

МЕТОДЫ КОММУТАЦИИ В ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

Система коммутации решает задачу
внутрисетевого распределения информационных
ПОТОКОВ.

Существуют два основных принципа коммутации:

- непосредственное соединение — без запоминания передаваемой информации;
- соединение с накоплением информации.

При непосредственном соединении (коммутации каналов) осуществляется физическое соединение входящих линий с исходящими линиями по соответствующему адресу, поэтому здесь возможна организация диалога между абонентами (время задержки в передаче сообщения невелико) и абоненты имеют возможность вести передачу сообщений после установления соединения независимо от нагрузки, поступающей от других абонентов.

Однако в случае отсутствия свободных каналов пользователь получает отказ в установлении связи, поэтому систему с коммутацией каналов называют **системами с потерями вызовов.**

Соединение (коммутация) с накоплением информации бывает двух видов: *коммутация сообщений* и *коммутация пакетов*.

В системах с коммутацией сообщений пользователь не получает отказа в случае отсутствия свободных каналов.

Его сообщение временно записывается в память коммутационной станции и выдается после освобождения канала, поэтому **системы с накоплением называются системами с ожиданием**

Данный метод нашел применение на телеграфной сети общего пользования.

В системах с коммутацией пакетов исходящее сообщение делится на «пакеты», каждый из которых содержит часть полезной информации и заголовок.

Заголовок пакета, в свою очередь, содержит: идентификатор, определяющий принадлежность пакета к сообщению, адреса абонентов, порядковый номер пакета и др.

Реализация всех методов коммутации
(пакетов, каналов и (или)сообщений) в
телекоммуникационных сетях «лежит на
плечах» устройств, называемых
маршрутизаторами

Технологии масштабируемости.

Существуют три основные технологии масштабирования:

1. нивелирование времени ожидания связи;
2. распределение;
3. репликация.

Нивелирование времени ожидания связи применяется в случае географического масштабирования в целях сведения к минимуму времени ожидания ответа на запрос от удаленного сервера.

Например, если была запрошена служба удаленной машины, альтернативой ожиданию ответа от сервера будет осуществление на запрашивающей стороне других возможных действий.

Это означает разработку запрашивающего приложения в расчете на использование исключительно асинхронной связи.

Когда будет получен ответ, приложение прервет свою работу и вызовет специальный обработчик для завершения запроса, отправленный заранее.

Асинхронная связь часто используется в системах пакетной обработки и параллельных приложениях, в которых во время ожидания одной задачей завершения связи предполагается выполнение других более или менее независимых задач.

Для осуществления запроса может быть запущен новый управляющий поток выполнения.

Хотя он будет блокирован на время ожидания ответа, другие потоки процесса продолжат свое выполнение.

Многие приложения не в состоянии эффективно использовать асинхронную связь.

Например, когда в интерактивном приложении пользователь посылает запрос, он обычно просто ждет ответа от сервера.

В этих случаях наилучшим решением будет сократить необходимый объем взаимодействия, например переместив часть вычислений, обычно выполняемых на сервере, на клиента, процесс которого запрашивает службу.

Стандартный случай применения этого подхода
— доступ к базам данных с использованием
форм.

Обычно заполнение формы сопровождается
ссылкой отдельного сообщения на каждое поле и
ожиданием подтверждения приема от сервера,
как показано на рисунке

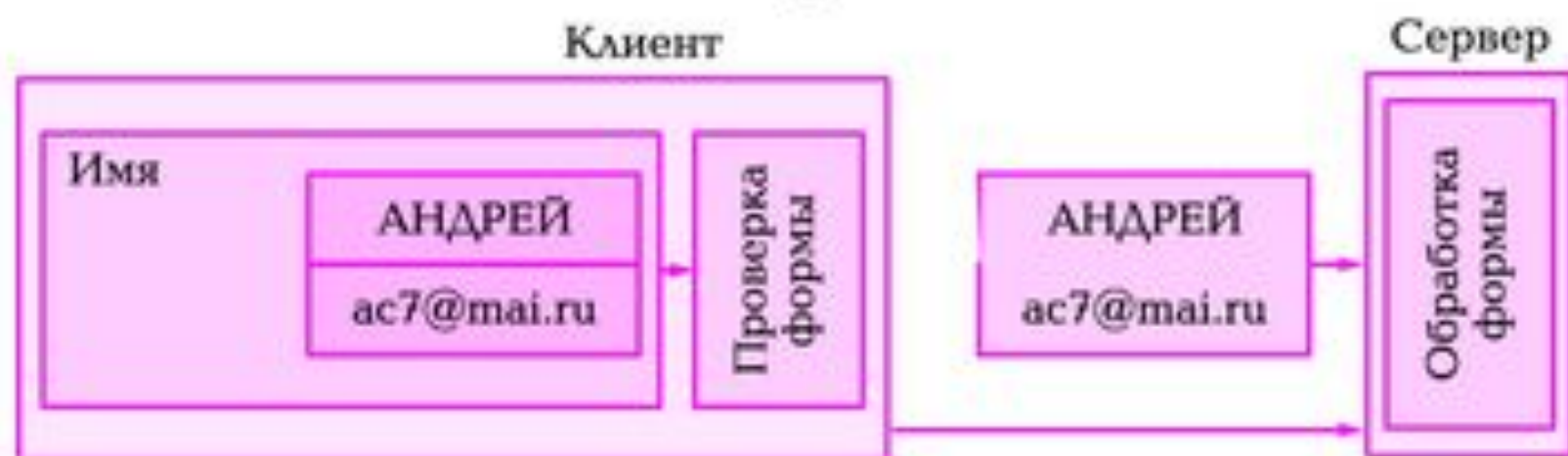
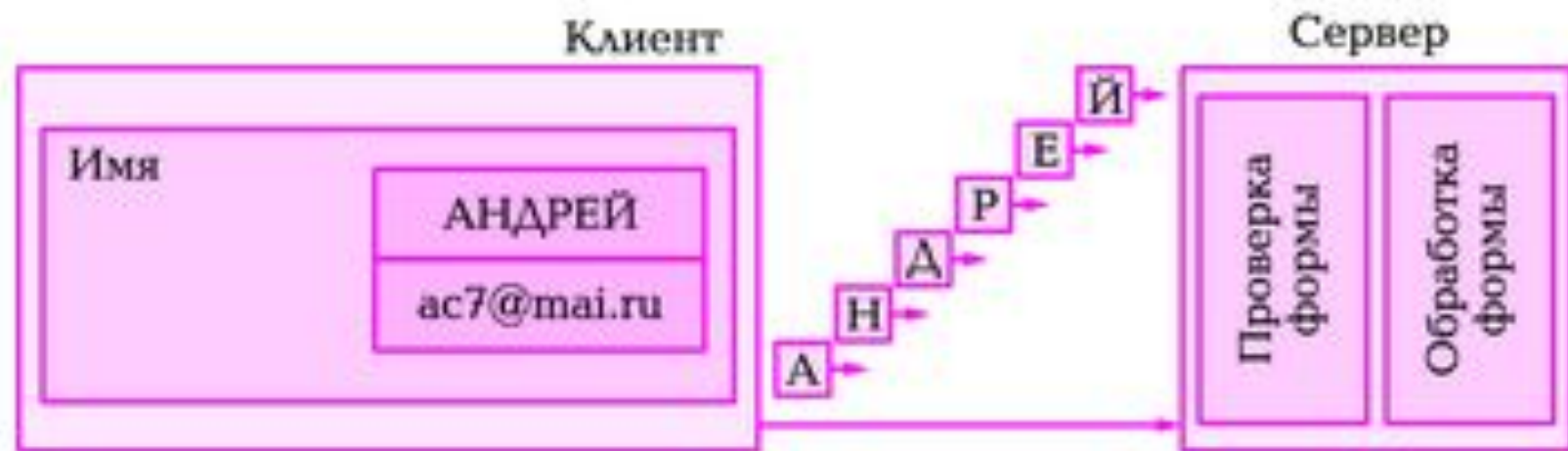


Рис. 1.9. Различие в обработке формы на сервере (а) и на клиенте (б)

Сервер может перед приемом введенного значения проверить его на синтаксические ошибки.

Более успешное решение, заключается в том, чтобы перенести код для заполнения формы и, возможно, проверки введенных данных на клиента, чтобы он мог послать серверу целиком заполненную форму.

Распределение— технология масштабирования, предполагающая разбиение компонентов на мелкие части и последующее разнесение этих частей по системе.

Примером распределения является система доменных имен Интернета (DNS), организованная иерархически, в виде дерева доменов, которые разбиты на неперекрывающиеся зоны, как показано на рисунке

Метод доступа **приоритет запросов** реализуется в сети, состоящей из интеллектуальных концентраторов.

Каждый концентратор циклически выполняет опрос своих портов.

Станция, желающая передать пакет, посылает сигнал концентратору, запрашивая передачу кадра и указывая свой приоритет.

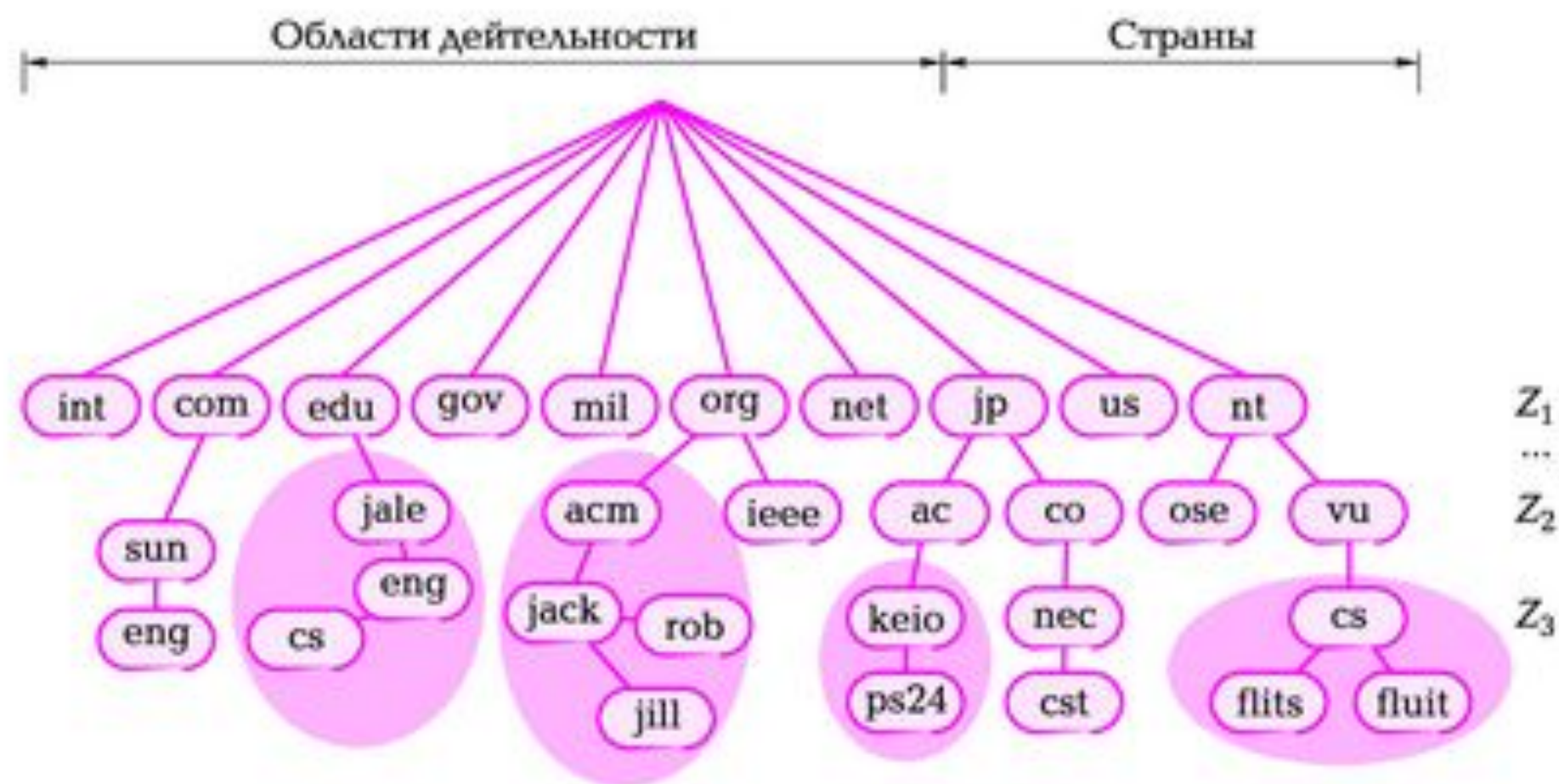


Рис. 1.10. Система доменных имен зоны Интернета

Имена каждой зоны обрабатываются одним сервером имен.

Каждое доменное имя является именем хоста в Интернете, которое ассоциируется с сетевым адресом этого хоста.

В основном интерпретация имени означает получение сетевого адреса соответствующего хоста.

Репликация компонентов распределенной системы применяется при рассмотрении проблем масштабирования, часто проявляющихся в виде падения производительности.

Репликация не только повышает доступность, но и помогает выровнять загрузку компонентов, что ведет к повышению производительности.

В географически рассредоточенных системах наличие близко лежащей копии позволяет уменьшить остроту проблемы ожидания.