

Лекция 8. Сетевое ПО

- Операционные возможности
- Производительность. Средне суммарное быстродействие
- Время передачи сообщения. Пропускная способность, расстояние
- Стоимость обработки данных. Время и объем хранения данных

Все сети можно классифицировать по признакам:

- Корпоративность
 - Локальная
 - Глобальная
- Организация сетевого ПО
 - Одно ранговые. Все равны.
 - Сети централизованного управления.
- Метод доступа
 - Ethernet
 - Token ring
 - Arcnet
- Протоколы передачи данных
 - РТР. Peer to Peer
 - Множественные.
- Принципы коммутации
 - Каналы
 - Пакеты

Коммутация каналов предполагает, что сеть образует между узлами непрерывный физический канал, который построен из последовательно соединенных узлов, следовательно, перед передачей информации по сети передается управляющий пакет, который обеспечивает настройку физического соединения между транзитными узлами.

Коммутация пакетов. Каждый пакет имеет заголовок и конец, которые определяют сообщение.

- Сети с динамической коммутацией. Позволяют устанавливать соединения по инициативе пользователя сети. Коммутация выполняется только во время сеанса связи, а затем по инициативе одного из пользователей разрывается. В любой момент времени пользователь может соединиться с любым пользователем сети. Время соединения между пары пользователей составляет от нескольких секунд, до часов и завершаются при окончании передачи связи. Примеры: локальные сети, TCP/IP
- Сети с постоянной коммутацией. Позволяют паре пользователей заказать постоянное длительное соединение на требуемый период времени, при этом соединение устанавливается не пользователем, а оборудованием. Обычно выделяется длительная аренда канала, на котором выделяется обслуживание. Сети с постоянной коммутацией называют сетями с выделенным каналом. Наиболее популярными сетями являются SDH. В настоящее время сети SDH реализуются на технологиях, которые позволяют к дополнительной коммутации использовать динамическую коммутацию с целью разделения трафика.

Существуют 2 основных класса компьютерных сетей с коммутацией пакетов: дейтаграммы сети, сети с виртуальными каналами.

В диаграммных сетях каждый пакет обязан содержать информацию о получателе, причем этот адрес имеет иерархическую структуру. Каждый получатель анализирует фрагмент адреса и на основании собственной таблицы маршрутизации формирует адрес следующего по маршруту узла. Таким образом в случае дейтаграммы сетей пакеты имеющие одинакового получателя, могут быть направлены по разным маршрутам. Обычно дейтаграммы сети определяются как сети без гарантии доставки и чаще всего используются в случае, если потеря не существенная.

Сети с виртуальным каналом характеризуются:

- Маршрут.
- Номером виртуального канала.
- Записями в таблице маршрутизации.

Стратегии повторной передачи

Stop and wait. Каждый следующий пакет отправляется только тогда, когда получен предыдущий пакет. В целях реализации введено понятие квитанций. Оно определяет, что доставка пакета прошла успешно.

Go Back N. Предыдущий узел может отправить пакеты без ожидания. Каждый пакет имеет номер в пределах окна. Принимающий узел отправляет положительную квитанцию и номер следующего пакета. Нет затрат на хранение пакетов, которые были получены, но не требуемые.

Выборочная стратегия. Плавающее окно, но на принимающем узле есть маска, в которой отмечается номера пакетов, которые приняты без подтверждения. В этом случае при повторной передачи передаются только те пакеты, которые не приняты на принимающей стороне.

Распределенная обработка информации предполагает организацию доступа к одним и тем же информационным блокам различным приложениям. В данном случае под информационным блоком можно понимать ПО так и непосредственные блоки данных. Распределенная обработка разрешает недостатки чисто архитектуры клиент-сервер, так как включает в состав программно аппаратного средства дополнительный промежуточный слой, который содержит набор сервисов, отвечающий за разделение информации. Введение промежуточного слоя обусловлено недостатками:

- Ограниченные возможности масштабирования архитектуры
- Необходимость изменения клиентских приложений при изменении работы серверной логики

Middleware – класс ПО предназначенный для объединения компонентов распределенного клиент-сервисного приложения или различных приложений в единую информационную систему. Данное ПО представляет собой набор сервисов обращение к которым позволяет различным приложениям

расположенных на разных станциях взаимодействовать между собой. При этом разработчик не обязан вникать в тонкости удаленной реализации:

- 1) Прозрачный доступ к другим сервисам и приложениям
- 2) Независимость от сервисов
- 3) Высокая надежность и постоянная готовность

Используемый в настоящее время промежуточное ПО можно разбить на 2 класса:

- 1) ПО межпрограммного взаимодействия
- 2) ПО доступа к БД

Протоколы и продукты по межпрограммному взаимодействию обычно используются в корпоративных ИС и могут быть реализованные в 3-х формах:

- 1) Удаленный вызов процедур. Разработана и реализована ксероксом (компания). Программа выполняемая на машине клиента использует процедуры, находящиеся на машине сервиса абсолютном прозрачным способом. Работа выполнена в соответствии с работой прерывания. Стаб клиент моделирует работу связанную приостановкой выч процесса и получение результата вызываемой процедуры. На сервере стаб сервер – спец ПО, которое обеспечивает прием вызываемых параметров, вызов требуемой процедуры, получение результатов и обратная связь с клиентом.
- 2) Сервисы обмена сообщений. Ассинхронная система, которая обеспечивает взаимодействие между клиентом и сервером на базе обмена сообщений. Под сообщение понимается текстовый блок, который может быть как программа так и данными. Для передачи сообщения используют байт ориентированные протоколы (Http, smtp). Обмен сообщений
 - a. Непосредственны. Сообщения появляющиеся у приложения передаются менеджеру, который осуществляет связь с протоколами передачи данных и передает сообщения на другую машину.
 - b. Очереди. Для каждого приложения создается своя очередь, которая буферизируется как принимаемые, так и передаваемые. Обслуживаются менеджером на машине, который обеспечивает сетевое взаимодействие. Наличие очередей обеспечивает доставку сообщения, чаще всего по расписанию. При получении сообщения сервер должен указать успешность передачи данных через механизм повторной передачи и контрольной суммы.
 - c. Публикация\подписка. Обслуживание клиента по подписке работает по принципу, сходим на почту. Одно приложение публикует информацию в сети, а приложения подписываются на нее. Как только информация появляется в сети, то приложение ее может захватить. Для этого на машинах запускается ПО, которое принято называть демоном контактов. Основное достоинство: динамическая реконфигурация и полная независимость приложений.
- 3) Распределенные объектные системы

Мониторы обработки транзакции позволяют контролировать передачу данных от клиента при работе с распределенными БД. Он обеспечивает целостность данных и регулирует завершенность транзакции.

Монитор транзакции отслеживает:

- 1) Атомарность. Обеспечивает, что транзакция является неделимым блоком при невозможности транзакции, при возможности отката
- 2) Согласованность. По завершению транзакции все ресурсы должны находиться в определенном состоянии.

- 3) Изолированность. Одновременный доступ к транзакции различных приложений к разделяемым ресурсам координируется так, чтобы исключить влияние приложений друг на друга.
- 4) Долговременность.

Монитор транзакции отслеживает выполнение критерий и обеспечивает распределенную обработку баз данных.

Одним из механизмов является механизм программных гнезд. Это точка соединения, предназначенная для передачи блоков информации. Программные гнезда – сокет. Могут быть клиентскими и серверными. Клиентское обеспечивает точки доступа к требуемой информации. Непосредственную передачу и прием информации. Серверное используется для взаимодействия с одним или нескольких клиентов.

Модели сетевой архитектуры

Физический уровень определяет тип соединения среды, физические топологии и способы передачи данных. Распределение каналов связи с использованием мультиплексирования. Физический уровень не включает описание среды передачи, однако способ передачи различный. Не регламентируется программно, а аппаратно

Канальный уровень определяет логическую топологию сети, а так же правила получения в среде передачи данных. Решаются вопросы адресации физических устройств, а также вопросов сервисы соединения:

- Организация фреймов по передаче на физический уровень
- Обнаружение ошибок при передаче
- Управление потоками данных
- Идентификация компьютера в сети по их физическим адресам.
- Вопросы повторной передачи.

Сетевой. Служит для образования единой транспортной системы объединяющий несколько сетей, причем эти сети могут использовать различные технологии передачи данных. На сетевом уровне решаются вопросы согласования различных топологий, вопросы адресации и маршрутизации. Сообщение, формируемое на сетевом уровне – пакеты. Каждый пакет определяет адрес и отправителя, и получателя с целью обеспечения маршрутизации. Основными протоколами сетевого уровня являются протоколы формирования маршрутных таблиц, протоколы маршрутизации и разрешения сетевых адресов.

Транспортный уровень. Обеспечивает приложению возможность доступа к протоколам более низкого уровня. Отвечает за качество предоставляемых сетевых услуг, при этом критерием качества является сложность передачи, возможность восстановления, мультиплексирование соединений и способность обнаружения и исправления и передачи данных.

Сеансовый уровень обеспечивает управление диалогом, а именно фиксирование какая из взаимодействующих сторон является активной, а какая пассивной. Средства синхронизации. Диалог реализуется одним из 3-х способов:

- Симплекс. Передача в одну сторону.
- Дуплекс. Обе стороны.
- Полудуплекс. Не одновременно, а по очереди в обе стороны.

Представительский. Осуществляет разбор передачи от приложения информации и преобразовании ее к внутреннему формату. Разрешает синтаксические различия в кодировке символов. Например, SSL.

Прикладной уровень. Включает набор протоколов, с помощью которых пользователь получает доступ. Http, FTP, SMTP, Telnet и т.д.

TCP/IP

Физический (канальный и физический).

Сетевой уровень. IP/ICMP/RIP/OSPF/ARP

Транспортный и сеансовый. TCP, UDP

Прикладной.

- 1) 0.0.0.0 – служебный адрес, который используется хостом при загрузке
- 2) Нулевой номер сети – текущая сеть
- 3) Все единицы – широковещательный

DNS

Предназначена для представления символьного адреса в логический. Есть дерево, в корне которого находится самая последняя точка. Деление доменного пространства между ДНС серверами осуществляется по средством механизма зон. Зона – БД, в которой содержатся записи о соответствии множества доменных имен, IP адреса. Каждая зона представляет собой фрагмент доменного пространства. Зона рассматривается как основной административный элемент. На уровне зоны происходит управление пространством имен. Границы зоны не определяются доменной структурой, по сути одна зона может включать в себя несколько доменов. Зона может быть размещена на одном вычислительном узле или на нескольких. В случае разделения зоны по серверам обычно выделяется основной сервер и только он имеет право на внесение изменений в базу. Остальные реплицированные сервера только хранят копию и доступны для чтения.

DNS – приложение предназначено для ответа на DNS запрос. Выделяют несколько типов ДНС:

- Авторитативный ДНС. Он отвечает за доменную зону
- Мастер сервер\первичный сервер. Сервер имеющий право внесения изменений в данные зоны. Авторитативный
- Слейв\Вторичный сервер. Хранит информацию о зоне, не имеющий право на внесение изменений. Авторитативный
- Кэширующий. Сервер, не являющийся авторитативный. Сокращает время получения адресов
- Локальный DNS сервер. Обычно для локальной сети
- Перенаправляющий ДНС. Перенаправляет полученные запросы выше стоящему кеширующему серверу в виде рекурсивных запросов. Обычно используется для снижения нагрузки на авторитативные и другие кэширующие сервера.
- Корневой DNS. Авторитативный, который отвечает за корневые домены

- Регистрирующий DNS.

- 1) Рекурсивный метод.
- 2) Итеративный.