от старшего бита к младшему биту располагаются по группам:
<d4>[3]<b3>[10]<c3.d3>[16]<c2.d2>[16]

Сеть d4 из 8 адресов рассматривается как

- 2 битная IPv4r2 подсеть с хостами 0-3 (обратная маска /:-2, 0,3 разрешены как адреса хоста)
- 3 битная IPv4r2 подсеть с хостами 0-7 (обратная маска /:-3, 0,7 разрешены как адреса хоста)
- 6 битная IPv4r2 подсеть с хостами 0-7 (обратная маска /:-6, 0 разрешен как адрес хоста)

а) корневая IPv4 сеть (первый уровень)
в которой IPv4 адрес a1.b1.c1.d1, 32 бита
выдаются так же, как и сейчас в интернете
эти a1.b1.c1.d1 адреса являются точками входа в деревья IPv4r2 сетей
эти деревья бывают разных типов по своей структуре, например:
a1.b1.c1.1 это региональная IPv4r2 структура (классификация по странам, подобно ru/ua доменным корням)
a1.b1.c1.2 это функциональная IPv4r2 структура (классификация по типу, подобно com/org доменным корням)
и т.д.

Наличие корневой IPv4 сети для адресации IPv4r2 позволяет обычным сетям IPv4 по крайней мере доставлять IPv4r2 пакеты в адрес IPv4 точек входа в сети IPv4r2 адресации, а шлюзы IPv4, которые маршрутизируют пакеты на пути до этих IPv4 точек входа и сами эти IPv4 точки входа уже должны быть подсоединены к сетям IPv4r2 и уметь доставлять пакеты анализируя полный IPv4r2 адрес.

Далее рассмотрим IPv4r2 сеть с региональной структурой.

б) второй уровень
в которой IPv4 адрес c2.d2, 16 бит
формат c2.d2 зависит от структуры этой IPv4r2 сети (от корневой точки входа a1.b1.c1.d1)
для IPv4r2 сети региональной структуры
c2.d2 имеют формат <региональный код>[10]<региональная сеть>[6]
так что что для каждой точки входа a1.b1.c1.d1 предоставляется
1024 региона Земли и 64 региональные сети в каждом таком регионе (64 региональных провайдера)
адреса c2.d2 выдаются владельцем корневой точки входа a1.b1.c1.d1

в) третий уровень
в которой IPv4 адрес a3.b3.c3.d3, 26 бит
в сети IPv4r2 региональной структуры
для каждого регионального провайдера c2.d2
адреса a3.b3.c3.d3 имеют формат <клиент>[26]
это 64 миллиона клиентов регионального провайдера
адреса <клиент> выделяются региональным провайдером c2.d2 своему клиенту

г) четвертый уровень
в которой IPv4 адрес d4, 3 бита
в сети IPv4r2 региональной структуры
адрес d4 имеет формат <сеть пользователя>[3]
для каждой a1.b1.c1.d1 IPv4r2 сети региональной структуры
для каждого регионального провайдера c2.d2,
для каждого клиента регионального провайдера a3.b3.c3.d3,
<сеть пользователя> это d4 сеть клиента с 8 адресами
адреса <сеть пользователя> распределяет сам пользователь на свои ресурсы

Таким образом, полный вид IPv4r2 асимметричного адреса назначения в IPv4r2 сети региональной структуры такой:
a1.b1.c1.d1[32]/c2.d2[16]/a3.b3.c3.d3[26]/d4[3] [всего 77 бит]
даже если занять только несколько адресов a1.b1.c1.d1 в корневой сети IPv4 под точки входа в такие IPv4r2 сети
это уже даст сотни миллиардов глобальных IPv4r2 адресов с асимметричной адресацией
которых хватит на всю солнечную систему даже без модификации IPv4 заголовка

-) Режим расширенной IPv4r2 адресации 80 бит.

Биты поля <расширения адреса>[48] от старшего бита к младшему биту располагаются по группам:
<d4>[6]<b3>[10]<c3.d3>[16]<c2.d2>[16]

Сеть d4 из 64 адресов рассматривается как
- 2 битная IPv4r2 подсеть с хостами 0-3 (обратная маска /:-2, 0,3 разрешены как адреса хоста)
- 3 битная IPv4r2 подсеть с хостами 0-7 (обратная маска /:-3, 0,7 разрешены как адреса хоста)
- 6 битная IPv4r2 подсеть с хостами 0-63 (обратная маска /:-6, 0,63 разрешены как адреса хоста)

Первые 8 хостов из 64 те же что в режиме 77 бит.

В сетях IPv4r2 региональной структуры этот режим отличается от предыдущего 77 битного только тем, что
на четвертом уровне 6 бит вместо 3,
т.е.:
для каждой a1.b1.c1.d1 IPv4r2 сети региональной структуры
для каждого регионального провайдера c2.d2
есть 64 миллиона a3.b3.c3.d3 клиентов
с d4 сетями по 64 адреса

Таким образом, полный вид IPv4r2 асимметричного адреса назначения в IPv4r2 сети региональной структуры такой:

a1.b1.c1.d1[32]/c2.d2[16]/a3.b3.c3.d3[26]/d4[6] [всего 80 бит]
даже если занять только несколько адресов a1.b1.c1.d1 в корневой сети IPv4 под точки входа в такие IPv4r2 сети
это уже даст сотни миллиардов глобальных IPv4r2 адресов с адресацией 80 бит
которых хватит на всю солнечную систему даже без модификации IPv4 заголовка

-) Режим короткой IPv4r2 адресации 72 бита.

Биты поля <расширения адреса>[40] от старшего бита к младшему биту располагаются по группам:
<d4>[2]<b3>[6]<c3.d3>[16]<c2.d2>[16]

Сеть d4 из 4 адресов рассматривается как

- 2 битная IPv4r2 подсеть с хостами 0-3 (обратная маска /:-2, 0,3 разрешены как адреса хоста)
- 3 битная IPv4r2 подсеть с хостами 0-3 (обратная маска /:-3, 0 разрешен как адреса хоста)
- 6 битная IPv4r2 подсеть с хостами 0-3 (обратная маска /:-6, 0 разрешен как адрес хоста)

Это первые 4 из 8 тех же хостов что и в режиме 77 бит.

В сетях IPv4r2 региональной структуры этот режим отличается от предыдущего 77 битного только тем, что
на третьем уровне 22 бита вместо 26,
а на четвертом уровне 2 бита вместо 3,
т.е.:
для каждой a1.b1.c1.d1 IPv4r2 сети региональной структуры
для каждого регионального провайдера c2.d2
есть 4 миллиона b3.c3.d3 клиентов
с d4 сетями по 4 адреса

Таким образом, полный вид IPv4r2 асимметричного адреса назначения в IPv4r2 сети региональной структуры такой:
a1.b1.c1.d1[32]/c2.d2[16]/b3.c3.d3[22]/d4[2] [всего 72 бит]
даже если занять только несколько адресов a1.b1.c1.d1 в корневой сети IPv4 под точки входа в такие IPv4r2 сети
это уже даст сотни миллиардов глобальных IPv4r2 адресов с адресацией 72 бита
которых хватит на всю солнечную систему даже без модификации IPv4 заголовка

-) Обобщенное IPv4r2 расширение адресации.

В обобщенном IPv4r2 расширении адресации, адреса внутренних уровней (это имеет значение для коротких адресов) раздает
владелец корневого адреса (a1.b1.c1.d1).

Адреса, численно совпадающие с основной IPv4r2 адресацией, хотя и достижимы в кодировании обобщенной IPv4r2 адресации,

но распределяются в рамках основной IPv4r2 адресации.

-) Дополнительная IPv4r2 адресация пользовательской сети.

Повторим, что это отдельное адресное пространство, распределение адресов в котором осуществляет сам владелец выделенного IPv4r2 адреса для своих ресурсов.

При маршрутизации пакеты в адрес дополнительной сети пользователя передаются на основной IPv4r2 адресат, для которого этот адрес дополнительной сети определен, но в отличие от индекса локального сокета шлюза, расширять можно как адрес назначения, так и оба адресата сразу.

При работе через через шлюз провайдера локальный адрес дополнительной сети пользователя может также подменяться, как и локальный IPv4/IPv4r2 адрес.

Работа IPv4r2 в полном режиме.

-) Подсчет контрольной суммы IPv4r2 заголовка.

Чтобы указать, что подсчет IPv4 контрольной суммы в этом заголовке IPv4r2 не производился, создатель IPv4r2 пакета записывает значение 0xFFFF в поле контрольной суммы этого IPv4r2 заголовка.

Причины возможности резервирования значения 0xFFFF в поле контрольной суммы IPv4r2 заголовка:

- суммирование IPv4 заголовка производится так, что в результате сумма может быть равна 0 только если все поля IPv4r2 заголовка равны 0, а поскольку как минимум поле версии заголовка содержит число 4, то сумма 0 никогда не может получиться;
    - в поле контрольной суммы IPv4r2 заголовка записывается инверсия этой суммы, так чтобы при сложении получилось значение 0xFFFF, значит раз сумма не равна 0x0000, то значение 0xFFFF никогда не может появляться в поле контрольной сумме IPv4r2 заголовка и может использоваться как флаг.

Проблемы резервирования значения 0xFFFF:

- Отказ IPv4r2 от вычисления контрольной суммы не является 100% совместимым с IPv4, который предполагает ее вычисление, если контрольная сумма рассматривается IPv4 маршрутизатором как поврежденная (на деле в ней записано резервное IPv4r2 значение 0xFFFF), то такой IPv4 хост не сможет отвечать ICMP ответом, поскольку отправитель неизвестен, либо IPv4 должен уметь отвечать ICMP ответом на стандартный широковещательный адрес ошибок (предыдущий шлюз находится в той же сети, где и IPv4 маршрутизатор), отправляя туда IPv4r2 заголовок;
- это нерационально расходует 2 байта IPv4r2 заголовка, позволяя этим только не считать контрольную сумму;

Старое оборудование, переносящее IPv4 пакеты, становится разделенным на два класса: совместимое с IPv4r2 по интерпретации поля контрольной суммы и несовместимое. Как бороться с несовместимостью?

a) Маршрутизатор IPv4r2 должен знать, поддерживает ли промежуточный IPv4 адрес, куда он отправляет пакет для IPv4r2 адресата, IPv4r2 отказ от подсчета контрольной суммы IPv4 заголовка.
b) Модификация модификации рознь. Для исправления программ, обслуживающих оборудование поддерживающее исходный IPv4, с целью внесения совместимости по IPv4r2 интерпретации поля контрольной суммы, изменения будут откровенно косметическими (по сравнению с ситуацией внедрения IPv6), даже "решение в лоб" путем:
   - фильтрации всех входящих IPv4 пакетов с полем контрольной суммы 0xFFFF на предмет подсчета контрольной суммы;
   - фильтрации всех исходящих IPv4 пакетов установкой поля контрольной суммы в значение 0xFFFF;

   без учета эффективности, равной занесению правильных значений в момент создания IPv4r2 заголовка.

Так вот, изменения для программ с открытым исходным кодом, связанные с использованием резервного значения 0xFFFF в поле контрольной суммы, будут очень простыми, для модификации не потребуется смена ядра, переделка всего стека протоколов и не нужны т.п. глобальные обновления.

-) Формат IPv4r2 заголовка в основном режиме.

В полном режиме IPv4r2 заголовок может задействовать резервный бит флагов IPv4 заголовка, устанавливая этот бит в 1.

Установка в 1 резервного бита флагов IPv4 заголовка указывает на формат IPv4r2 заголовка. Это приводит к следующему:

- в IPv4 поле протокол резервируется специальное значение номера протокола=0 (число 0)
  - если номер протокола не равен 0, то
    - предполагается работа IPv4r2 протокола в режиме межсетевого взаимодействия с MTU 1152 (работа в интернет);
    - поле протокола имеет такое же значение, как и для IPv4 заголовка
  - если номер протокола равен 0, то
    - предполагается работа IPv4r2 в режиме локальных и специальных сетей с MTU больше чем 1152
    - используется формат расширения статических полей IPv4r2 заголовка (см далее обобщенное расширение IPv4r2 заголовка);

Установка в 1 резервного бита флагов и поле протокол не равно 0 (MTU 1152):

- запрещает IPv4 фрагментацию;
- запрещает подсчет IPv4 контрольной суммы;
- включает асимметричный 45 бит режим базовой IPv4r2 адресации (основной IPv4r2 режим межсетевого взаимодействия 77 бит), при этом освободившиеся поля IPv4 заголовка передаются под IPv4r2 поле расширение адреса (дополнительно 45 бит), состоящее из:

    - IPv4 поле контрольной суммы
      хранит <c2.d2>[16]
    - IPv4 поле идентификатора
      хранит <c3.d3>[16]
    - IPv4 поле смещения сегмента
      хранит <d4>[3]<a3.b3>[10]
- IPv4 поле флаги имеет новое значение:
    - 100b (IPv4 резервный флаг)
      признак IPv4r2 заголовка
      поле расширение адреса 45 бит IPv4r2 заголовка принадлежит адресу назначения
    - 010b (IPv4 D флаг)
      поле расширение адреса 45 бит IPv4r2 заголовка принадлежит адресу источника вместо адреса назначения
    - 001b (IPv4 M флаг),
      расширение максимального размера IPv4r2 заголовка до 80 байт

Асимметричный 45 бит режим базовой IPv4r2 адресации предназначен для работы с IPv4r2 серверами пользователя в сети интернет через шлюз провайдера, поэтому по умолчанию расширяется поле адреса назначения. При отправке обратных пакетов включается флаг 010b, который указывает что расширяется поле источника.

Фактически такой формат IPv4r2 заголовка позволяет использовать IPv4r2 глобальную адресацию не используя опции IPv4 заголовка, т.е. при использовании 77 битной IPv4r2 адресации IPv4r2 заголовки имеют размер 20 байт, как и для 32 битных IPv4 адресов. Если сеть провайдера поддерживает IPv4r2, то этот формат IPv4r2 заголовка является наиболее часто используемым.

Расширение размера IPv4r2 заголовка до 80 байт прибавляет число 20 к размеру заголовка хранящемуся в IPv4r2 поле размер заголовка, т.е.:

- IPv4r2 поле размер заголовка=5..F означает IPv4r2 заголовок=5*4..15*4=20..60 байт;
- IPv4r2 поле размер заголовка=0..F в режиме расширения до 80 байт означает IPv4r2 заголовок=20+(0*4..15*4)=20..80 байт.

Проблемы в использовании резервного бита флагов:

- отметим, что существует rfc3095, предлагающее использовать этот резервный бит флагов для "передачи IPv4 пакетов в сетях сотовой связи", но доставка пакетов в сетях "точка-точка" позволяет обертывать IP пакеты в дополнительную транспортную оболочку, которая никак не связана с IP заголовком и может позволять как сжимать IP пакеты в целом, так и добавлять средства контроля целостности данных;

- также отметим, что сотовые компании не очень заинтересованы в том, чтобы трафик с абонента был бы минимальным, что для "общества свободного рынка не уравновешенного системой ценностей" есть ситуация обычная, если один абонент займет весь канал, владелец сети получит столько же, как если бы канал был занят тысячей абонентов - для узких каналов продается их пропускная способность, а не безлимитный доступ (количество абонентов);
- нам выгодно, включив резервный флаг IPv4, добиться изменения формата IPv4 заголовка для нужд IPv4r2 и использовать освободившиеся IPv4 поля для расширения IPv4r2 адресации.

Старое оборудование, переносящее IPv4 пакеты, становится разделенным на два класса: совместимое с IPv4r2 по использованию IPv4 резервного флага и протоколом не равным 0 и несовместимое. Как бороться с несовместимостью?

a) Маршрутизатор IPv4r2 должен знать поддерживает ли промежуточный IPv4 адрес, куда он отправляет пакет для IPv4r2 адресата, использование IPv4 резервного флага и протокола не равного 0 или нет.
b) Правильное поведение шлюза IPv4, когда он встречает IPv4r2 пакет с резервным битом в неправильном положении, это отбросить такой IPv4 пакет, поскольку формат IPv4 заголовка для такого взведенного бита, кроме поля версия и резервного бита поля опции, может быть произвольным (в описании IPv4 формат заголовка с таким битом не определен). Но существуют IPv4 реализации (или опции настройки такого поведения), которые полагают, что резервные биты можно безопасно игнорировать, а не отбрасывать пакеты с такими битами в неправильном состоянии.
c) Модификация модификации рознь. Для исправления программ, обслуживающих оборудование поддерживающее исходный IPv4, с целью внесения совместимости по IPv4r2 интерпретации IPv4 резервного флага поля флагов и протокола не равного 0, изменения будут откровенно косметическими (по сравнению с ситуацией внедрения IPv6), даже "решение в лоб" путем:
    - фильтрации всех входящих IPv4 пакетов на предмет проверки IPv4 установленности резервного флага и протокола на неравенство 0 и переноса адресных полей из IPv4 заголовка в IPv4r2 опции;
    - фильтрации всех исходящих IPv4 пакетов на предмет установки IPv4 резервного флага и переноса адресных полей из IPv4r2 опций в IPv4 заголовок;
    - расширение размера IPv4r2 заголовка до 80 байт во многих случаях не потребует никакой существенной переделки, кроме добавления исправленной копии оригинальной IPv4 функции, обрабатывающей максимальный размер IPv4 заголовка и механизма выбора функции в зависимости от флага заголовка.

-) Обобщенное расширение IPv4r2 заголовка.

Обобщенное расширение размера IPv4r2 заголовка предназначено:

- для работы IPv4r2 в локальных и специальных сетях, когда MTU больше чем 1152.

Как и в случае расширение максимального размера IPv4r2 заголовка до 80 байт, это обобщенное расширение не предназначено для замены специальных протоколов опциями IPv4.

Для обобщенного расширения максимального размера IPv4r2 заголовка:

- должен быть включен резервный бит IPv4 поля флагов;
- в IPv4 поле протокол записывается номер специального протокола=0;

важно отметить, что при этих условиях заголовок IPv4r2 не состоит из двух раздельных заголовков двух разных протоколов, а состоит только из одного заголовка одного протокола IPv4r2, значение 0 в IPv4 поле протокола указывает что для IPv4r2 это вовсе не поле протокола.

Установка в 1 резервного бита флагов и поле протокол не 0 (MTU более 1152):

- запрещает IPv4 фрагментацию;
- запрещает подсчет IPv4 контрольной суммы;
- освободившиеся поля IPv4 заголовка зарезервированы:
  - IPv4 поле контрольной суммы
    зарезервировано <0>[16]
  - IPv4 поле идентификатора
    зарезервировано <0>[16]
  - IPv4 поле смещения сегмента
    зарезервировано <0>[13]
- IPv4 поле флаги имеет новое значение:
  - 100b (IPv4 резервный флаг)
    признак IPv4r2 заголовка
  - 010b (IPv4 D флаг)
    резерв 0
  - 001b (IPv4 M флаг),
    резерв 0
- к фиксированной части IPv4r2 заголовка прибавляются новые поля, идущие заместо IPv4 списка опций, начиная со по смещения 20:

```
0               1                 2                 3
 0 1 2 3 4 5 6 7  8 9 0 1 2 3 4 5   6 7 8 9 0 1 2 3   4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| EIHL          | EProtocol       | ELength                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

  - поле EIHL
    содержит значение размера дополнительного заголовка в 4 байтовых словах
    размер EIHL включает в себя и сами новые поля (минимальное значение EIHL = 1)
    значение IHL поля размера IPv4 заголовка при этом расширении всегда = 5
    максимальный размер IPv4r2 заголовка равен 1020(EIHL)+20(IHL)=1040 байт

- поле EProtocol
  содержит протокол который должен быть в поле IPv4 заголовка Protocol, если не применяется обобщенное расширение IPv4r2 заголовка
  значение EProtocol=0 также зарезервировано
  - если EProtocol не равен 0, то после новых статических полей IPv4r2 заголовка идет поле списка опций, аналогичное IPv4 заголовку;
  - если EProtocol равен 0, то формат IPv4r2 пакета не определен;
- поле ELength
  старшие два байта размера IPv4r2 пакета (младшие два байта в поле размер пакета IPv4 заголовка)
  максимальный размер IPv4r2 пакета при этом 4Гбайта

В режиме локальной и специальной сети асимметричная IPv4r2 адресация по умолчанию не задействована, IPv4r2 адресация достигается с помощью IPv4r2 адресных опций записанных в IPv4r2 поле опций (после новых статических полей IPv4r2 заголовка). Для локальной IPv4 сети будет достаточно IPv4 адресов IPv4 локальной сети (являющихся адресами корневой IPv4r2 сети).

Старое оборудование, переносящее IPv4 пакеты, становится разделенным на два класса: совместимое с IPv4r2 по использованию IPv4 резервного флага и протокола равного 0 и несовместимое. Как бороться с несовместимостью?

a) Маршрутизатор IPv4r2 должен знать поддерживает ли промежуточный IPv4 адрес, куда он отправляет пакет для IPv4r2 адресата, использование IPv4 резервного флага или нет.
b) Модификация модификации рознь. Для исправления программ, обслуживающих оборудование поддерживающее исходный IPv4, с целью внесения совместимости по IPv4r2 интерпретации IPv4 резервного флага поля флагов и протокола равного 0, изменения будут откровенно косметическими (по сравнению с ситуацией внедрения IPv6):
   - поддержка IPv4r2 пакетов 4Гбайта во многих случаях не потребует никакой существенной переделки, кроме добавления исправленной копии оригинальной IPv4 функции, обрабатывающей максимальный размер IPv4 пакета и механизма выбора функции в зависимости от флагов заголовка.

Версии UDPr2/TCPr2 в стеке протоколов IPv4r2.

В стеке протоколов IPv4r2, протоколы UDP/TCP имеют новые версии

- TCPr2 предположительно номер 0xA6
- UDPr2 предположительно номер 0xB1

-) Адресация портов UDPr2/TCPr2 увеличена с 16 до 32 бит.

Это нужно для поддержки шлюзов и защитных сетевых фильтров, делаем это потому, что все равно надо модифицировать заголовок, чтобы поддержать UDP/TCP размер пакета для локальных сетей до 4 Гбайт.

-) Протокол TCPr2 теперь имеет поле размер пакета.

Поле размер пакета определяет размер пакета адресуемый TCPr2 полем порядковый номер, это потому что UDPr2/TCPr2 не требуют для своей работы какого-либо сетевого протокола (например, IPv4r2).

Обращаясь к локальному серверному порту, UDPr2/TCPr2 могут обращаться к серверу пользователя, который установил канал связи с машиной в локальной сети, т.е. может существовать иной, кроме IPv4r2, программный механизм отображения сетевых компьютеров на локальные порты, так что протоколам UDPr2/TCPr2 нет нужды знать о сетевых адресах и конфигурации сети, если пользователь нужные локальные сетевые ресурсы к локальным портам UDPr2/TCPr2 подключает сам.

Также сетевой протокол IPv4r2 может содержать служебные данные в IPv4r2 пакете, часть из которых не относится к данным UDPr2/TCPr2 или заголовку IPv4r2.

Т.е. для UDPr2/TCPr2 существует единственный уникальный сетевой адрес "локальная машина", которому не нужно присваивать фиктивный сетевой IPv4r2 адрес, это случается, когда сетевой адрес просто не указан.

-) TCPr2 поле размер пакета имеет размер 32 бит.

Это нужно для поддержки локальных сетей. TCPr2 поле размер пакета идет после статических полей заголовка TCP (смещение 24) и записывается четырьмя байтами, размер пакета в байтах, размер TCPr2 пакета вырос до 4 Гбайт.

TCP опция Window scaling запрещена в TCPr2.

-) Протокол TCPr2 теперь имеет поле отметка времени 32 бит.

Это поле записывается четырьмя байтами после поля размер пакета (смещение 28).

Отметку времени проставляет источник TCPr2 сегмента, во время осуществления связи приемник может синхронизировать время источника и оценивать время отправления каждого пришедшего с момента установления связи сегмента.

-) Размер TCPr2 заголовка.

- минимальный размер заголовка в TCPr2 поле смещение данных был 5 стал 8 (32 байта)
- максимальный размер заголовка в TCPr2 поле смещение данных 15 (60 байт)

Для поддержки IPv4r2 заголовка в режиме межсетевого взаимодействия размером 80 байт, реализация TCPr2 работающая через IPv4r2 должна уметь ограничивать максимальный размер TCPr2 заголовка 48 байтами (значение 12).

-) UDPr2 поле размер пакета увеличено с 16 до 32 бит.

Это нужно для поддержки локальных сетей, размер дейтаграммы увеличен до 4 Гбайт. Между UDPr2 полями размер пакета и контрольная сумма идут два резервных байта 0.

-) Протокол UDPr2 теперь имеет поле отметка времени 32 бит.

Это поле идет в UDPr2 заголовке последним (смещение 16).

Отметку времени проставляет источник UDPr2 дейтаграммы, приложение принимающее дейтаграммы может синхронизировать время источника и оценивать время отправления каждой прибывшей дейтаграммы.

-) Размер UDPr2 заголовка.

UDPr2 имеет заголовок постоянного размера, этот размер вырос на 12 байт и заголовок UDPr2 равен 20 байтам (минимальный размер UDPr2 дейтаграммы).

-)Контрольная сумма псевдо- заголовка.

Протоколы UDPr2/TCPr2 более не подсчитывают контрольную сумму для псевдо- заголовка, а только для заголовка и блока данных самого пакета UDPr2/TCPr2, попытка UDP/TCP сетей IPv4 включать сетевой адрес в псевдо- заголовок нарушает инкапсуляцию данных по уровням, UDPr2/TCPr2 полностью доверяет сетевую адресацию сетевому уровню, например IPv4r2.

Расширения UDP/TCP в стеке протоколов IPv4r2.

Далее, кроме отличий в заголовках, под "UDP/TCP" подразумеваются оба варианта протоколов UDP/TCP в сетях IPv4r2:

- старый UDP/TCP (0x11/0x06)
- новый UDPr2/TCPr2 (0xA6/0xB1)

-) Контрольная сумма псевдо- заголовка.

Протоколы UDP/TCP могут не подсчитывать контрольную сумму, помещая в поле контрольной суммы 0xFFFF, это возможно для UDP/TCP, поскольку поле длина пакета или смещение данных как минимум включает ненулевой размер заголовка UDP/TCP.
-) TCP поле Window Size.
Протокол TCP имеет в поле Window Size значение

- 512 для IPv4
- 1024 для IPv4r2
- любое иное для локальных сетей

- 0 при перегрузке

По истечении таймера перегрузки, если подтверждения не было получено отправляется пробный пакет с размером:
- 512 для IPv4
- 1024 для IPv4r2
- любое иное для локальных сетей

TCP опция Maximum segment size имеет такие же правила 512/1024 для IPv4/IPv4r2 и поддерживает второй, 32 битный формат для локальных сетей:
- 2,6,SSSS

-) Максимальный размер дейтаграммы UDP для IPv4r2.

UDP работающий локально в стеке IPv4r2 гарантирует, что как минимум 1024 байта самих данных в дейтаграмме может быть всегда принято самим протоколом UDP.

Для доставки UDP дейтаграмм через сеть IPv4r2 в межсетевом режиме IPv4r2, максимум 1024 байта данных в дейтаграмме может быть доставлено, если используются стандартные IPv4r2 (без расширения IPv4r2 заголовка или с расширением IPv4r2 заголовка до 80 байт) и UDP заголовки, иначе пакет не пройдет через IPv4r2 без фрагментации и может быть не доставлен до UDP назначения по причине IPv4r2.

-) Арифметика TCP полей порядковый номер и номер подтверждения.

При переполнении эти поля подчиняются арифметике по модулю 2, а не той арифметике, что для контрольной суммы TCP.

===
Конец текста