**Упрощенный алгоритм удаленной маршрутизации пакетов с использованием таблиц**

Маршрутизацию можно условно разделить на прямую и косвенную. Прямая маршрутизация выполняется в том случае, если источник сообщения находятся в той же сети, что и приёмник. То есть прямая маршрутизация не требует использования маршрутизаторов, а предполагает широковещательную рассылку пакета по своей сети. Косвенная маршрутизация осуществляется в том случае, если отправитель и получатель находятся в разных IP-сетях.

В этом случае процесс передачи пакета по сети начинается с его направления на маршрутизатор, который настроен как принятый по умолчанию шлюз для отправителя данных. То есть отправка (первоначальная) пакета на шлюз (маршрутизатор) и будет прямой маршрутизацией.

Получив пакет для удаленной пересылки, маршрутизатор извлекает из него целевой IP-адрес, по сетевому протоколу которого осуществляется определение пути следования пакета (следующего маршрутизатора на пути следования пакета).

Использование для маршрутизации сетевых префиксов, а не IP-адресов отдельных компьютеров позволяет значительно сократить объем сетевых таблиц (таблиц маршрутизации). Таким образом, принятие решения о передаче пакета осуществляется на маршрутизаторе на основании таблицы маршрутизации.

**Два подхода к выбору маршрута:**

1. Одношаговый;
2. Маршрутизация от источника.

При одношаговом методе каждый маршрутизатор принимает участие в выборе только одного шага передачи пакета. Одношаговая маршрутизация (метод одношаговый маршрутизации) основывается на виде таблицы маршрутизации, в которой для отдельной сети указывается не весь маршрут передачи пакета, a IP-адрес следующего маршрутизатора, находящегося на пути следования пакета. Алгоритмы одношаговой маршрутизации являются наиболее часто используемыми в сети Интернет.

При маршрутизации от источника выбор всего маршрута определяется первым маршрутизатором на пути следования дейтаграммы (то есть таблица должна содержать последовательность IP-адресов маршрутизаторов, через которые должна пройти дейтаграмма). Остальные маршрутизаторы только отрабатывают выбранный маршрут (метод применяется в IP-сетях только при их отладке).

**Классы алгоритмов одношаговой маршрутизации:**

1. Алгоритмы фиксированной маршрутизации (статическая маршрутизация, выполняемые с использованием таблиц, созданных “вручную” администратором).
2. Алгоритмы простой маршрутизации:
3. случайная маршрутизация (дейтаграмма передаётся в любом направлении, кроме исходного - не гарантирует доставку;
4. лавинная маршрутизация (пакет передается во всех направлениях, кроме исходного) - чрезмерное количество информации, рассылаемой по сети.
5. Алгоритмы адаптивной маршрутизации (маршрутизаторы периодически обмениваются между собой информацией с топологией сети) - динамическая маршрутизация - динамическое (автоматическое) перестроение таблиц в соответствии с изменениями в топологии сети.

Упрощенный формат таблиц для статической и динамической маршрутизации имеет одинаковый вид.

**Формат таблицы маршрутизации.**

**Сетевой префикс. Следующий маршрутизатор в сети. Метрика**

Поле сетевого префикса содержит сетевой адрес целевой сети, в которую осуществляется маршрутизация пакета, поле “Следующий маршрутизатор в пути” содержит IP-адрес маршрутизатора, куда передается пакет при одношаговой маршрутизации. Поле “Метрика” содержит числовое значение, позволяющее сравнивать маршруты между собой.

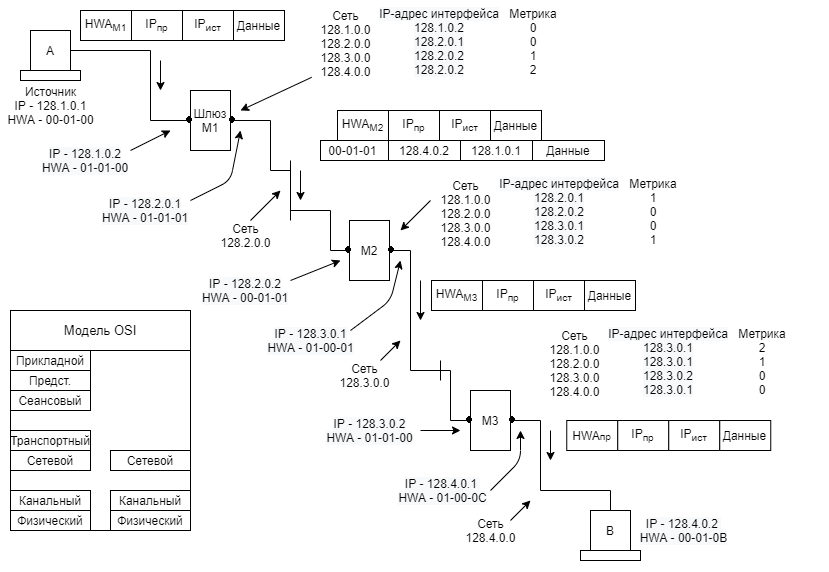
Так как для большинства протоколов в таблице содержит только один маршрут до целевой сети, поле “Метрика” используется в том случае, когда произошёл обмен с другими маршрутизаторами сети и необходимо идентифицировать, какой из маршрутов лучше – “старый”, находившийся в таблице или переданный в результате обмена таблицами.

В тоже время существуют некоторые протоколы (например, OSPF EIGRP), которые поддерживают несколько маршрутов целевой сети.

**Пример таблицы маршрутизации с несколькими маршрутами:**

|  |  |  |
| --- | --- | --- |
| Целевая сеть | След. маршрутизатор в пути | Метрика |
| 128.3.0.0 | 128.5.3.2 | 3 |
| 128.3.0.0 | 128.5.4.7 | 5 |

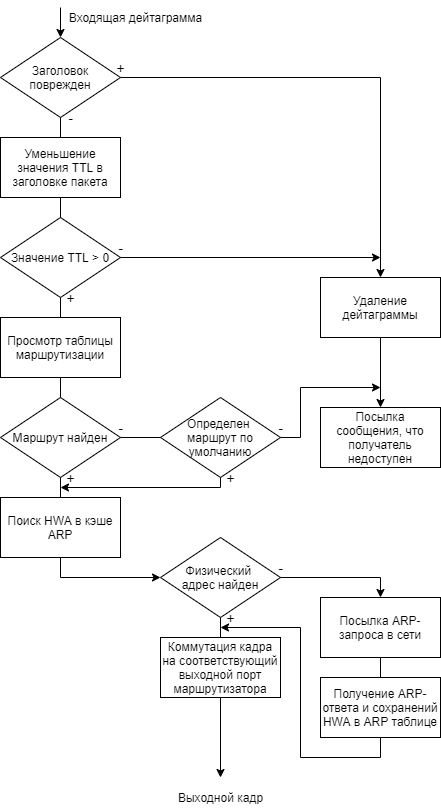
**Процесс пересылки пакетов при удалённой маршрутизации:**



1. Процесс маршрутизации пакета начинается с просмотра соответствия источником сетевой части целевого IP-адреса и своего сетевого префикса. В случае, если “свой” и “целевой” сетевые префиксы различаются, необходимо переслать пакет на шлюз для маршрутизации.
2. Адрес (IP) принятого по умолчанию шлюза настроен в ПО системы А. По этому адресу осуществляется определение HWA и на маршрутизатор пересылается пакет.
3. Шлюз (маршрутизатор 1), приняв на свой HWA кадр, извлекает из кадра пакет, выделяет на сетевом уровне из пакета целевой IP-адрес (его сетевой префикс) и по таблице определяет IP интерфейса следующего маршрутизатора на пути пакета (то есть определил куда далее должен быть доставлен пакет).
4. На основе выделенного из таблицы IP-адреса шлюз (М1) выполняет широковещательную рассылку протокола ARP, в которой указывается IP следующего маршрутизатора на пути пакета (IP = 128.2.0.2). Получив HWA соответствующего интерфейса маршрутизатора 2, шлюз формирует кадр, оставляя без изменения передаваемый пакет. Пакет передаётся на М2 по соответствующему HWA.
5. Процесс передачи пакетов между маршрутизаторами повторяется до тех пор, пока пакет не достигнет маршрутизатора 4.
6. М4 определит, что целевой узел находится в сети, подключенной непосредственно к нему и косвенную маршрутизацию выполнять не надо. То есть пакет должен быть выставлен в сеть через интерфейс 128.4.0.1. М4 генерирует запрос на аппаратный адрес целевого компьютера, формируют кадр и пересылает его на компьютер B.

Таким образом, сетевой уровень не только выполняет маршрутизацию пакета, но и управляет коммутацией пакета при передаче его по сети (управляет протоколом ARP и формированием кадра).

Алгоритм маршрутизации пакетов должен быть представлен в следующем виде на примере протокола RIP, в котором для исключения петель маршрутизации введён параметр TTL дейтаграммы (максимальное число транзитных участков, которое может пройти пакет до его удаления).



**Классификация протоколов динамической маршрутизации**

**По способу построения таблиц:**

Таблицы маршрутизации, формируемые и поддерживаемые на узлах, должны отслеживать возможные изменения топологии сети. За изменением топологии сети следят алгоритмы динамической маршрутизации. В зависимости от способов построения и поддержания в рабочем состоянии таблиц маршрутизации различают два типа протоколов: *протоколы вектора расстояния* и *протоколы состояния каналов связи*.

**Протоколы вектора расстояния**

Алгоритм построения таблицы маршрутизации:

1. Рассылка запроса на соседние маршрутизаторы об информации в них;
2. Анализ поступивших маршрутов соседних узлов с точки зрения их наименьшей стоимости. Занесение оптимальных маршрутов себе в таблицу.

Алгоритм поддержания таблицы (на примере протокола RIP):

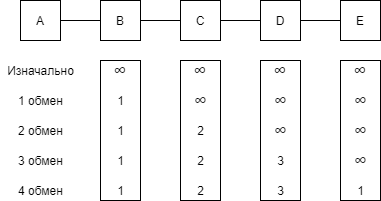
1. Маршрутизатор с определенной периодичностью извлекает адреса получателей информации и метрику из своей таблицы и помещает эти данные в рассылаемые соседям сообщения;
2. Получив сообщение, маршрутизатор изменяет свои таблицы если:
3. маршрутизатору, от которого пришло сообщение известен более короткий путь к получателю или в списке рассылки есть получатель, сведения о котором отсутствуют в таблице.
4. маршрутизатор выполняет маршрутизацию до конечной сети через маршрутизатор, от которого пришло сообщение, но расстояние от этого маршрутизатора до конечной сети изменилось.

Недостатки протоколов векторной маршрутизации:

1. Избыточность передаваемой по сети служебной информации;
2. Замедленная реакция протоколов на изменение топологии сети (обновление таблиц при изменении структуры сети происходит с большой задержкой).

**Пример, поясняющий обновление таблиц при включении маршрутизатора сеть.**

Происходит включение маршрутизатора А в сеть (RIP):



Таблицы на других маршрутизаторах отображают пути до А.

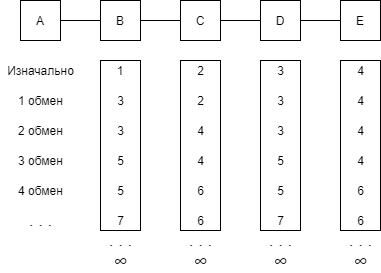
Первоначально, так как маршрутизатор А недоступен, расстояние до него во всех таблицах - ∞ (маршрут отсутствует).

При включении А, он посылает сообщение своему соседу. Все маршрутизаторы обмениваются таблицами синхронно, через одинаковые промежутки времени.

Для сети с максимальной длиной маршрута (N маршрутизаторов) сообщение дойдет до последнего (удаленного) узла в сети через N -1 цикл обмена таблицами.

Пример выключения маршрутизатора из сети (узел А перестает работать вследствие сбоя).

Узел B не получает никакой информации от А. С сообщает, что он знает путь от А и метрика пути 2. В результате B заносит в таблицу путь до А (неработающего) стоимостью 3.



При следующем обмене на С приходит сообщение от B и D о том, что у них есть путь до A, равный 3. C корректирует свою таблицу (метрика 4). Данный обмен может осуществляться до бесконечности.

Данная проблема алгоритма вектора расстояния называется проблемой возрастания до бесконечности.

**Дополнительный недостаток протокола вектора расстояния.**

Если алгоритм маршрутизации анализирует только число транзитных участков (RIP), то выбранные маршруты могут быть не оптимальными. Протокол может выбрать более короткий маршрут, но с более медленными каналами связи. Протокол IGRP (фирма Cisco) является комбинированным протоколом (гибридным протоколом), анализирующем наряду с числом транзитных участков и скорость передачи данных по каналам связи.

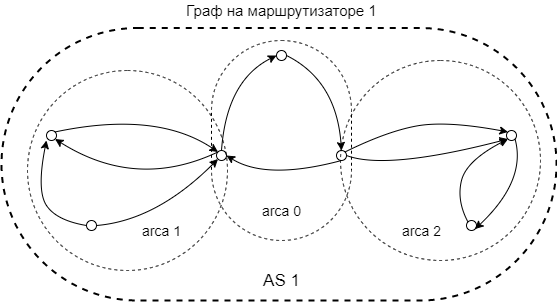
**Протоколы маршрутизации по состоянию канала связи**

Протоколы по состоянию канала связи анализируют маршрут на основе таких критериев как (сложная метрика): скорость передачи, задержка линии, надёжность передачи пакетов. Алгоритм функционирования протокола предполагает, что каждый маршрутизатор внутри AS должен иметь полную информацию о топологии (топологической схеме) сети в зоне. У каждого маршрутизатора есть карта, на которой изображены все маршрутизаторы, а также сети, к которым они подключены.

**Алгоритм построения таблицы маршрутизации протоколом по состоянию канала связи (после включения маршрутизатора в сеть)**

1. Первоначальная рассылка через физические интерфейсы подключаемого маршрутизатора пакетов - приветствий на соседние узлы. Соседний узел сообщает данные о себе - состояние своих КС (параметры задержки и пропускной способности, IP-интерфейсы).
2. Тестирование своих каналов связи с целью выяснения метрики каждого канала (посылка эхо-запросов с целью выяснения задержки, пропускной способности).
3. Рассылка полученных параметров своих соединений соседним (в зоне) маршрутизаторам сети. Получив пакет, маршрутизатор сохраняет копию в своей БД и посылает пакет с объявлением всем остальным своим соседям. Пакет снабжается адресом отправителя, поэтому каждый маршрутизатор знает место его расположения.
4. После получения информации со всех узлов внутри зоны маршрутизатор формирует топологию сети.
5. Построение ориентированного графа, отображающего топологию. Соединения “точка-точка” в графе (между узлами) представляются парой дуг (по одной в каждом направлении). Стоимости дуг отличаются друг от друга. Стоимости дуг отображают метрику.

**Пример ориентированного графа топологии сети:**



В вершине графа находится маршрутизатор, на котором осуществляется построение таблиц (источник рассылки пакетов).

1. На основании графа ищутся пути до всех сетей от исходной вершины (то есть до целевой вершины).

Оптимальные пути до каждой из сетей (с максимальной стоимостью) заносятся в таблицу маршрутизации.

Стандартный OSPF поддерживает два пути - с максимальной стоимостью и следующий за ним. Вообще могут поддерживаться до 6 маршрутов с одинаковой метрикой.

**Последовательность сигналов при построении таблицы:**

1. Рассылка приветствия;
2. Тестирование каналов связи;
3. Рассылка сообщений по сети с характеристиками КС;
4. Построение БД топологии сети;
5. Построение таблицы маршрутизации.

В процессе построения используется две таблицы: БД топологии сети, БД маршрутов.

**Протоколы внутришлюзовой маршрутизации. Протокол RIP. Протокол вектора расстояния**

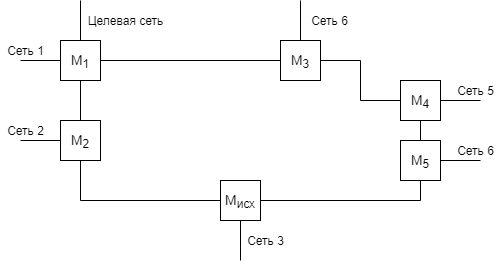
**Преимущество RIP-протокола:**

**Недостатки RIP:**

1. Увеличение трафика за счёт рассылки широковещательных сообщений;
2. Ограничивает размер автономной системы, так как он поддерживает длины путей, которые содержат не более 15 переходов.

Протокол реализует алгоритм вектора расстояния (длины вектора). Каждое сообщение об обновлении состоит из набора парных значений. Пара представляет собой IP целевой сети и расстояние до этой сети, выражающее количество транзитных участков . Так как данная пара чисел поступает от соседних маршрутизаторов, (IP которых известны) на исходном маршрутизаторе вычисляется новая метрика маршрутов через каждый из соседних маршрутизаторов и в таблицу заносится наикратчайший путь.

**Пример пересылки информационных сообщений о структуре сети:**



Осуществляется построение таблицы на маршрутизаторе Mисх (исходный маршрутизатор).

Маршрутизаторы М2 и M3 получают по сети пару чисел (1, 1) (1-я сеть за один проход). Маршрутизатор М4 – (1, 2). Маршрутизатор М5 – (1, 3).

Таким образом, маршрутизатор Mисх принимает пары (1, 2) и (1, 3) через свои интерфейсы и сохраняют маршрут через М2, так как он кратчайший (IP М2 известен на Mисх). В результате на Mисх будет присутствовать маршрут (1, 2).

Таким образом, алгоритм маршрутизации основывается на том, что каждый маршрутизатор может вычислить наикратчайший путь на основе полученной пары (сеть, метрика). Маршрутизатор из двух (и более – сколько портов) выбирает тот соседний маршрутизатор, который находится на коротком маршруте до получателя. Метрика вычисляется на основе транзитной информации с других маршрутизаторов.

**Формат таблицы маршрутизации, поддерживаемой протоколом RIP:**

|  |  |  |  |
| --- | --- | --- | --- |
| IP сети | IP маршрутизатора на пути | Метрика | Таймер |

Здесь “IP маршрутизатора на пути” не только определяет маршрут, но и позволяет контролировать работоспособность маршрутизатора, отвечающего за этот маршрут (если с указанного IP через интервал времени не пришли обновленные маршруты, маршрутизатор с этим IP считается вышедшим из строя). Поле “таймер” определяет время с момента последнего обновления записи.

**Процедура обновления таблицы и исключения маршрута из неё:**

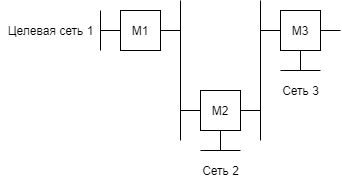
1. Рассылка пар (сеть, метрика) через каждые 30 секунд;
2. Если маршрутизатор не получает сообщения от маршрутизатора, который отвечает за определенную запись в таблице, через 180 секунд он предполагает, что соседний маршрутизатор недоступен;
3. Маршруты через этот маршрутизатор удаляются. Все маршруты (до тех же сетей) строятся теперь через другого соседа.

*Примечание:*

Интервал в 180 секунд (6 раз по 30 секунд) необходим для того, чтобы избежать исключения маршрута при случайной потере сообщения об обновлении.

**Образование петель маршрутизации протокола RIP**

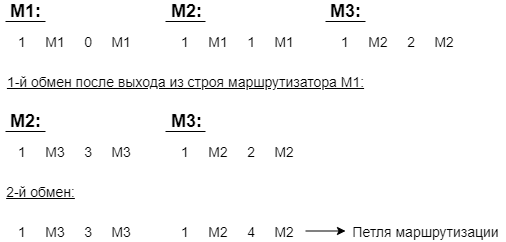
Причиной образования петель маршрутизации при использовании протокола RIP является проблема возрастания до бесконечности при выходе из строя одной из сетей (медленное удаление информации из таблиц).



**Расширенный формат таблицы:**

|  |  |  |  |
| --- | --- | --- | --- |
| Сеть | Маршрут | Метрика | Маршрут, отвечающий за маршрутизацию |

**Изначальный вид таблиц на маршрутизаторах**



Таким образом, пакеты с М2 пересылаются на М3, a с М3 – на M2, то есть образовалась петля маршрутизации.

Обмен пакетами между маршрутизаторами продолжается до тех пор, пока поле TTL не станет равным нулю. При образовании петель затрудняется обмен маршрутной информацией между маршрутизаторами, необходимой для разрыва образовавшихся петель.

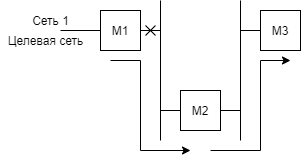
**Технологии ускорения сходимости протокола**

Сходимость - это приведение таблиц к виду, отражающему действительную топологию сети.

1. Расщепление горизонта;
2. Обратный реверс.

**1. Расщепление горизонта**

Согласно технологии “расщепление горизонта” маршрутизатор не будет распространять информацию об определенном маршруте через порт, который является источником данной информации (то есть маршрутизатор не будет информировать о достижимости получателя своего соседа, от которого получена информация о маршруте к получателю).



**Исходный вид таблиц:**



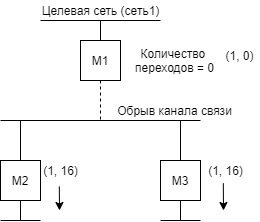
Через 180 секунд маршрутизатор не получит в сообщении об обновлении таблиц маршрут до сети 1 (а от М3 **(???)**, поэтому маршрут будет удален). Пересылаемые с М3 пакеты на М2 будут отбрасываться.

Через 180 секунд, не получив в сообщении обновления таблиц от М2 маршрут до М1, маршрутизатор М3 удаляет маршрут до сети 1 (маршрут становится недействительным, если через 180 секунд не поступило сообщение, подтверждающее его).

**2. Обратный реверс**

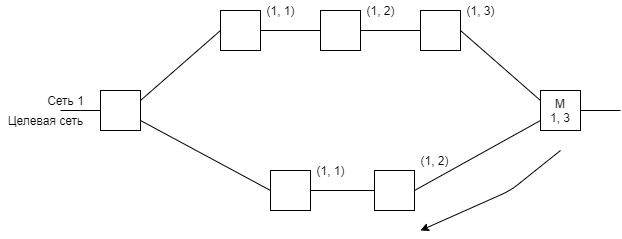
Технология “обратный реверс” предполагает оповещение соседних маршрутизаторов об отсутствии маршрутов к целевой сети. При этом в передаваемых сообщениях указывается метрика, равная 16, что определяет недостижимость целевой сети. Таким образом, данная метрика будет передана всем маршрутизаторам в сети и данная сеть в таблицах станет недостижимой.

**Пример функционирования технологии “обратный реверс”:**

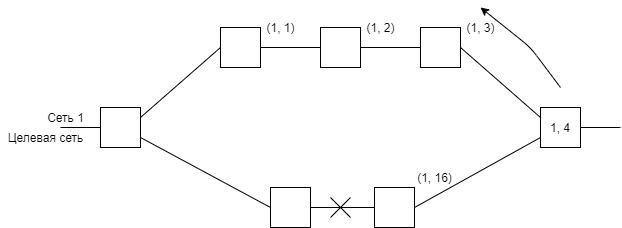


**Процесс построения таблиц при обрыве КС:**

Механизм, поясняющий первоначальное построение таблицы на маршрутизаторе М – доступность целевой сети – сети 1.



Перестроение таблицы на маршрутизаторе М при разрыве канала связи:



**Метод наибольшего совпадения при маршрутизации в сетях VLSM (с масками разной длины)**

Если получателю соответствует более чем одна запись в таблице маршрутизации, используется маршрут, имеющий наиболее совпадающий префикс.

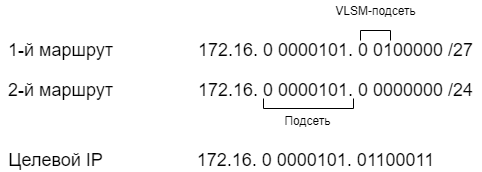
**Например, в таблице маршрутизации присутствуют два маршрута:**

*127.16.5.32/27 – маршрут к VLSM-сети.*

*127.16.5.0/24 – маршрут к подсети.*

Передаче подлежит пакет с IP-адресом 172.16.5.99.

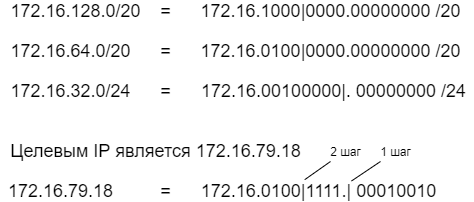
Осуществляется наложение масок на первый и второй маршруты и сравнение полученных сетевых префиксов с IP-адресом пункта назначения.



Наложение маски VLSM*-*подсети на целевой IP показывает несоответствие VLSM*-*префикса подсети в таблице и соответствующих бит в целевом IP-адресе, поэтому выбирается маршрут №2, по которому пересылается пакет.

**Пример для рисунка (\*):**

Маршрутизации выполняется на маршрутизаторе А.



Накладывается маска с большим префиксом. Определяется, что сетевой префикс целевого адреса (маска 24) не соответствует маршруту в сеть с префиксом /24. Накладывается маска с меньшим сетевым префиксом. Определяется принадлежность сетевого префикса подсети 170.16.64.0, куда и направляется пакет.

Пересылка суммарных маршрутов не вызывает запуска алгоритма построения дерева путей и расчёта маршрутов до целевых сетей на маршрутизаторах, не входящих в зону, являющуюся источником.

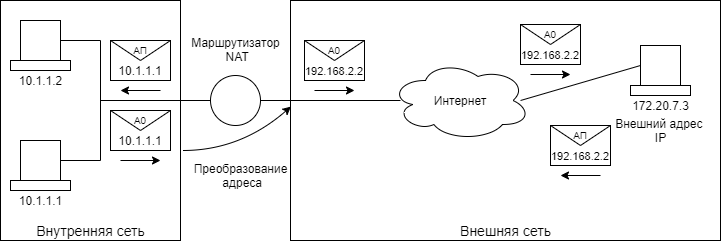
**Служба преобразования сетевых протоколов**

Служба преобразования сетевых адресов (Network Address Translation – NAT) позволяет осуществлять эффективное использование зарегистрированных IP-адресов. Служба NAT представляет с собой способ использования одних и тех же IP-адресов (“частных” IP-адресов) в нескольких внутренних подсетях. Тем самым уменьшается потребность в зарегистрированных IP-адресах. Устройство из “внутренней” сети, в которой используются частные адреса, посылает пакет через маршрутизатор NAT, где осуществляется преобразование внутренних адресов в официально зарегистрированные IP-адреса. Преобразование внутренних IP-адресов в зарегистрированные позволяет передавать пакеты по открытым сетям.

**Терминология службы NAT:**

1. Внутренний локальный IP-адрес – это IP-адрес, назначенный узлу во внутренней сети. Такой адрес должен быть локально уникальным (внутри своей сети) и может быть выбран из частного адресного пространства другой организации (фирма по сетям организует сети на нескольких предприятиях, при этом IP-адреса устройств в этих сетях повторяются);
2. Внутренний глобальный IP-адрес – зарегистрированный IP-адрес (назначенный провайдером), который представляет один или несколько внутренних локальных IP-адресов во внешнем адресном пространстве. То есть один зарегистрированный IP-адрес может соответствовать группе адресов внутри локальной сети;
3. Внешний IP-адрес – IP-адрес устройства в глобальной сети;
4. Запись преобразования – запись в таблице NAT, которая отображает один IP-адрес в другой (внутренний локальный IP-адрес во внутренний глобальный IP-адрес).

**Пример функционирования службы NAT, поясняющий введённые термины:**



Технология NAT позволяет использовать незарегистрированные (“внутренние”) IP-адреса при подключении к открытой сети (глобальной) передачи данных.

Маршрутизатор NAT располагается на границе тупиковой сети и преобразует внутренние локальные IP-адреса (незарегистрированные) в зарегистрированные IP-адреса перед отправкой пакетов во внешнюю сеть.

**Целесообразность использования службы NAT:**

1. При необходимости сохранения существующей схемы адресации с “внутренних” IP-адресов внутри ЛС при подключении её к Интернету; При этом необходимо поддерживать новую схему адресации, назначенную провайдером для устройств внутри ЛС (нет необходимости перестраивать и отлаживать схему адресации внутри ЛС);
2. Устройство внутри ЛС не имеют назначенных глобально уникальных IP-адресов (нехватка адресного пространства), но все ЛС (все её устройства) должны быть подключены к Интернет.

**Преимущества использования NAT:**

1. Сохранение отлаженной схемы адресации внутри ЛС при подключении её к Интернет;
2. Увеличение адресного пространства за счет “дублирования” уже использованных адресов в глобальной сети адресации в локальной сети с их последующим преобразованием в один общий IP-адрес;
3. Уменьшение количества случаев наложения адресов, выделенных для использования во внутренней корпоративной сети, на адреса, существующие в глобальной сети.

**Недостатки NAT:**

1. Задержка при передаче пакетов, связанная с необходимостью преобразования IP-адресов.

**Типы преобразований в службе NAT:**

Статическое преобразование – адреса в таблице соответствий конфигурируется пользователем. Внутренние и внешние адреса статически отображаются однозначным образом.

Динамическое преобразование – NAT-маршрутизатор настраивается таким образом, чтобы он знал, какие внутренние адреса должны быть преобразованы, какой диапазон внешних адресов может быть использован. Преобразование при этом осуществляется автоматически.

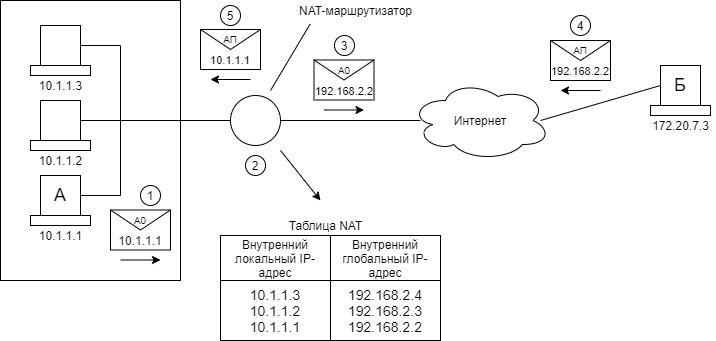
*Поддержка передачи пакетов следующих протоколов:* http, PTP, NFS и т.д.

*Не поддерживает передачу информации:* обновление таблиц маршрутизации, передачу зон DNS, сообщение SNMP.

**Использование NAT для сохранения схемы адресации внутри ЛС**

В этом случае NAT осуществляет тождественные преобразования внутренних созданных ранее адресов устройств в ЛС в выделенные в процессе подключения к Интернету IP-адреса.

Таблица NAT обязательных тождественных преобразований настраивается статически.



**Алгоритм преобразования и пересылки пакетов:**

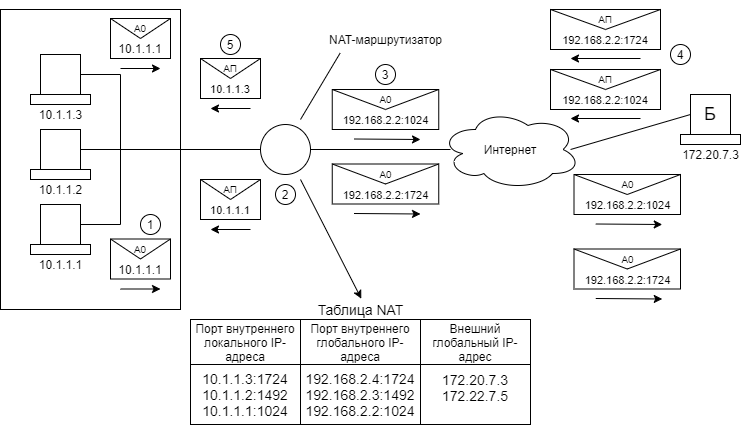
1. Узел А (10.1.1.1) открывает соединение с узлом Б путем посылки запроса;
2. NAT-маршрутизатор сопоставляет полученный в запросе адрес источника (узла А – 10.1.1.1) с содержимым статической таблицы NAT. Если соответствие найдено, выполняется преобразование, пакет отсылается во внешнюю сеть. Узел Б ответит узлу А, используя его внутренний глобальный IP-адрес (192.168.2.2);
3. Если запись для “внутреннего” IP не найдена, маршрутизатор выделяет внутренний глобальный IP-адрес для соответствующего внутреннего локального IP, запись таблицы NAT сохраняется.

Используемые записи в таблице NAT – простые записи.

**Использование NAT для расширения адресного пространства.**

В этом случае один глобальный внутренней адрес может быть использован для одновременного представления внутренних локальных адресов.

Преобразование осуществляется динамически. В NAT-таблице комбинация адреса и порта (уникального кода) делает каждый глобальный IP-адрес уникальным. Идентификация обобщенного (одного) внутреннего глобального IP-адреса и внутреннего локального адреса осуществляется по номеру порта, который присваивается динамически. В процессе преобразования используется таблица расширенных записей.



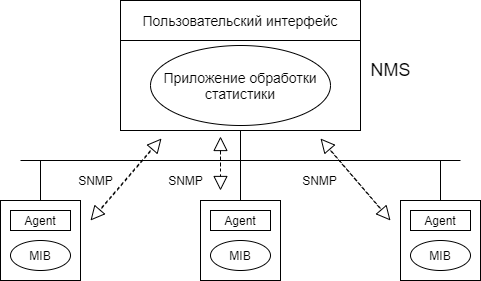
При конфигурации службы NAT выделяется только один глобальный IP и некоторое количество локальных. Генерация N порта происходит автоматически.

**Протокол SNMP (cлужба контроля и управления работой сети)**

Служба SNMP (Simple Network Management Protocol) осуществляет сбор статиcтической информации по работе сети. Пользуясь информацией SNMP (загруженность маршрутизаторов, показатель числа пакетов в одну секунду, коэффициент сетевых ошибок) администраторы обнаруживают проблемы в сети и управляют её производительностью.

**Модель управления в SNMP**

Протокол SNMP имеет клиент-серверную архитектуру. На клиентской стороне действуют агенты SNMP (клиенты – управляемые устройства). Агенты осуществляют сбор информации об управляемых устройствах, на которых они работают. Агенты передают статистическую информацию о работе управляемых устройств агентам управления сетями (Network Management System – NMS), действующих на серверной стороне. Обмен сообщениями между NMS и агентами осуществляется за счёт использования сообщений SNMP.



Например, агентам на управляемом устройстве могут отслеживаться следующие параметры:

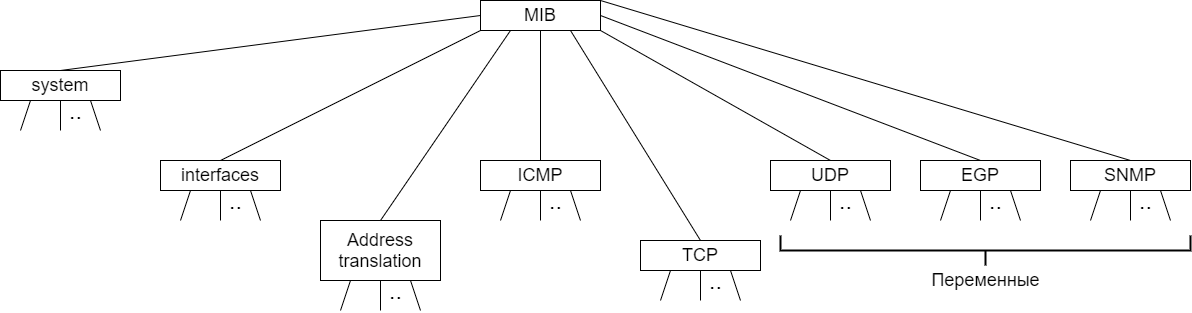
1. Число байтов и пакетов, входящих и исходящих из данного устройства;
2. Максимальная длина очереди на выходе (для маршрутизаторов);
3. Число определённых видов полученных сообщений о неисправности (протокола ICMP).

**База данных управления**

Управляемые объекты (статистика по работе устройства) содержатся в информационной базе управления (Management Information Base – MIB). MIB представляет собой иерархическую базу данных, нижними элементами которой являются переменные, хранящие значение статистики.

Информация, хранимая в MIB определяет, статистику по функционированию служб и протоколов на управляемых устройствах, а также переменные, характеризующие систему.

**Структура верхнего уровня БД MIB:**



Таким образом, MIB состоит из разделов, каждый из которых имеет определённую совокупность переменных (параметров), значения которых характеризуют общую информацию о системе и статистику её работы.

Информационной база MIB – это БД, которая обслуживается агентом, менеджер может запросить информацию из БД или записать её туда. SNMP на сервере (менеджер) обращается к переменной (на агенте, чтобы получить её значение или установить его). Обращение осуществляется только к листовым узлам дерева MIB.

**Структура сообщения SNMP**

Структура сообщения – имя сообщества и часть данных. Имя сообщества назначает среду доступа для NMS агентов, которые используют это имя. Устройство, не поддерживающее правильное имя, исключается из операций SNMP (сервер или клиент).

Строка сообщества – это строка символов, используемая как пароль (то есть сообщество используется SNMP как средство запрета доступа к статистической информации).

Информационная часть состоит из кода команды (команды получения значений из MIB-агента или установления значений в MIB) и связанных с ней операндов (переменных), значения которых отображают статистику, используемую для управления.

**Формат SNMP-сообщения**

|  |  |  |
| --- | --- | --- |
| Имя сообщества | Код команды | Операнды |

**Команды протокола SNMP**

При обмене данными менеджер и клиент обмениваются сообщениями 5 типов:

1. Запросить от агента значение его переменной из MIB – команда:

*GET-REQUEST <имя\_параметра>*

1. Запросить следующую переменную из MIB-агента – команда:

*GET-NEXT-REQUEST <имя\_переменной>*

1. Установить значение переменной в MIB-агенте – команда:

*SET-REQUEST <имя\_параметра | значение>*

1. Выдать менеджеру SNMP запрошенную им переменную – команда:

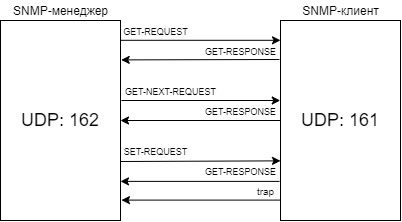
*GET-RESPONSE <имя\_переменной | значение>*

(данная команда позволяет выполнить ответ на команды *GET-REQUEST,* *GET-NEXT-REQUEST*).

1. Уведомить сервер SNMP об аварийной ситуация на агенте – команда:

*trap <код\_ошибки>*

**Схема обмена сообщениями между клиентом и сервером SNMP.**



TRAP – ошибки в интерпрет. сообщ. *GET*, *SET* и т.д.

**Особенность функционирования SNMP:**

1. Одна система может быть как менеджером, так и клиентом SNMP.

**Информационная структура MIB (на агенте).**

MIB состоит из разделов и содержащихся в них переменных, характеризующих статистику. Разделы:

1. *system* – отображает общую информацию о системе (имя домена узла, идентификаторы предоставляемых сервисов – прикладной, сетевой, канальный);
2. *interface* – содержит информацию об интерфейсах системы. Переменные:
3. *itType* – тип интерфейса (*Ethernet, Tokenring, PPR* и т.д.);
4. *itMTU* – MTU интерфейса;
5. *itSpeed* – скорость в битах в секунду.
6. *at* – группа трансляции адресов (осуществляет тождественное отображение IP HWA-адресов, назначенных соответствующим интерфейсом. Переменные:
7. *atPhysAddress* – физический адрес интерфейса;
8. *atItIndex* – номер интерфейса;
9. *atNetAddress* – IP интерфейса.

(Переменные задаются подряд в одной команде).

1. группа IP – характеризует работу сетевого уровня модели OSI по приему и передаче дейтаграмм. Переменные:
2. *ipInReceives* – количество дейтаграмм, полученных со всех интерфейсов;
3. *ipForwDatagrams* – количество дейтаграмм, для которых была сделана попытка перенаправить (маршрутизировать);
4. *ipOutNoRouters* – количество дейтаграмм, отброшенных из-за того, что не был найден маршрут;
5. *ipRouteProto* – протокол маршрутизации (8-RIP, 13-OSPF, 14-BGP и т.д.);
6. *ipRouteAge* – количество секунд после последнего обновления маршрута.
7. группа TCP контролирует работу TCP-протокола – количество принятых с ошибками пакетов, количество повторно переданных пакетов. Данные параметры могут характеризовать состояние канала передачи данных. Переменные:
8. *tcpRetransSeqs* – полное количество повторно переданных сегментов;
9. *tcpInErrs* – количество сегментов, принятых с ошибками.

Просмотр значений переменных, характеризующих статистику работы устройств (маршрутизаторов) осуществляется в ОС IOS CISCO командой:

*slow SNMP MIB <имя\_переменной>*