**Лекция за 07.03.2012, часть 2**

***Базовые архитектуры или топологии вычислительных сетей.***

***Что это такое базовая архитектура?*** Любую сеть можно представить в виде некоторого графа. В зависимости от свойств этого графа у сетей можно найти какие-то общие свойства, и использовать их, допустим, при проектировании вычислительных сетей, для анализа производительности и других целей. Существует несколько базовых топологий. Но до сих пор не принято окончательное решение - относить к отдельной топологии топологию ***«облако»***, или нет.

Немного терминологии. Когда говорят о сетях, обычно используют термин ***хост***. В простейшем случае хост – это один компьютер. Но на самом деле надо внимательно следить, на каком уровне модели стека протоколов мы находимся. То есть если мы говорим, например, о сетевом протоколе и выше, то один хост - это практически гарантированно один компьютер. Если мы говорим о физическом уровне, то хостом может быть и один сетевой интерфейс. Чтобы не возникало путаницы, то когда хотят выделить несколько сетевых интерфейсов в одном компьютере, то все-таки используют термин интерфейс. Соединение двух хостов называют ***линком.***

*Рис 1. Топология «точка-точка»*

Самая простая топология – это топология ***«точка-точка»***. На физическом уровне она выглядит как два хоста, которые соединены одним каналом связи. У этого типа соединения есть одно важное преимущество, но, к сожалению, оно уже не действует. Это соединение требует минимального overhead-а для передачи служебной информации: так как всего два абонента в сети, то абсолютно точно известно – кто отправитель, а кто – получатель данных. На этом достоинства заканчиваются. Одним фатальным недостатком такого типа сети является ее немасштабируемость. На физическом уровне к сетям типа точка-точка относились модемные соединения. При этом модем выступал в роли сетевого интерфейса, соединялся он по телефонной линии и, в общем-то, гарантированно расширяемым не являлся. Но с другой стороны практически на каждом уровне модели ISO/OSI можно найти логические соединения точка-точка. На сеансовом уровне это может быть соединение клиента и сервера в рамках одного сеанса. На транспортном уровне, например TCP соединение хостов по TCP каналу. Еще раз стоит отметить, что когда вы говорите о той или иной топологии, всегда необходимо указывать, если не очевидно из контекста, к какому уровню относится эта топология. В соответствии с этим еще один хороший пример соединения точка-точка – это VPN – соединение (Virtual Private Network). Причем там именно его немасштабируемость является существенным плюсом, поскольку она обеспечивает безопасность соединения. При осуществлении финансовых транзакций многие организуют сети таким образом, чтобы соединение было типа точка-точка. Так что помимо overhead-a есть еще одно достоинство – теоретически самая высокая конфиденциальность. На этом достоинства этой архитектуры, пожалуй, заканчиваются.

Резюмируя достоинства и недостатки топологии «точка-точка»:

+ минимальный overhead для передачи служебной информации;

+ теоретически самая высокая конфиденциальность;

- немасштабируемость.

*Рис 2. Топология «звезда»*

Следующий тип топологии – топология ***«звезда»***. Если соединение точка-точка мы изображаем в виде отрезка, то звезда – это совокупность соединений точка-точка, у которых одна точка – общая. Здесь начинается жуткая проблема с терминологией, побороть которую, видимо, уже невозможно. Общий узел в звезде называется по-английски ***hub***. По-русски когда говорят о топологии сетей, то он называется ***коммутатор***. Но когда мы будем изучать Ethernet, то там он называется ***концентратор***. Эта терминологическая путаница устоялась, и сделать с ней, пожалуй, ничего невозможно. Ничего более умного, чем запомнить, что в разных контекстах оно переводится по-разному, видимо, нельзя. Причина возникновения путаницы очень простая – эта терминология пришла к нам еще из телефонии. В телефонии то, что в английском языке называется hub, в русском языке всегда называлось коммутатор. А в связи с тем, что понятие базовых топологий древнее, чем компьютерные сети, то понятие коммутатор перекочевало и сюда.

Сети с такой топологией нечувствительны к выходу из строя отдельных хостов и отдельных соединений: что бы ни произошло с конечными лучами звезды – остальные будут работать. К сожалению, это не относится непосредственно к коммутатору. То есть самая важная часть звезды – это устройство, которое осуществляет коммутацию. Естественно, что выход его из строя немедленно выводит из строя всю звезду. С этим связана дороговизна таких сетей - наличие в ней выделенного высоконадежного коммутатора естественно увеличивает стоимость сети, вообще говоря, в разы. Это условие привело к тому, что сети с топологией звезда внедрялись достаточно медленно. Другой важной положительной стороной такой сети является ее конфиденциальность, то есть если коммутатор не скомпрометирован, то компрометация отдельных узлов сети не влияет на остальные узлы сети – абсолютно аналогично выходу из строя. Связано это с тем, что, по крайней мере, теоретически у нас нет никаких корреляций между отдельными узлами в этой топологии. Можно отметить, что вся проводная телефонная сеть построена по принципу топологии звезда. В качестве коммутатора выступают телефонные станции, а в качестве хостов – телефоны. Даже сотовая связь, при всех ее отличиях от проводной, тоже во многом повторяет звезду: в качестве коммутаторов тут выступают базовые станции, а в качестве хостов – телефонные аппараты. «По науке» телефон в сотовой связи называется «пользовательский терминал», что вполне логично: например, GSM модем - это тоже пользовательский терминал, но со всей очевидностью, телефоном он не является. На других уровнях модели ISO/OSI можно обнаружить соответствующие звезды. Например, любой web-сервер действует как коммутатор звезды, так как он может одновременно предоставлять данные нескольким абонентам.

Резюмируя достоинства и недостатки топологии «звезда»:

+ нечувствительность сети к выходу отдельных хостов из строя (высокая надежность)

+ нечувствительность сети к компрометации отдельных хостов (высокая конфиденциальность)

- высокие требования к надежности коммутационного оборудования => его высокая стоимость

- ограниченная масштабируемость - количество одновременных физических соединений ограниченно количеством отверстий в коммутаторе.

*- маркер*

Рис 3. Топология «кольцо»

Еще одна базовая архитектура – это ***кольцо*** или ***маркерное кольцо***. В чем смысл маркерного кольца? Это сеть, в которой каждый узел связан ровно с двумя соседями. Причем информация передается по кольцу на манер эстафеты. Направление обхода задано. То есть для того, чтобы попасть из узла в предыдущий против направления обхода, надо пройти через все кольцо. Отсюда сразу следует неутешительный вывод: у кольцевой сети есть проблемы с конфиденциальностью и с надежностью. То есть существенным недостатком кольца является то, что выход из строя любого узла или любого соединения мгновенно приводит к неработоспособности всего кольца. С другой стороны, компрометация любого узла в кольце компрометирует все кольцо. Поскольку злоумышленник, который может перехватывать данные, которые проходят через один узел, может перехватывать все данные. Каким образом работает кольцо? В кольце есть специальный объект, который носит название ***маркера***. Маркер - это либо электрический сигнал специальной формы, который может быть гарантированно выделен на фоне сигналов, использующих сигнал передачи данных, либо – последовательность данных, которая гарантированно не встречается при заданном способе представления данных в кольце. Маркер может иметь два состояния: так называемые ***загруженное*** и ***незагруженное***. Если маркер загружен, то следом за ним движется поток данных. Если маркер незагружен, то следом за ним любой хост может вставить свои данные и перевести маркер из незагруженного состояния в загруженное, например, сменив его полярность. Когда маркер доходит до точки назначения, то хост-получатель изымает данные из сети и меняет состояние маркера с загруженного на незагруженное. После этого возможна дальнейшая передача данных. По этому описанию видно, что на самом деле любой хост должен заниматься ретрансляцией данных. Потому что даже если он получатель, то он сначала получает загруженный маркер, а уже потом понимает, что данные предназначены именно ему. Это также накладывает еще одно ограничение непосредственно на интерфейсы такого кольца. Эти интерфейсы должны быть всегда включены. Например, интерфейсы Token Ring-a и FDDI всегда имели независимое питание.

Было две реализации, доведенные до промышленного использования: это сеть ***Token Ring***, она же стандарт 802.5 и сеть ***FDDI*** ***(fiber-optics data distribution interface)***. Все что написано про кольцо выше, было написано про сеть Token Ring.

FDDI отличалась тем, что она была не одинарным кольцом, а двойным. Причем маркеры двигались по ним в разные стороны. Это можно было использовать либо для увеличения пропускной способности, либо для увеличения надежности передачи данных. Скорость передачи данных в одном кольце FDDI была 144 Мбит, в двух соответственно – 288. И это на конец 80-хх – начало 90-хх гг.! По тем временам это было замечательной скоростью. Поэтому FDDI называли «сетью-мечтой». Носителем в ней служил оптический кабель. Ethernet в то время преимущественно был представлен своей 10 Мбит-ной разновидностью. Сравните: 10 и 144. К сожалению, стоимость FDDI составляла еще больше, чем ее скорость. То есть позволить себе FDDI сети могли Пентагон и другие гос. организации США, крупнейшие корпорации и кампусы крупнейших университетов, для которых оборудование продавалось с очень большим дисконтом. К сожалению FDDI был совместим лишь с самим собой – он использовал свой собственный формат представления данных. Единственный способ получить что-либо наружу из кольца FDDI – было использование какого-нибудь IP маршрутизатора. То есть, например хост, в котором есть FDDI адаптер и еще плата Ethernet. Причем поначалу FDDI контроллер из себя представлял довольно крупных размеров коробку, которая подключалась к плате, которая вставлялась непосредственно в компьютер, либо в высокоскоростной последовательный порт – оборудование было еще и крайне громоздко. То есть изначально эта коробка была с небольшой тортик, а под конец жизни FDDI размер этой коробки уменьшился до пары сигаретных пачек. То есть FDDI погубили высокая стоимость и несовместимость ни с чем.

С Token Ring ситуация была другая. Первоначально в качестве носителя Token Ring использовал коаксиальный кабель, а в дальнейшем стал использовать витую пару. При этом скорость его достигала около 30 Мбит/с, и обсуждалось дальнейшее развитие стандарта сначала 50 Мбит/с, а потом и 100 Мбит/с. Причем, забегая вперед, хочется отметить, что пропускная способность на 30 Мбит-ном Token Ring-e была примерно как на 100 Мбит-ном Ethernet-e. Token Ring сгубила маркетинговая политика фирмы IBM. Вообще, если говорить об эпических fail-ax в сфере компьютерной индустрии, то пальму первенства соберет IBM. Удивительно, что после всех этих fail-ов она еще остается на плаву и остается одним из лидеров бизнеса. Так что же сгубило Token Ring? Начиналась конкуренция между Ethernet и Token Ring. Основным достоинством Ethernet было то, что он был дешев и нелицензируем. IBM сначала пыталась монополизировать рынок адаптеров для Token Ring, а потом, когда стало понятно, что она проигрывает схватку, она стала требовать лицензионных отчислений от производителей оборудования. Token Ring. В те времена адаптер Ethenet стоил порядка 100$, адаптер Token Ring стоил порядка 500$, а адаптер FDDI стоил порядка 5000$. Соответственно через 10 лет стоимость адаптера Eternet упала до 10$, а Token Ring и FDDI просто ушли со сцены. IBM потребовала отчислений порядка 25 долларов за устройство. При этом при цене 500$ это было хоть как-то разумно – если делать их в Китае, то они будут стоить не 500, а 300. Но если речь идет о суммах меньше 100 долларов, то 25 долларов отчислений IBM играют существенную роль. Вот такой вот fail.

Достоинства кольцевой топологии: топология кольца приближается к максимально возможной теоретической пропускной способности. То есть ограничение на пропускную способность связано только с тем, что если маркер уже загружен, то пока он не будет освобожден, дальнейшая передача невозможна. Наихудший вариант, конечно, когда посылается широковещательное сообщение по всей сети – в этом случае изымать маркер или изменять его состояние будет сам отправитель. В Token Ring MAC-адрес такой же, как и в Eternet – 6 байтовая последовательность. Если в Ethernet широковещательный запрос – это запрос по адресу из одних нулей, то в Token Ring широковещательный запрос помимо запроса по адресу из одних нулей мог быть еще и запросом по адресу самого отправителя, то есть адрес отправителя и получателя в данном запросе совпадали.

Какие есть расширения идеи маркерного кольца? Например, ***многомаркерное кольцо***. Самый предельный вариант – это когда количество маркеров совпадает с количеством абонентов. Но в реальных сетях так никто не делает, потому что это требует очень высокого качества кабельного монтажа. Забегая вперед к теме «Физическое оборудование сетей Ethernet», можно сказать, что некоторые коммутаторы сетей Ethernet используют внутри архитектуру с кольцевой шиной. Как раз там и используется идея многомаркерных колец.

Итого, достоинства и недостатки топологии «кольцо»:

+ близкая к теоретически возможной пропускная способность;

+ относительно легкая масштабируемость – вставить хост в кольцо не составляет труда;

- чувствительность по надежности и конфиденциальности к каждому своему хосту.