32.DNS. Динамические обновления; нотификации об изменениях; инкрементальные обновления.

**Динамические обновления:**

Причина появления: наличие динамических систем управления адресами DHCP

DNS Dynamic Update – RFC 2136

Позволяет внешним авторизованным источникам менять адреса.

Принцип работы: Авторизованный клиент посылает DNS серверу запрос об изменении информации о ресурсной записи. Если сервер не первичен, то запрос пересылается наверх. Когда дошли до первичного DNS, то первичный сервер модифицирует свою базу данных, изменяет серийный номер зоны. Через какое то время вторичные серверы получают от первичного обновленную информацию.

**Нотификация об изменениях:**

Асинхронное информирование об изменении информации.

Раньше вторичные серверы получали информацию от первичных синхронно, то есть, в соответствии с записью зоны синхронно получали обновление. Это не всегда удобно, хотелось бы получать уведомления об изменениях, данных от первичного сервера как можно раньше. Например, в случае динамических обновлений, надо чтобы эту информацию быстро узнал интернет.

DNS Notify – RFC 1996

Если включена эта опция, то первичный сервер посылает обновление всем известным ему вторичным серверам. Далее вторичные посылают другим вторичным. Каждое сообщение об нотификации подтверждается. Когда дошли «до конца» , вторичные серверы осуществляют запрос зоны как при периодическом обновлении.

**Инкрементальная передача зон (инкрементальные обновления):**

Классический DNS передавал всю зону. То есть если что-то изменилось, то первичный передаёт вторичному весь огромный файл зоны, сколько бы там записей не было. При динамических обновлениях происходит слишком много трансферов зон.

Для уменьшения передаваемой информации, передаётся только инкремента (приращение) зоны, вместо всей зоны. В запросе вторичный указывает серийный номер зоны, которая у него есть, а первичный вычисляет разницу содержимого и высылает её.

Тип запроса изменяется на IXFR.

33. Автоматизированная настройка параметров. Протоколы BOOTP и DHCP.

Управление сетевыми параметрами.

Два подхода:

1. Ручная настройка: вручную задаются IP адреса, маски, DNS серверы и прочие маршрутизаторы.
2. Автоматизированная настройка: перечисленные выше параметры настраиваются автоматически либо автоматизировано.

Идея автоматизированной настройки пришла от того, что когда имеется большая компьютерная сеть, то снабжать всех пользователей инструкцией как и что надо настроить, для администратора становится очень сложно. Тем более при изменении, какого либо параметра придется объяснять ещё раз. Появилась идея сделать так, чтобы компьютер сам настраивал параметры забирая их из сериализованного хранилища.

Для этого было разработано несколько протоколов:

- RARP (Reverse Address Resolution Protocol) – устаревший протокол

- BOOTP (Bootstrap Protocol) – устаревший протокол

- DHCP (Dynamic Host Configuration Protocol)

**RARP:**

Описан в RFC 903

Принцип работы: в сети должен быть RARP сервер, при включении в сеть клиент делает запрос на этот сервер. В ответ приходит IP адрес и присваивается сетевому адаптеру. Разделяет один формат пакетов с ARP.

Используется только для получения IP, чего недостаточно в современных сетях.

**BOOTP:**

Описан в RFC 951, 1533, 1542

Принцип работы: в сети должен быть BOOTP сервер. В момент, когда бездисковая или автоконфигурируемая станция собирается загрузиться, делается широковещательный запрос по протоколу BOOTP к этому серверу. BOOTP :

1. IP адрес.
2. Маска сети.
3. Маршрутизатор по умолчанию.

Выдается достаточно параметров, чтобы компьютер стал автономным, и всё остальное, при необходимости, мог загрузить из сети.

Протокол позволяет делать цепочку BOOTP ретрансляторов до BOOTP сервера. То есть, если в вашей сети нет BOOTP сервера, то отправляется запрос к BOOTP ретранслятору, ретранслятор отправляет запрос дальше, пока не будет достигнут BOOTP сервер и не будет получена информация.

Реализация: у BOOTP сервера имеется статическая таблица соответствий MAC адресов и параметров узла. То есть, администратор, когда хочет сконфигурировать BOOTP сеть, то заходит на сервер, и, зная все компьютеры в сети, для каждого из них прописывает необходимые параметры.

**DHCP:**

Протокол разработан компанией Microsoft

Описан в RFC 2131, 2132

Разработан не как самостоятельный протокол, а как расширение BOOTP протокола. Все новые, получаемые дополнительно поля, располагаются в поле опций протокола BOOTP.

Поддерживается 3 режима:

1. Ручное распределение адресов: точно такое же, как в BOOTP, администратор пишет соответствие MAC адресов и выдаваемых параметров.
2. Автоматическое распределение: формируется пул адресов, которые раздаются клиентам по мере их подключения. После чего выданные данные заносятся в таблицу, и в будущем никому кроме выбранной станции, адрес не выдается. Количество подключаемых станций соответствует количеству свободных адресов. Не подходит для современного интернета.
3. Динамическое распределение адресов: адреса не закрепляются за конкретным устройством навсегда, выдаются в аренду, по истечению срока аренды, адрес становится вновь свободным и может быть выдан любому устройству, которое захочет подключиться.

Информация передается через опции протокола BOOTP.

Типы опций:

1. Базовые параметры:

- маска сети

- default gateway (маршрутизатор по умолчанию)

- DNS сервер

- название вашего узла (текстовое)

- название домена по умолчанию

1. Параметры узла:

-IP forwarding

-Default TTL

- и так далее

1. Параметры интерфейсов

- MTU (maximum transfer unit)

- Broadcast

- Static routes

1. Параметры TCP:

- TCP default TTL

- KeepAlive time

- И тому подобное

1. Параметры приложений:

- NIS, NIS+

- WINS

- POP3, SMTP, NNTP

1. Параметры аренды:

- Запрашиваемый IP

- Срок аренды

- Идентификатор сервера, который сделал эту аренду

- и тому подобное

Для управления арендой есть утилита ipconfig

37. Управление в сетях. Протокол SNMP.

Модель систем управления компьютерными сетями:

Классическая модель ISP/OSI ( ITU – T X.700 ) рассматривает 5 задач управления:

1. Управление конфигурацией
2. Управление учетом использования ресурсов
3. Управление неисправностями
4. Управление эффективностью
5. Управление защитой

Архитектура систем управления сетями:

1. Management entities: управляющие элементы, управляющие программы, в задачи которых входит управление другими устройствами.
2. Managed devices: управляемые устройства, которые должны в своём составе иметь:

* Управляющий агент, через которого management entities управляет устройством
* Управляющая БД, которая проецирует сетевые параметры на стандартные схемы представления параметров в современных протоколах

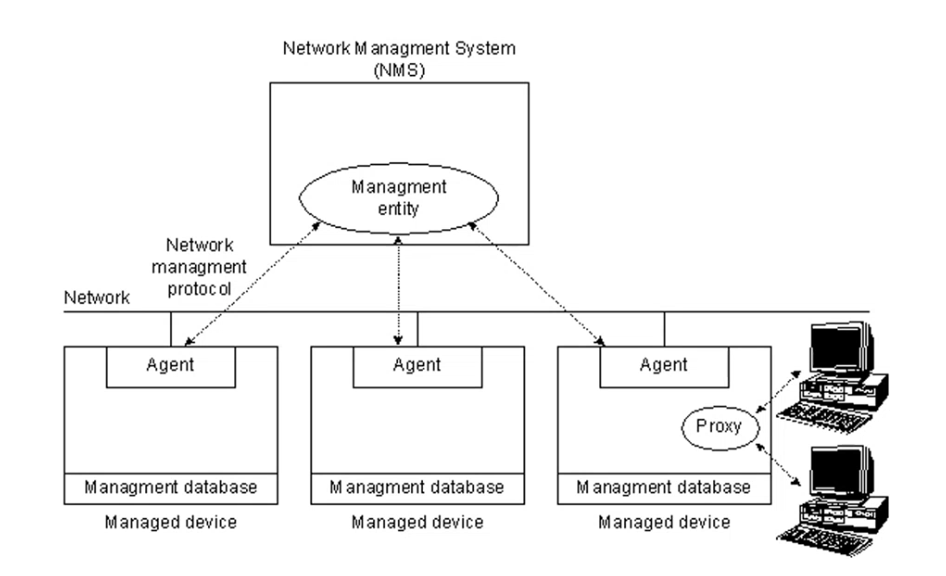
1. Management proxies: посредники управления, используются когда управляемое устройство очень «глупое» , то есть само не может реализовывать элемент архитектуры, тогда для этого нужен посредник.

* Управляющий агент
* Управляющая БД

1. Management environment: среда управления, та компьютерная сеть, по которой происходит передача управляющих команд. В современных компьютерных сетях, это одна и та же среда, что используется для прикладных задач

- Обычно используется прикладная КС

Архитектура систем управления сетями



Основная задача, которая реализуется – задача управления конфигурацией.

Управление конфигурацией :

1. Контролирование информации о сетевой и системной конфигурации.
2. Управление программными компонентами
3. Управление аппаратными компонентами

Реализация архитектуры в TCP/IP сделана с помощью протокола SNMP

**SNMP (Simple Network Management Protocol):**

Первый стандарт 1988 г. Является основным протоколом управления в TCP/IP.

Стандарты:

- RFC 1157

-RFC 1441-1450, RFC 1901-1910, RFC2263-2265, RFC 2273-2275, RFC 2573-2575

-RFC 3411 – 3418

Существует три основных версии протокола:

1. SNMPv1,
2. SNMPv2 (v2c, v2p,v2u – реализуют разные методы аутентификации)
3. SNMPv3

Цель: хотим с помощью одного универсального протокола управлять разными устройствами: коммутаторы, маршрутизаторы, серверы, NAT серверы, web серверы, системы защиты информации и так далее. Поэтому необходимо общее представление управляющих команд.

Для этого была придумана общая среда – база управления MIB (management information database). Эта база представляет все задачи управления в виде дерева переменных, которые одинаковы для всех устройств, которые используются. Также SNMP устройства должны представлять информацию в виде этого дерева переменных. То есть SNMP агент на управляемых устройствах должен понимать это дерево переменных, и в таком виде отдавать данные SNMP менеджеру и отвечать в таком виде на запросы SNMP менеджера.

Концепция переменных:

1. Операция чтения переменных
2. Операция записи переменной
3. Операция чтения следующей переменной (в случае если переменная списочная)

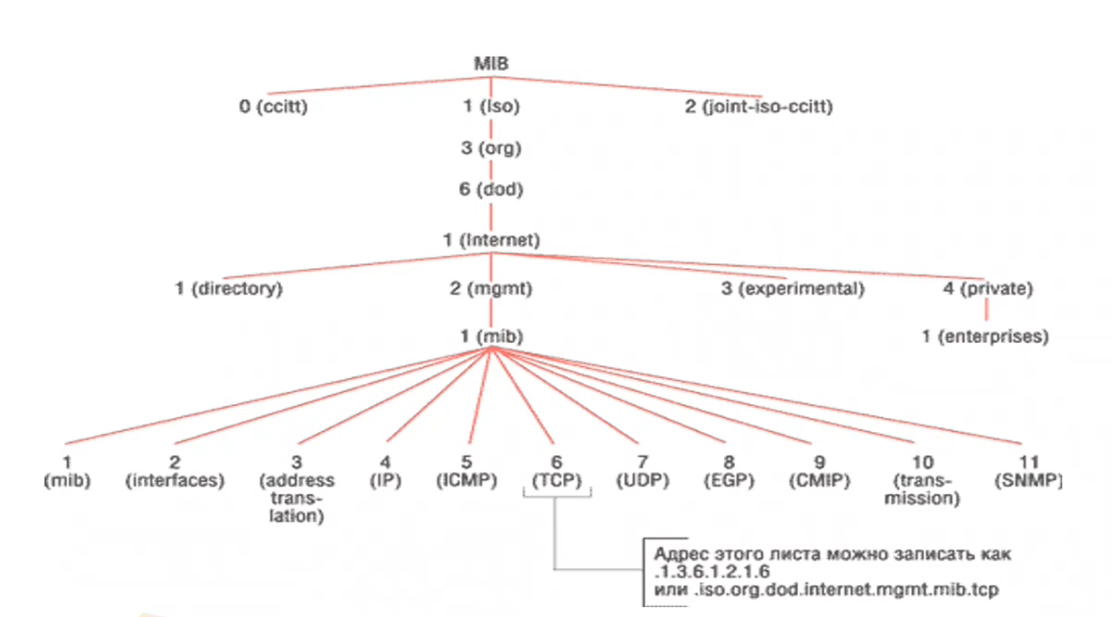
Переменные:

1. Скалярные
2. Таблицы

Протокол SNMP:

1. Независим от транспортного протокола, однако обычно используют UDP
2. На уровне представлений используется нотация ASN.1 Basic Encoding Rules, для кодирования сообщений для передачи по сети.
3. Порт управления – 161 (передача команд от менеджера к агенту)
4. Порт уведомлений (snmp trap(ловушки)) – 162 (сообщение менеджеру о чём-то, что произошло в управляемом устройстве)
5. Концепция «ловушек» - при нештатной ситуации менеджер оповещается об этом

Часть дерева системы MIB



Используется не только для интернета, а во всех телекоммуникациях.

Объект SNMP:

1. Текстовое имя
2. Тип объекта

- Network address

- IP address

- Counter (счетчик со сбросом)

- Gauge (счетчик без сброса)

- Ticks (для измерения временных параметров

- Opaque (строки)

1. Определение объекта
2. Доступ к объекту

- Read-only

- Write-only (для подачи управляющих команд)

- Read-write

- No-access

1. Статус

- Обязательный (переменная должна быть реализована в любом устройстве)

- Необязательный

- Устаревший

Безопасность протокола SNMP:

1. Модель безопасности на основе сообществ

- Вводится понятие сообщества, нешифруемая строка, которая используется в запросе менеджеру, если строка совпадает, то можем подавать управляющие команды.

- Используется:

- SNMPv1

- SNMPv2c

1. Модель безопасности на основе сторон

- Вводится понятие стороны:

- идентификатор стороны

- логический сетевой адрес

- протокол аутентификации и связанные параметры

- протокол шифрования и связанные параметры

- Используется:

- SNMPv2p

1. Модель безопасности на основе пользователей

- Вводится понятие пользователь

- идентификатор пользователя

- протокол аутентификации

- протокол шифрования

- ключи шифрования

- ключи аутентификации

- Используется:

- SNMPv2u

- SNMPv3