

РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я

№ 2

1994

УДК 681.324

В. М. Гольдфельд, А. Я. Иванченко, Д. Л. Литавр
(Новосибирск)

РЕАЛИЗАЦИЯ МНОГОПРОЦЕССНОГО ВЗАИМОДЕЙСТВИЯ
В ЛОКАЛЬНОЙ СЕТИ
С ИСПОЛЬЗОВАНИЕМ КОММУТАЦИОННОГО СЕРВЕРА

Описывается метод обеспечения взаимодействия группы пользовательских процессов, исполняющихся на различных рабочих станциях локальной вычислительной сети. Метод основывается на использовании универсального центрального управляющего процесса (коммутационного сервера), предоставляющего широкие возможности по администрированию сеанса связи. Особое внимание уделено обеспечению секретности и надежности при передаче сообщений, принципам организации коммутационного сервера на базе невыделенной рабочей станции локальной сети.

Введение. Локальные вычислительные сети (ЛВС), объединяющие персональные ЭВМ, получили широкое распространение в мире в последние 15 лет и теперь становятся такой же привычной рабочей средой, как и ПЭВМ. Поэтому все большее значение приобретает предоставление разнообразных услуг конечным пользователям ЛВС.

Существует широкий круг задач распределенной обработки данных, требующих реализации комплексного межпрограммного взаимодействия, подразумевающего возможность передачи и приема сообщений прикладными процессами на рабочих станциях сети. Такие задачи возникают в ситуации, когда несколько прикладных процессов занимаются коллективной однородной деятельностью (например, передачей и приемом пользовательских сообщений по ходу компьютерной телеконференции), при которой связь типа «точка—точка», достаточно легко реализуемая с помощью примитивных функций прикладного программного интерфейса (Application Program Interface — API) [1, 2], уже не удовлетворяет данной задаче. Существующие сетевые операционные системы ЛВС ориентированы в основном на разделение общих ресурсов (файловых серверов, сетевых принтеров и т. д.) и не предоставляют штатной поддержки межпроцессного взаимодействия, полностью оставляя реализацию задач подобного класса разработчикам прикладных программ.

Наиболее простым способом реализации многопроцессного взаимодействия в локальной сети является использование широковещательных сообщений (broadcasts), которые доставляются всем рабочим станциям сети. Очевидным недостатком данного метода является невозможность ограничения набора адресатов и реализации контроля за правами доступа к информации. Известна также техника рассылки групповых сообщений (multicasts), при которой сообщения принимаются только рабочими станциями, пользователи которых относятся к одной определенной группе. Такой метод, однако, затрудняет возможность динамического изменения набора адресатов (участников связи), например, в случае, когда не все члены группы вовлечены в данную работу. При использовании простого дублирования пакетов для всех участников сеанса существенно возрастает объем операций, выполняемых каждым пользовательским процессом, и трафик сети (что особенно отрицательно сказывается

при использовании низкоскоростных каналов связи между удаленными фрагментами сети).

При реализации задач распределенной обработки данных в сети используется также метод управления взаимодействием с помощью выделенного центрального процесса, позволяющий полностью контролировать информационный поток и управлять им. Обычно подобные реализации носят узко специализированный характер и функции центрального процесса содержат в себе специфику конкретной задачи.

Авторами была поставлена задача разработки методик и универсальных программных инструментариев для организации многопроцессного взаимодействия в развитых коммуникационных системах, которые могли бы использоваться при реализации различных задач распределенной обработки данных в локальных сетях. В процессе решения поставленной задачи предпочтение было отдано созданию в качестве центрального управляющего процесса независимого от специфики задачи коммутатора сообщений — коммутационного сервера, представляющего собой программный модуль, запускаемый на рабочей станции сети в фоновом режиме. Прикладной целью данной работы было создание с применением разработанного инструментария пакета программ для организации компьютерных телесовещаний между пользователями, находящимися, в частности, в удаленных фрагментах локальной сети, связанных между собой низкоскоростными линиями.

1. Реализация коммутационного сервера. Коммутационный сервер реализован в сетевой операционной системе NetWare фирмы "Novell", предоставляющей пользователям богатую функциональную среду для создания сложных многопользовательских систем с распределенной архитектурой обработки данных. Такие свойства ОС Novell NetWare [3—5], как производительность, высокий уровень обеспечения защиты информации, широкая совместимость, инвариантность относительно физических и топологических характеристик среды, делают ее особенно привлекательной с точки зрения удобства реализации межпроцессного взаимодействия.

1.1. *Функционирование коммутационного сервера.* Коммутационный сервер, получая информацию от процессов, занимается рассылкой ее остальным процессам, участвующим в сеансе, что позволяет упорядочивать трафик сети (в частности, для удаленных клиентов уменьшать плотность информационного потока через низкоскоростные каналы связи), гибко реагировать на изменение среды (сбой в сети, отключение и подключение процессов), предоставлять некоторые дополнительные расширяемые услуги по контролю и управлению сеансом связи.

Таким образом, организация взаимодействия с использованием коммутационного сервера может иметь преимущества в двух аспектах: в динамической организации групп пользователей и экономичном использовании каналов связи. Однако необходимо определить разумные границы применения этого способа.

Рассмотрим сложную локальную сеть, состоящую из нескольких удаленных фрагментов и рабочих станций (рис. 1), и предположим, что пользователи подчеркнутых рабочих станций участвуют в электронной конференции, реализованной с помощью коммутационного сервера, расположенного во фрагменте (а). Реально каждая станция обменивается сообщениями только с коммутационным сервером, а не друг с другом, но каждый участник конференции получает сообщения всех остальных участников. В этом случае при передаче или приеме одного сообщения станцией, находящейся во фрагменте (b) или (c), через низкоскоростной канал связи проходит только одно сообщение. Однако с увеличением количества участников сеанса во фрагменте (b) увеличивается и количество предназначенных для них пакетов, проходящих через рассматриваемый канал. При этом может быть разумным размещение еще одного коммутационного сервера во фрагменте (b). Однако следует отметить, что при реализации контроля прав доступа и управления сеансом коммутационный сервер использует информацию, хранящуюся на файловом

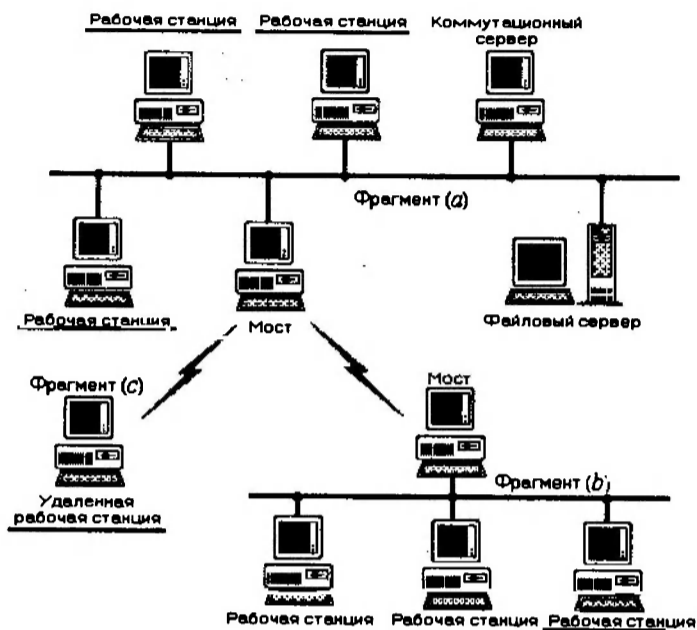


Рис. 1. Пример использования коммутационного сервера

сервере, и потому, по возможности, должен находиться с ним в одном фрагменте (или быть связан с ним достаточно высокоскоростным каналом).

1.2. *Основные принципы построения коммутационного сервера.* Схема взаимодействия сервера с сетевой операционной системой представлена на рис. 2. При разработке коммутационного сервера особое внимание было уделено обеспечению надежного бесперебойного функционирования. Создана библиотека функций для реализации задач в фоновом режиме, позволяющая управлять резидентными программами и гарантирующая безопасность использования в этих программах любых функциональных вызовов DOS и BIOS; разработан специальный алгоритм буферизации, гарантирующий немедленное предоставление буферов для приема пакетов и их рассылку; применены механизмы оперативного отслеживания состояния линий связи. Использование многоуровневой функциональной организации сервера позволяет легко модифицировать пользовательские протоколы и облегчает процесс переноса системы в другую сетевую операционную среду.

Для передачи сообщений между участниками сеанса связи и коммутационным сервером используются базовые коммуникационные протоколы сетевой операционной среды: IPX (датаграммного межсетевое пакетного обмена) и SPX (последовательного пакетного обмена) [2, 6, 7]. При реализации приложений, требующих последовательного корректного обмена сообщениями, безусловно, необходимо использовать протокол SPX. Однако возможны и ситуации, когда применение датаграммного протокола IPX оказывается предпочтительным (например, при коммутации пакетов с речевой информацией). Поэтому коммутационный сервер должен обеспечивать возможность выбора протокола.

Важной частью коммутационного сервера являются контрольно-управляющие функции. В данном случае под контрольно-управляющими функциями понимаются такие возможности, как проверка прав доступа клиентов, учет использования ресурсов сервера, управление расписанием его работы, регистрация сервиса в сети (Service Advertizing), рассылка диагностических сообщений и т. д. Наиболее предпочтительным способом реализации этих функций, взятым за основу при разработке коммутационного сервера, является использование возможностей протокола доступа к ядру операционной

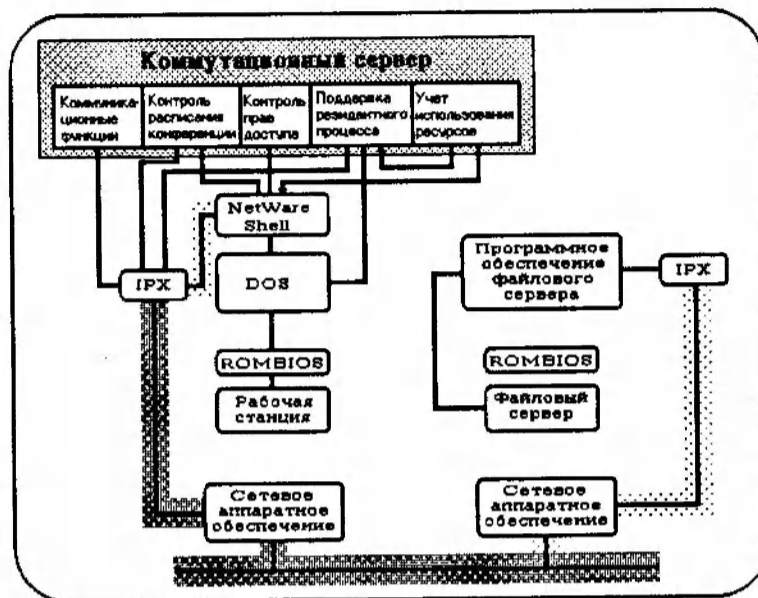


Рис. 2. Функционирование коммутационного сервера в сетевой операционной среде NetWare

системы Novell NetWare (NCP — NetWare Core Protocol) [8]. Например, регистрация сервиса в сети реализуется путем рассылки широковещательных IPX-сообщений и создания соответствующей записи в базе данных управления ресурсами сети (BINDERY). Выдача сообщений об ошибках из фонового процесса реализуется более эффективно с помощью «Службы сообщений» NetWare, чем с помощью стандартных средств ДОС. Использование NCP является достаточно универсальным и весьма эффективным средством при разработке программных продуктов такого рода, автоматически решает проблему контроля прав доступа (в том числе проверку паролей, прав подключения участников к сеансу связи, прав администраторов сеанса) стандартными средствами NetWare и некоторые проблемы функционирования программы в фоновом режиме. Функции протокола NCP активно используются коммутационным сервером.

Для хранения управляющей информации коммутационного сервера как нельзя лучше подходит база данных управления ресурсами ОС Novell NetWare (BINDERY), доступ к которой также осуществляется посредством NCP. В задаче организации многопроцессного сеанса связи BINDERY используется для хранения расписания работы сервера, идентификаторов пользовательских групп, принимающих участие в сеансе связи, идентификаторов администраторов сеанса и сетевого адреса сервера (рис. 3).

Особо необходимо упомянуть об использовании центрального управляющего процесса при организации связи с удаленными рабочими станциями. Испытания разработанной системы, проведенные в локальной сети с удаленными рабочими станциями, продемонстрировали достоинства предложенного решения при использовании медленных линий. Применение коммутационного сервера позволило сделать удаленных клиентов полноправными участниками диалога, в то время как использование групповых сообщений с обеспечением минимального контроля прав доступа сделало бы телеконференцию практически невозможной из-за резкого повышения загруженности медленного канала, связанного с увеличением количества как информационных пакетов, так и служебных сообщений, которыми обмениваются удаленная рабочая станция и сервер для определения прав доступа клиентов. Эта ситуация может быть смоделирована путем помещения коммутационного сервера

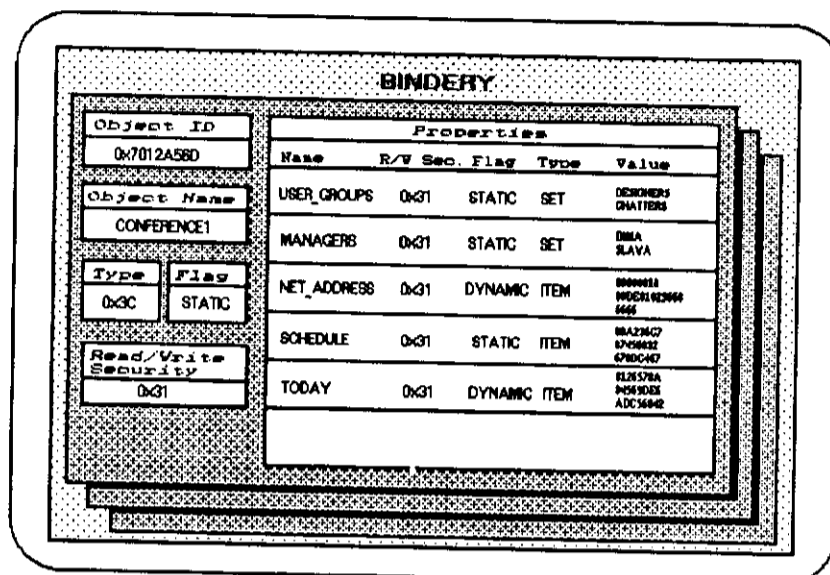


Рис. 3. Формат объекта базы данных управления ресурсами сети, используемого коммутационным сервером

на удаленную рабочую станцию, при этом весь поток сообщений проходит через низкоскоростную линию.

1.3. Структура коммутационного сервера. Коммутационный сервер реализует определенный набор функций, который может варьироваться в зависимости от нужд конкретных коммуникационных задач (с помощью задания определенных ключей при запуске сервера или при компиляции с использованием соответствующих функций, объединенных в специальную библиотеку). Эти функции можно разбить на группы следующим образом:

- коммутация IPX-сообщений;
- коммутация SPX-сообщений;
- проверка прав доступа к информации;
- использование расписания сеансов связи;
- ведение учета деятельности участников сеанса связи.

Функционирование коммутационного сервера осуществляется на трех логически независимых функциональных уровнях. Нижние уровни лишь предоставляют интерфейсные функции для использования их верхними уровнями, что обеспечивает гибкость при разработке конкретных протоколов обмена сообщениями и предоставляет широкие возможности для дальнейших модификаций программного продукта.

Нижний уровень обеспечивает интерфейс с протоколами IPX и SPX, реализованный на основе функций, предоставляемых библиотекой "NetWare C Interface Libraries" фирмы "Novell" и содержащих функциональные вызовы IPX и SPX, реализующие инициализацию виртуальных каналов, установление и разрыв соединений, посылку и прием пакетов, доступ к информации о состоянии соединения, планирование сетевых событий.

Средний уровень пакета содержит функции, реализующие внутренний протокол сеанса связи. Принцип, взятый за основу при разработке этих функций, заключается в том, что весь протокол телеконференции реализован внутри ESR-процедур («Процедуры обработки событий» приема/передачи пакета, установления/разрыва соединения и т. д.), а изменяемая в зависимости от конкретных приложений часть полностью сосредоточена внутри процедуры обработки события приема пакетов, что делает процесс внесения изменений в протокол достаточно удобным для разработчика. Функции, разбирающие принятый пакет, обрабатывающие ошибки приема и передачи, проверяющие права пользователей и ведущих, посылающие пакеты пользователям, вызыва-

ются из ESR-процедур событий приема и передачи. Лишь действия по проверке потерянных слотов во внутренней таблице соединений сервера и их восстановлению при сбойных ситуациях производятся в теле ESR-процедуры диспетчера событий («Диспетчер асинхронных событий» — это дополнительное средство протокола IPX для планирования во времени пользовательских процедур). Таким образом, функционирование коммутационного сервера полностью реализовано с помощью нескольких сложным образом взаимодействующих асинхронных процессов (рис. 4).

К верхнему уровню функций коммутационного сервера относятся супервизорские функции контроля за установленным временем сеанса связи и ведения учета использования ресурсов.

Система контроля расписания реализуется функцией, вызываемой периодически из ESR-процедуры диспетчера событий (аналогичной обработчику прерывания от таймера). В этой функции сравниваются значения времени начала и конца сеанса связи из объекта с расписанием, хранящегося в «Базе данных управления ресурсами» сети (BINDERY), с текущим временем и проверяются значения статусных флагов сервера. На основании собранной информации делается вывод о состоянии текущего сеанса связи и производятся соответствующие действия (например, с помощью функций NetWare Message Service рассылаются сообщения о начале или конце сеанса, запрещается или разрешается его проведение, закрывается сеанс, выдаются предупреждения о завершении выделенного для сеанса связи временного отрезка).

Ведение учета деятельности участников сеанса связи реализовано с помощью специального набора процедур управления резидентным процессом, так как эта услуга требует записи в файлы из резидентной программы, что оказывается невозможным без специальных «приспособлений». Сам по себе процесс учета заключается в создании журнала — файла, содержащего в названии текущую дату, и записи в него информации о начале и конце сеансов, подключении и отключении пользователей. Этот файл в дальнейшем обрабатывается специальной программой супервизора, входящей в состав программного комплекса для реализации межпроцессного взаимодействия. Это позволяет, например, устанавливать оплату за участие в сеансе связи или собирать статистику. При изменении принципов сбора информации не сложно добавить другую необходимую для записи в файл информацию (например, количество переданных пакетов).

Все описанные функции центральной части пакета встроены в общую структуру, обеспечивающую надежное и производительное функционирование резидентного процесса. К этой структуре относятся как ESR-процедуры событий установления соединения, приема и передачи пакетов, процедура «Диспетчера асинхронных событий» (AES), так и специально разработанные функции библиотеки поддержки резидентных программ, позволяющие работать с файлами, использовать функции с прерываниями в резидентных программах. На основе этой библиотеки реализована система коммуникационных прерываний, с помощью которых программа супервизора может отслеживать

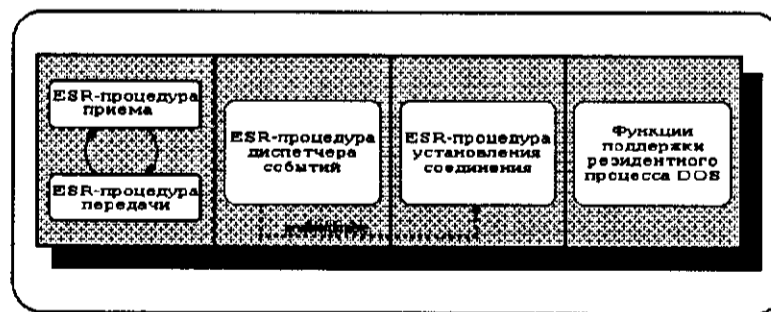


Рис. 4. Структура асинхронных процессов коммутационного сервера

текущую статистику сеанса связи (количество и тип ошибок, число переданных пакетов, динамику использования сервера) и давать некоторые управляющие команды. Нетрудно заметить, что при использовании асинхронных процедур IPX возникает парадоксальная для разработчика программ в DOS ситуация: резидентный процесс может активно функционировать, не захватывая при этом ни одного прерывания. Это, прежде всего, повышает надежность программы, уменьшает объем занимаемой памяти, а также минимизирует загрузку процессора.

Заключение. Таким образом, были разработаны методики и программный инструментарий для организации многопроцессного взаимодействия в многофрагментных локальных сетях с использованием центрального управляющего процесса (коммутационного сервера).

На базе разработанного коммутационного сервера был создан программный комплекс «Телеконференция», предназначенный для проведения различного рода «электронных» совещаний, компьютерных биржевых торгов, аукционов, взаимодействия с удаленными клиентами локальных сетей и т. д. Пакет программ «Телеконференция» состоит из программного обеспечения коммутационного сервера, программы для клиентов и установочного модуля.

Испытания программного комплекса, проведенные в локальной сети с удаленными рабочими станциями, продемонстрировали достоинства предложенного решения при использовании медленных линий. Применение коммутационного сервера дало возможность сделать удаленных клиентов полноправными участниками диалога.

Обобщенность подхода при разработке коммутационного сервера позволила повторно использовать наработанные алгоритмы и программные модули при реализации других задач распределенной обработки данных в локальных сетях.

СПИСОК ЛИТЕРАТУРЫ

1. Durfey D. C. Functionality overview of NetWare APIs // NetWare Techn. Journ.—1989.—N 1.
2. Kroontz M. Chatting with IPX and SPX // Ibid.—N 4.
3. Barkley J. Personal Computer Networks: NBS Special Publication, 1986.
4. Brown J. Novell to lay out NetWare OSI strategy // Network World.—1989.—N 42.
5. Lewis J. Local-Area Network Operating System Report 1987: Novell Inc., 1987.
6. Lewis J. Communicating in NetWare environment // An In-Depth Look At Novell Communications Options: Novell Inc., 1987.
7. McCann J. Skeleton programming with IPX and SPX // NetWare Techn. Journ.—1989.—N 4.
8. Krumrey A. NetWare in control // PC Techn. Journ.—1985.—N 11.

Поступила в редакцию 5 мая 1993 г.