Этот цикл статей направлен в основном на администраторов. Здесь я буду рассказывать о различных аспектах построения и администрирования сетей. Само собой напрашивается начать с основ.

Основным я считаю правильное построение сети. От правильного построения зависит вся её работа. Одной из главных проблем в этом является оптимизация трафика. Сеть нужно построить так, чтобы нагрузка на всех участках сети не приближалась к критической отметке.

Любой администратор должен учитывать, что трафик в сети складывается неравномерно. В связи с этим, необходима оптимизация отдельных участков. Для этого необходимо проделать небольшую аналитическую работу. Как известно, большинство пользователей выполняет однородные задачи, и работают с одним и тем же сервером и с одними и теми же компьютерами, поэтому необходимо выделить таких пользователей в отдельную группу. Администратор должен выявить такие группы пользователей и предоставить им отдельную сеть. Основным признаком показывающим, что группу надо выделить в отдельную сеть является больший поток информации между членами группы чем с другими группами.

Такое разделение легко провести даже с учётом должностей (если сеть проводиться на производственном предприятии). В этом случае можно выделить отдельную сеть для бухгалтеров, отдельную сеть для кадровиков и так далее.

Для того, чтобы разделить сеть на подсети используются коммутаторы и мосты. С их помощью группа изолируется. В этом случае весь трафик, проходящий внутри группы, не выходит наружу. За пределы сети пакет данных может выйти, только если он направлен какому-либо компьютеру, находящемуся вне данной сети.

Если ты решил использовать в качестве изолятора коммутаторы, то сеть будет изолирована так как описано выше. А если используются мосты, то за пределы групп будут выходить ещё и широковещательные пакеты. Всё это связано с тем, что мосты и коммутаторы работают по протоколам канального уровня. В этих протоколах не определены понятия подсетей или сегментов.

Ещё одним недостатком канального уровня является то, что топология сети должна быть простой. Если в сети есть петли, то пакет не сможет дойти до адресата.

От этих недостатков позволяет избавиться сетевой уровень. Это не значит, что всё, что я говорил можно выбросить в мусорный ящик. Сетевой уровень работает поверх канального, поэтому он просто добавляет новые возможности к канальному. С помощью сетевого уровня можно доставлять пакеты, даже если в сети есть петли. Для реализации сетевого уровня, в сети должен стоять маршрутизатор, с помощью которых отдельные сети объединяются в одну.

Маршрутизатор выбирает маршрут для проходящего через него пакета в соответствии с конфигурацией сети и времени прохождения пакета. Маршрут пакета - это последовательность маршрутизаторов, которые проходит этот пакет до пункта назначения.

Хочу ещё немного рассказать о работе маршрутизаторов. Я раньше много раз задавал себе вопрос: "Как маршрутизатор узнаёт о конфигурации сети?". Ответ оказался прост. В нм хранится небольшая база данных, с помощью которой он узнаёт, куда направлять пакет с

определённым адресом. Если в этой базе данных отсутствует информация о каком-либо IP адресе, то он отправляет этот пакет другому маршрутизатору, выбранному по умолчанию. Так происходит пока пакет не найдёт своего адресата.

Модель взаимодействия открытых систем:

Модель взаимодействия открытых систем (OSI - Open Systems Interconnection -) разработанна Международной Организацией по Стандартам (ISO - International Organization for Standardization). В соответствии с этой моделью, сетевое взаимодействие делятся на семь уровней. Первые три уровня обеспечиваются оборудованием, таким как маршрутизаторы, концентраторами, мостами и др. Последние три обеспечиваются операционной системой или приложением. Четвёртый уровень является промежуточным.

- *Физический уровень.* - передача битов по физическим каналам (коаксиальный кабель, витая пара, оптоволоконный кабель). Здесь определяются характеристики физических сред и параметры электрических сигналов.
- *Канальный уровень* - передача кадра данных между любыми узлами сетях типовой топологии или соседними узлами произвольной топологии. В качестве адресов на канальном уровне используются MAK-адреса.
- *Сетевой уровень* - доставка пакета любому узлу в сетях произвольной топологии. На этом уровне нет никаких гарантий доставки пакета.
- *Транспортный уровень* - доставка пакета любому узлу с любой топологией сети и заданным уровнем надёжности доставки. На этом уровне имеются средства для установления соединения, буферизации, нумерации и упорядочивания пакетов.
- *Сеансовый уровень* - управление диалогом между узлами. Обеспечена возможность фиксации активной на данный момент стороны.
- *Уровень представления* - здесь возможно преобразование данных (шифрация, компрессия).
- *Прикладной уровень* - набор сетевых сервисов (FTP, E-mail и др.) для пользователя и приложения.

Сетевой уровень - это продолжение развития канального уровня. В нё сохранены все реквизиты предка и добавлены новые, позволяющие более эффективно передавать пакеты данных. Так, например, в заголовок был добавлен "номер пакета", что позволяет фрагментировать и дефрагментировать пакеты. Помимо этого была добавлена информация о загруженности сети, что позволяет согласовать скорость передачи пакетов.

Основным нововведением был ввод времени жизни пакета. На первый взгляд, этот параметр очень сильно увеличивает вероятность пропажи пакетов, но на самом деле он обезопасил работу в сети. Это связано с ошибкой заложенной с самого начала в работу маршрутизаторов. Ошибка заключалась в том, что возникали моменты, когда пакет заклинивал, гуляя туда суда между двумя маршрутизаторами.

В качестве адресов в сетевом уровне используются уже не MAK-адреса, а пара чисел: номер сети и номер компьютера в этой сети. Это означает, что сетевой уровень поддерживает межсетевые связи на уровне адресации. Внутри одной сети передача пакетов осуществляется на канальном уровне, а для передачи пакетом между сетями используется сетевой.

Сегодня я решил рассказать об адресации в сетях IP. Я не буду говорить как важно выбрать правильную адресацию. Ты и сам поймёшь, когда работа твоей сети парализуется, из-за досадной ошибки.

Для адресации используются три типа адресов:

- Физический или MAK-адрес. Эти адреса состоят из 6-и байтов и устанавливаются производителем в маршрутизаторы, сетевые адаптеры. Первые три идентифицируют фирму производителя, а вторые три назначаются этим производителем уникальным образом. В сетях типа x.25, frame relay эти адреса назначаются администратором. Эти адреса используются на канальном уровне.
- IP-адрес. Он состоит из четырёх байт. Такие адреса назначаются администратором. Чуть ниже мы рассмотрим этот тип адреса более подробно, скажу только, что он используется на сетевом уровне.
- Символьный тип, например адрес моей странички. Скорей всего он выглядит у тебя в браузере как символьная строка (я не думаю, что ты использовал IP адрес). В принципе, это просто псевдоним для IP адреса. Вводить понятные человеку слова легче, чем бесполезные цифры.

А начнём мы рассматривать адресацию с IP-адреса. Не знаю почему, но мне так захотелось. О MAK-адресах мы уже говорили в прошлом номере, поэтому о нём я буду уже только упоминать. А вот IP и символьные адреса мы рассмотрим подробненько.

IP-адрес делиться на две части: номер сети и номер узла. Сколько бит означает номер сети, а сколько номер узла определяется с первых битов адреса. Компьютер или маршрутизатор может иметь несколько IP-адресов, что позволяет ему входить в несколько сетей одновременно.

Размер IP-адреса равен четырём байтам, их принято записывать отдельно с разделением их точкой. IP-адрес может быть записан в двоичной, шестнадцатеричной, в десятичной и др. системах исчисления. Есть пять видов IP, все они представлены на рисунке 1.

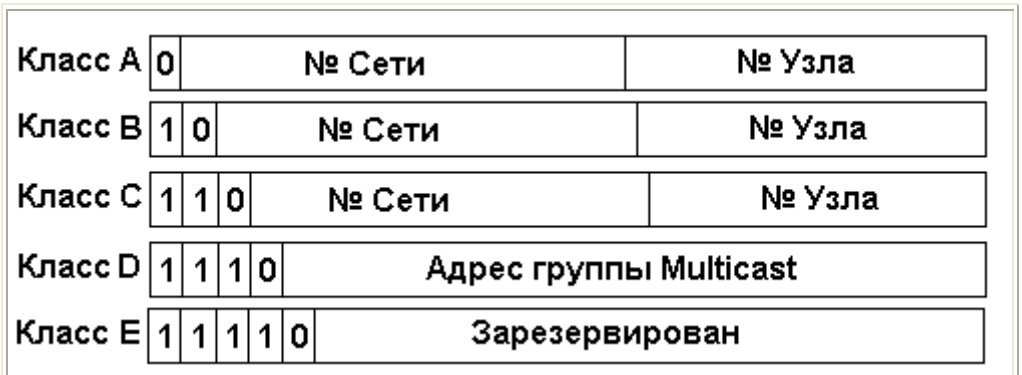

Рис 1. Структуры 5-и видов IP-адреса

Как видно из рисунка, адрес делиться на три части:

1. Идентификатор, показывающий к какому виду относится этот адрес.
2. Номер сети
3. Номер узла в этой сети

Количество байт отведённых под адрес сети и адрес узла зависит от идентификатора. Рассмотрим каждый вид в отдельности:

A. Адрес начинается с нуля. Номер сети занимает 1 байт, номер узла -3 байта. Номера сети изменяются в диапазоне от 1 до 126. Нулевой номер не используется, а 127-1 зарезервирован. Этот вид предназначен для построения самых больших сетей, число узлов должно быть в интервале от 216 до 224. Диапазон значений: от 01.0.0.0 до 126.0.0.0
B. Адрес начинается с 10. Номер сети занимает 2 байт, номер узла -2 байта. Это сети средних размеров с числом узлов от 28-215. Диапазон значений: от 128.0.0.0 до 191.255.0.0
C. Адрес начинается с 110. Номер сети занимает 3 байт, номер узла 1 байт. Число узлов в сети не больше 28. Диапазон значений: от 192.0.1.0 до 223.255.255.0
D. Адрес начинается с 1110. Это особый вид сетей - multicast. Пакеты, в которых в качестве адреса назначения стоит такой адрес, являются широковещательными и должны быть доставлены всем узлам указанной сети. Диапазон значений: от 224.0.0.0 до 239.255.255.255
E. Адрес начинается с 1110. Зарезервирован. Диапазон значений: от 240.0.0.0 до 247.255.255.255

Теперь рассмотрим специальные адреса:

Первый тип адресов будет не маршрутизируемый. Это адреса, которых просто не может быть в интернете. Они зарезервированы для использования в локальных сетях. Например, адреса начинающиеся на 192.168.x.x ты можешь смело назначать компьютерам своей сети. Если даже эти компьютеры будут подключены к сети Internet, то конфликтов не будет. Если ты назначишь компьютеру адрес уже существующий в Internet, то возникнут конфликты. Поэтому пользуйся немаршрутизируемыми адресами, так ты избавишся от возможных проблем.

Если адрес получателя состоит из двоичных нулей, то он означает адрес узла сгенерировавшего пакет. Если номер сети адреса получателя состоит из нулей, то получатель и отправитель находятся в одной и той же сети. Если номер узла адреса получателя состоит из единиц, то пакет рассылается всем узлам из сети отправителя.

IP-адреса узлов могут назначаться администратором вручную, а могут назначаться динамически. Если ты решишь настраивать всё самостоятельно, то приготовься сделать свои IP-адреса статическими. Переконфигурация таких сетей отнимает много сил и нервов.

Для динамического конфигурирования был разработан специальный протокол DHCP (Dynamic Host Configuration Protocol). Этот протокол может настраивать компьютер пользователя тремя способами: ручной, статический и динамический.

Ручной. При таком способе настройки администратор должен настроить соответствие IP-адресов физическим адресам. Эта информация будет передаваться клиенту на его запросы.

Статический. Администратор указывает DHCP серверу диапазон допустимых IP-адресов. При первом соединении, клиент получает адрес из этого диапазона, а сервер устанавливает соответствие выданному IP-адресу физический адрес устройства-клиента.

Динамический. Адрес в этом случае выдаётся как и при статическом способе из допустимого диапазона, но на определённое время. В этом случае можно построить сеть, в которой количество устройств значительно превышает количество допустимых IP-

адресов. Примером может служить твой провайдер, у которого клиентов намного больше чем допустимых адресов. Ему не зачем держать больше адресов, чем входных телефонных линий и сетевых устройств в его сети.

Достоинство этого протокола в том, что администратор сидит за сервером и управляет всем процессом адресации. Ему не надо бегать от компьютера к компьютеру.

С IP-адресацией покончено. Я не смог рассказать всего, и наверно ещё не раз вернусь к этой теме, но для первого знакомства этого достаточно. Теперь я перейду к рассмотрению символьному типу адресов.

Символьное представление адреса было придумано для того, чтобы их легче было запоминать. Для маршрутизации таких адресов используется служба DNS (Domain Name System). На самом деле DNS представляет собой большую распределённую базу данных, в которой каждому символьному имени поставлен в соответствие IP-адрес. Как-то сложно я сказал, попробую объяснить на примере. Когда ты вводишь символьный адрес любой странички, то браузер отправляет его DNS серверу, который находит в базе соответствующий IP-адрес и возвращает его. После этого браузер обращается к страничке уже по IP-адресу.

Распределённость базы заключается в том, что один DNS сервер не может вместить в себя IP всех символьных адресов, потому что он не сможет обрабатывать запросы всей сети. DNS-серверы распределены по всему миру, на каждом из которых находится какая-то часть от этой базы. Если один сервер не в состоянии обработать символьный адрес, то он запрашивает у другого сервера, и так до тех пор, пока не найдётся соответствующий IP-адрес.

Сегодня я возвращаюсь к теме сетей. Нам предстоит изучить Microsoft TCP/IP. В самой первой статье по TCP/IP я уже говорил про модель OSI. MS как всегда выделилась, и сделала свою реализацию этой модели. Сегодня я напомню тебе OSI и расскажу про MS TCP/IP.

Напомню, что модель OSI (Open Systems Interconnection - взаимодействие открытых систем) предлагает основу программистам для разработки сетевых протоколов. Для того, чтобы программисты не писали протоколы "Кто в лес, кто по дрова", ассоциация ISO разработала спецификацию для протоколов передачи данных по сети. Эта модель описывает уровни и функции, которые должны присутствовать в протоколе. Но при реализации любого протокола не обязательно дословно следовать OSI, потому что это только предложение, которого желательно придерживаться.

Модель состоит из семи уровней:

- *Физический уровень.* - передача битов по физическим каналам (коаксиальный кабель, витая пара, оптоволоконный кабель). Здесь определяются характеристики физических сред и параметры электрических сигналов.
- *Канальный уровень* - передача кадра данных между любыми узлами сетях типовой топологии или соседними узлами произвольной топологии. В качестве адресов на канальном уровне используются MAK-адреса.
- *Сетевой уровень* - доставка пакета любому узлу в сетях произвольной топологии. На этом уровне нет никаких гарантий доставки пакета.

- *Транспортный уровень* - доставка пакета любому узлу с любой топологией сети и заданным уровнем надёжности доставки. На этом уровне имеются средства для установления соединения, буферизации, нумерации и упорядочивания пакетов.
- *Сеансовый уровень* - управление диалогом между узлами. Обеспечена возможность фиксации активной на данный момент стороны.
- *Уровень представления* - здесь возможно преобразование данных (шифрация, компрессия).
- *Прикладной уровень* - набор сетевых сервисов (FTP, E-mail и др.) для пользователя и приложения.

Как работает протокол по этой модели? Всё начинается с прикладного уровня. Пакет попадает на этот уровень и к нему добавляется заголовок и прикладной уровень отправляет этот пакет на следующий уровень (уровень представления). Здесь ему также добавляется свой собственный заголовок, и пакет отправляется дальше. Так до физического уровня, который занимается непосредственно передачей данных. Физический уровень отправляет пакет.

Другая машина, получив пакет начинает обратный отсчёт. Пакет с физического уровня попадает на канальный. Канальный уровень убирает свой заголовок и поднимает пакет выше (на уровень сети). Уровень сети убирает свой заголовок и поднимает пакет выше. Так пакет подымается до уровня приложения, где остаётся чистый пакет без служебной инфы которая была прикреплена на исходной машине перед отправкой пакета.

MS как всегда пошла своим путём и реализовала модель OSI в TCP/IP по-своему. Здесь, вместо семи уровней есть только четыре. На рис 1 я графически сопоставил модель MS TCP и модель OSI. Не смотря на то, что в MS TCP только четыре уровня, они реализуют все семь. Например, Уровень приложения MS TCP реализует в себе три уровня OSI (см рис 1).

Рис 1. Модель MS TCP

Интерфейс сокетов (я больше люблю выражение WinSock) представляет собой программную реализацию, которая облегчает взаимодействие между приложениями и сетью.

NetBIOS наверно придётся изучать отдельно, потому что это достаточно большая тема и на нём я останавливаться не буду.

Очень интересным тут является интерфейс TDI - это программный интерфейс, который работает между уровнями приложения и транспорта MS TCP. Этот интерфейс позволяет

создавать приложения сеансового уровня, абсолютно ничего не понимая в транспортном уровне.

Далее идут на уровне транспорта протоколы TCP и UDP.

- TCP - отвечает за надёжную передачу данных по сети. Для передачи создаётся виртуальное соединение. После установки связи две машины могут начинать обмен информацией. Каждый TCP-пакет содержит номер порта отправителя и получателя, номер пакета (если данный фрагментированы), контрольную сумму и сами данные. Помимо этого, каждый пакет содержит уникальный номер, который служит защитой от взломов. Этот номер получается случайным образом при установке связи и в течении всего сеанса увеличивается на 1 с каждым последующим пакетом.
- UDP - не устанавливает связи и не обеспечивает надёжной передачи данных. Пакеты просто отправляются машине получатели без каких либо ожиданий подтверждения. Если передаваемые данные фрагментированы и хотя бы один пакет затерялся, то восстановить всю информацию уже невозможно.

Межсетевой уровень обеспечивает маршрутизацию пакетов как внутри сети, так и между сетями. Если ты разбираешься в сетевом оборудовании, то ты уже должен был догадаться об этом, потому что все маршрутизаторы работают на третьем уровне. Помимо маршрутизаторов, на этом уровне могут работать и коммутаторы третьего уровня, но это достаточно дорогое оборудование.

Протокол IP похож на UDP протокол. Он также не устанавливает соединения и для передачи данных использует датаграммы. IP пакет включает:

- Адрес узла отправителя
- Адрес узла получателя
- Идентификатор протокола, который работает поверх IP
- Контрольная сумма
- TTL (Time To Live) время жизни пакета.

Это максимальное количество времени, которое может прожить пакет. Для того, чтобы пакеты не зацикливались в сети из-за ошибочных записей в сети. Если пакет не дошёл до узла назначения за время TTL, то пакет уничтожается. При отправке пакета TTL устанавливается в определённое системой число. Проходя через маршрутизаторы, каждый из них уменьшает это число на единицу. Если число становится равным нулю, то пакет уничтожается.

ARP (Address Resolution Protocol) - это протокол сопоставления адреса. Он также работает на третьем уровне. RARP - это протокол выполняющий обратные действия ARP. Это всё, что тебе нужно знать на этом этапе. Эта тема большая и интересная, поэтому мы поговорим о ней немного позже.

ICMP - это протокол для передачи и получения информации о переданных пакетах. Этот протокол используется в маршрутизаторах для контроля скорости передачи пакетов. Когда моршрутизатор перегружен, то он отправляет ICMP-сообщение, которое убедительно просит узел-отправитель не торопится :) и отсылать пакеты пореже.

IGMP - протокол для управления группами. Его используют узлы для регистрации себя в какой-нибудь группе. Маршрутизаторы используют эту инфу при отправке сообщений предназначенных целой группе.

А вот теперь мы поговорим про ARP (Address Resolution Protocol). Любой пакет передаваемый по сети должен содержать в себе MAC - адрес (аппаратный адрес сетевого устройства). Этот адрес прошит в производителем в сетевое устройство. Если ты хочешь узнать MAC адрес своей карты, то запусти Ipconfig.exe или winipconfig.exe из директории Windows. Для winipconfig.exe нажми кнопку "Сведения>>" и ты сможешь увидеть окно, как на рисунке 2.

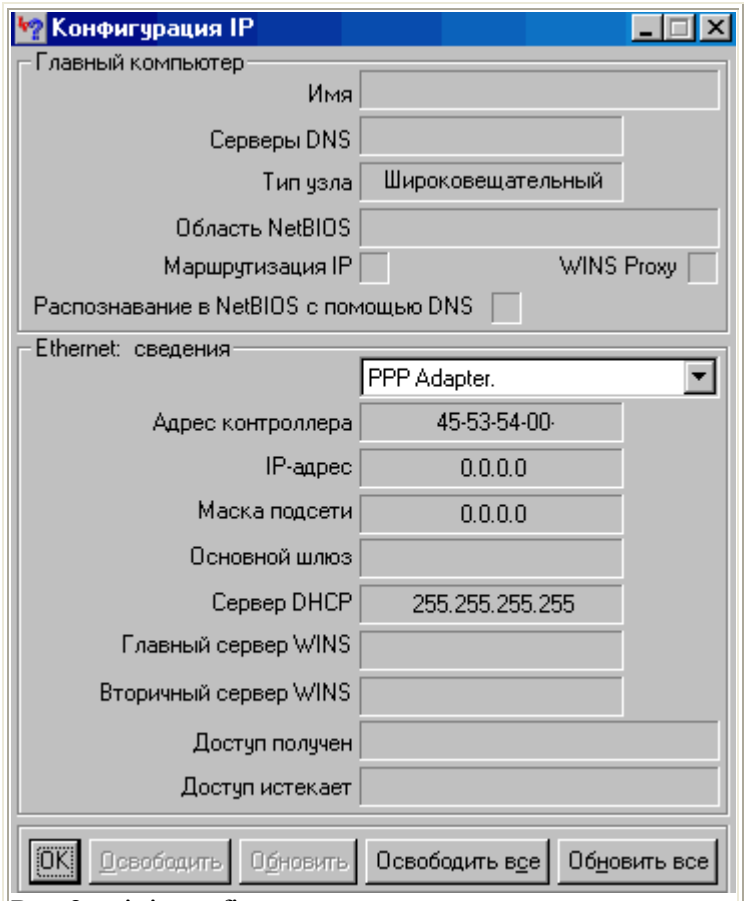

Рис 2. winipconfig

В выпадающем списке ты можешь увидеть PPP-адаптер (если ты подключён к Инету) и имя своей сетевой карты (если она есть). Выбирая одно из них, ты можешь увидеть их свойства.

Итак, прежде чем пакет будет отправлен, машина должна знать адрес получателя. Протокол ARP занимается поиском этого адреса. В общем случае поиск MAC адреса происходит так:

- Сначала происходит поиск в кэше. Если адрес не найден, то переходим дальше
- Посылается широковещательный ARP запрос. В этом запросе устанавливается MAC адрес FF-FF-FF-FF-FF и указывается IP адрес нужной машины. Если какая-нибудь машина в сети знает о существовании этого IP адреса и знает его MAC адрес, то она возвращает ответ с MAC адресом нужной машины. Полученный адрес помещается в кэш.
- Если и после этого не найден адрес, то пакет отправляется в шлюз.

Если IP адрес найден в локальной сети, то компьютер получает реальный MAC адрес. Если нет, то запрос отправляется маршрутизатору, который ищет MAC адрес в удалённой сети. Когда он найдёт MAC адрес, он возвращает компьютеру не его, а свой MAC адрес. Таким образом, компьютер будет посылать пакеты на IP а указывать MAC адрес маршрутизатора, а он будет переправлять пакет куда надо. Таким образом маршрутизатор становится "прокси-сервером".

Когда компьютер получает MAC-адрес, то он сохраняет его в кэше. Адреса в этом кэше валяются в течении определённого времени (по умолчанию 10 минут). Если компьютер в течении 10 минут ещё раз обращался по этому IP адресу, то может начаться повторный отсчёт с самого начала. Но такое бывает не на всех системах.

Для просмотра ARP кэша в Windows можно воспользоваться командой ARP с параметром -g или -a.

Существует обратная реализация ARP - RARP (Reverse Address Resolution Protocol). Он выполняет обратное действие для ARP, то есть определяет IP адрес по MAC адресу. Но этот протокол используется очень редко и я даже не могу придумать пример, для его использования. Но я всё же решил упоминуть RARP просто для общего развития.