Конспект по теме: «Основные сетевые термины и сетевые модели»

Сеть - это совокупность устройств и систем, которые подключены друг к другу (логически или физически) и общающихся между собой (это сервера, компьютеры, телефоны, маршрутизаторы и так далее).

Размер сети может достигать размера Интернета, а может состоять всего из двух устройств, соединенных между собой кабелем.

Компоненты сети по группам:

1) Оконечные узлы: устройства, которые передают и/или принимают какие-либо данные. Это могут быть компьютеры, телефоны, сервера, какие-то терминалы или тонкие клиенты, телевизоры.

2) Промежуточные устройства: это устройства, которые соединяют оконечные узлы между собой. Сюда можно отнести коммутаторы, концентраторы, модемы, маршрутизаторы, точки доступа Wi-Fi.

3) Сетевые среды: это те среды, в которых происходит непосредственная передача данных. Сюда относятся кабели, сетевые карточки, различного рода коннекторы, воздушная среда передачи. Если это медный кабель, то передача данных осуществляется при помощи электрических сигналов. У оптоволоконных кабелей, при помощи световых импульсов. Ну и у беспроводных устройств, при помощи радиоволн.

Использование сетей:

1) Приложения: При помощи приложений отправляем разные данные между устройствами, открываем доступ к общим ресурсам. Это могут быть как консольные приложения, так и приложения с графическим интерфейсом.

2) Сетевые ресурсы: Это сетевые принтеры, которыми, к примеру, пользуются в офисе или сетевые камеры, которые просматривает охрана, находясь в удаленной местности.

3) Хранилище: Используя сервер или рабочую станцию, подключенную к сети, создается хранилище доступное для других. Многие люди выкладывают туда свои файлы, видео, картинки и открывают общий доступ к ним для других пользователей. Пример, который на ходу приходит в голову, — это google диск, яндекс диск и тому подобные сервисы.

4) Резервное копирование: Часто, в крупных компаниях, используют центральный сервер, куда все компьютеры копируют важные файлы для резервной копии. Это нужно для последующего восстановления данных, если оригинал удалился или повредился. Методов копирования огромное количество: с предварительным сжатием, кодированием и так далее.

5) VoIP: Телефония, работающая по протоколу IP. Применяется она сейчас повсеместно, так как проще, дешевле традиционной телефонии и с каждым годом вытесняет ее.

Виды топологии:

1) Топология с общей шиной (англ. Bus Topology)

Одна из первых физических топологий. Суть состояла в том, что к одному длинному кабелю подсоединяли все устройства и организовывали локальную сеть. На концах кабеля требовались терминаторы. Как правило — это было сопротивление на 50 Ом, которое использовалось для того, чтобы сигнал не отражался в кабеле. Преимущество ее было только в простоте установки. С точки зрения работоспособности была крайне не устойчивой. Если где-то в кабеле происходил разрыв, то вся сеть оставалась парализованной, до замены кабеля.

2) Кольцевая топология (англ. Ring Topology)

В данной топологии каждое устройство подключается к 2-ум соседним. Создавая, таким образом, кольцо. Здесь логика такова, что с одного конца компьютер только принимает, а с другого только отправляет. То есть, получается передача по кольцу и следующий компьютер играет роль ретранслятора сигнала. За счет этого нужда в терминаторах отпала. Соответственно, если где-то кабель повреждался, кольцо размыкалось и сеть становилась не работоспособной. Для повышения отказоустойчивости, применяют двойное кольцо, то есть в каждое устройство приходит два кабеля, а не один. Соответственно, при отказе одного кабеля, остается работать резервный.

3) Топология звезда (англ. Star Topology)

Все устройства подключаются к центральному узлу, который уже является ретранслятором. В наше время данная модель используется в локальных сетях, когда к одному коммутатору подключаются несколько устройств, и он является посредником в передаче. Здесь отказоустойчивость значительно выше, чем в предыдущих двух. При обрыве, какого либо кабеля, выпадает из сети только одно устройство. Все остальные продолжают спокойно работать. Однако если откажет центральное звено, сеть станет неработоспособной.

4)Полносвязная топология (англ. Full-Mesh Topology)

Все устройства связаны напрямую друг с другом. То есть с каждого на каждый. Данная модель является, пожалуй, самой отказоустойчивой, так как не зависит от других. Но строить сети на такой модели сложно и дорого. Так как в сети, в которой минимум 1000 компьютеров, придется подключать 1000 кабелей на каждый компьютер.

5)Неполносвязная топология (англ. Partial-Mesh Topology)

Как правило, вариантов ее несколько. Она похожа по строению на полносвязную топологию. Однако соединение построено не с каждого на каждый, а через дополнительные узлы. То есть узел A, связан напрямую только с узлом B, а узел B связан и с узлом A, и с узлом C. Так вот, чтобы узлу A отправить сообщение узлу C, ему надо отправить сначала узлу B, а узел B в свою очередь отправит это сообщение узлу C. В принципе по этой топологии работают маршрутизаторы. Приведу пример из домашней сети. Когда вы из дома выходите в Интернет, у вас нет прямого кабеля до всех узлов, и вы отправляете данные своему провайдеру, а он уже знает куда эти данные нужно отправить.

6) Смешанная топология (англ. Hybrid Topology)

Самая популярная топология, которая объединила все топологии выше в себя. Представляет собой древовидную структуру, которая объединяет все топологии. Одна из самых отказоустойчивых топологий, так как если у двух площадок произойдет обрыв, то парализована будет связь только между ними, а все остальные объединенные площадки будут работать безотказно. На сегодняшний день, данная топология используется во всех средних и крупных компаниях.

Сетевая модель OSI из 7 уровней и каждый уровень выполняет определенную ему роль и задачи.

1) Физический уровень (Physical Layer): определяет метод передачи данных, какая среда используется (передача электрических сигналов, световых импульсов или радиоэфир), уровень напряжения, метод кодирования двоичных сигналов.

2) Канальный уровень (Data Link Layer): он берет на себя задачу адресации в пределах локальной сети, обнаруживает ошибки, проверяет целостность данных. Если слышали про MAC-адреса и протокол «Ethernet», то они располагаются на этом уровне.

3) Сетевой уровень (Network Layer): этот уровень берет на себя объединения участков сети и выбор оптимального пути (т.е. маршрутизация). Каждое сетевое устройство должно иметь уникальный сетевой адрес в сети. Думаю, многие слышали про протоколы IPv4 и IPv6. Эти протоколы работают на данном уровне.

4) Транспортный уровень (Transport Layer): Этот уровень берет на себя функцию транспорта. К примеру, когда вы скачиваете файл с Интернета, файл в виде сегментов отправляется на Ваш компьютер. Также здесь вводятся понятия портов, которые нужны для указания назначения к конкретной службе. На этом уровне работают протоколы TCP (с установлением соединения) и UDP (без установления соединения).

5) Сеансовый уровень (Session Layer): Роль этого уровня в установлении, управлении и разрыве соединения между двумя хостами. К примеру, когда открываете страницу на веб-сервере, то Вы не единственный посетитель на нем. И вот для того, чтобы поддерживать сеансы со всеми пользователями, нужен сеансовый уровень.

6) Уровень представления (Presentation Layer): Он структурирует информацию в читабельный вид для прикладного уровня. Например, многие компьютеры используют таблицу кодировки ASCII для вывода текстовой информации или формат jpeg для вывода графического изображения.

7) Прикладной уровень (Application Layer): это самый понятный для всех уровень. Как раз на этом уроне работают привычные для нас приложения — e-mail, браузеры по протоколу HTTP, FTP и остальное.

Процесс инкапсуляции и деинкапусуляции:

1) Представим ситуацию, что вы сидите у себя дома за компьютером, а в соседней комнате у вас свой локальный веб-сервер. И вот вам понадобилось скачать файл с него. Вы набираете адрес страницы вашего сайта. Сейчас вы используете протокол HTTP, которые работает на прикладном уровне. Данные упаковываются и спускаются на уровень ниже.

2) Полученные данные прибегают на уровень представления. Здесь эти данные структурируются и приводятся в формат, который сможет быть прочитан на сервере. Запаковывается и спускается ниже.

3) На этом уровне создается сессия между компьютером и сервером.

4) Так как это веб сервер и требуется надежное установление соединения и контроль за принятыми данными, используется протокол TCP. Здесь мы указываем порт, на который будем стучаться и порт источника, чтобы сервер знал, куда отправлять ответ. Это нужно для того, чтобы сервер понял, что мы хотим попасть на веб-сервер (стандартно — это 80 порт), а не на почтовый сервер. Упаковываем и спускаем дальше.

5) Здесь мы должны указать, на какой адрес отправлять пакет. Соответственно, указываем адрес назначения (пусть адрес сервера будет 192.168.1.2) и адрес источника (адрес компьютера 192.168.1.1). Заворачиваем и спускаем дальше.

6) IP пакет спускается вниз и тут вступает в работу канальный уровень. Он добавляет физические адреса источника и назначения, о которых подробно будет расписано в последующей статье. Так как у нас компьютер и сервер в локальной среде, то адресом источника будет являться MAC-адрес компьютера, а адресом назначения MAC-адрес сервера (если бы компьютер и сервер находились в разных сетях, то адресация работала по-другому). Если на верхних уровнях каждый раз добавлялся заголовок, то здесь еще добавляется концевик, который указывает на конец кадра и готовность всех собранных данных к отправке.

7) И уже физический уровень конвертирует полученное в биты и при помощи электрических сигналов (если это витая пара), отправляет на сервер.

Процесс деинкапсуляции аналогичен, но с обратной последовательностью:

1) На физическом уровне принимаются электрические сигналы и конвертируются в понятную битовую последовательность для канального уровня.

2) На канальном уровне проверяется MAC-адрес назначения (ему ли это адресовано). Если да, то проверяется кадр на целостность и отсутствие ошибок, если все прекрасно и данные целы, он передает их вышестоящему уровню.

3) На сетевом уровне проверяется IP адрес назначения. И если он верен, данные поднимаются выше. Не стоит сейчас вдаваться в подробности, почему у нас адресация на канальном и сетевом уровне. Это тема требует особого внимания, и я подробно объясню их различие позже. Главное сейчас понять, как данные упаковываются и распаковываются.

4) На транспортном уровне проверяется порт назначения (не адрес). И по номеру порта, выясняется какому приложению или сервису адресованы данные. У нас это веб-сервер и номер порта — 80.

5) На этом уровне происходит установление сеанса между компьютером и сервером.

6) Уровень представления видит, как все должно быть структурировано и приводит информацию в читабельный вид.

7) И на этом уровне приложения или сервисы понимают, что надо выполнить.

Token Ring и FDDI — не сетевые модели! Token Ring — это протокол канального уровня, а FDDI это стандарт передачи данных, который как раз основывается на протоколе Token Ring.

Таблица соответствия моделей OSI и TCP/IP.

|  |  |
| --- | --- |
| Модель OSI | МодельTCP/IP |
| Прикладной уровень | Прикладной уровень |
| Уровень представления |  |
| Сеансовый уровень |  |
| Транспортный уровень | Транспортный уровень |
| Сетевой уровень | Уровень межсетевого взаимодействия |
| Канальный уровень | Уровень сетевых интерфейсов |
| Физический уровень |  |