

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное
учреждение высшего образования
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

Т.М. Татарникова, В.А. Миклуш, Е.В. Краева

КОММУТАТОРЫ

Учебное пособие

Санкт-Петербург

РГГМУ

2020

УДК 621.316.5(075.8)

ББК 31.264я73

T12

Татарникова Т.М., Миклуш В.А., Краева Е.В.

Коммутаторы: учеб. пособие. – СПб.: РГТМУ, 2020. – 84 с.

Рецензент: Тюрликов А.М., д-р. техн. наук, профессор, заведующий кафедрой инфокоммуникационных сетей Санкт-Петербургского государственного университета аэрокосмического приборостроения.

Изложен материал для изучения основ моделирования и оценки характеристики центров сопряжения инфокоммуникационных сетей. Материал является дополнением к теоретическому курсу дисциплин «Защищенные корпоративные сети» и «Проектирование защищенные телекоммуникационных сетей». Приведенные в пособии модели и схемы расчетов центров сопряжения также будут полезны при выполнении лабораторных работ соответствующих дисциплин.

Предназначено для подготовки специалистов по специальности «Информационная безопасность телекоммуникационных сетей».

© Т.М. Татарникова, В.А. Миклуш, Е.В. Краева, 2020
© Российский государственный гидрометеорологический университет, 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. ОБЪЕДИНЕНИЕ РАЗНОРОДНЫХ СЕТЕЙ	5
1.1. Соответствие функций ЦС модели OSI	6
1.2. Объединение сетей на физическом уровне	8
1.3. Объединение сетей на канальном уровне	10
1.4. Объединение сетей на сетевом уровне	15
1.5. Объединение сетей на транспортном и выше уровнях	19
1.6. Обобщение функций ЦС	20
2. СТРУКТУРНО-ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ ЦС	22
2.1. Канальный модуль	22
2.2. Процессорный модуль	23
2.3. Схема комплексирования	26
2.4. Общее поле памяти ЦС	30
2.5. Виртуальный канал	31
2.6. Обобщенная модульная структура МУ	32
3. МОДЕЛИ СОГЛАСОВАНИЯ РАЗНОРОДНОСТЕЙ В ЦС	38
3.1. Модель фрагментации/сборки пакетов	38
3.2. Модели согласования скорости сетей, взаимодействующих через ЦС	41
3.2.1. Согласование скоростей по технологии скользящего окна	42
3.2.2. Согласование скоростей с помощью механизма управления темпом передачи	48
3.2.3. Согласование скоростей с помощью механизма «дырявое ведро»	53
4. МОДЕЛИ РАСЧЕТА ВЕРОЯТНОСТНО-ВРЕМЕННЫХ ХАРАКТЕРИСТИК ЦЕНТРА СОПРЯЖЕНИЯ	56
4.1. Вероятностно-временные характеристики ЦС	56
4.2. Математическое обеспечение расчета ВВХ ЦС	58
4.3. Модель коммутатора со схемой комплексирования «матрица»	66
4.4. Модель коммутатора со схемой комплексирования «разделяемая память»	71
4.5. Математическое моделирование коммутатора с общей шиной	75
4.6. Математическое моделирование ЦС высокого уровня	78
4.7. Оценка вероятности потерь в ЦС	81
ЗАКЛЮЧЕНИЕ	83
ЛИТЕРАТУРА	84

ВВЕДЕНИЕ

Успешная работа многих организаций и компаний зависит от средств коммуникаций. Корпоративная сеть есть важнейший информационный ресурс предприятия.

Задача развертывания корпоративной сети имеет свои проблемы. В основном они связаны с организацией эффективного взаимодействия отдельных разнородных сетей в рамках корпоративной сети. Разнородность включает несколько аспектов. В разных сетях – разная длина пакетов, разная скорость передачи данных, зависящая от качества канала и сетевой нагрузки, разная уровневая принадлежность к функциональным возможностям (уровни модели OSI), разная топология, разное программное обеспечение. Все функции по объединению и согласованию выполняются в межсетевых центрах разного уровня и назначения – мостах, коммутаторах, маршрутизаторах и шлюзах, обеспечивающих сопряжение и необходимое преобразование в терминах, как аппаратуры, так и программного обеспечения. Эти центры сопряжения являются основой построения больших корпоративных сетей и определяют ее распределенность и топологию.

Межсетевые центры сопряжения выполняются в виде многопроцессорных многопортовых систем, так как, выполняя основную функцию взаимодействия нескольких неоднородных сетей, обрабатывают большие потоки информации (нагрузка на центр суммируется), что приводит к необходимости распараллеливания во времени функций приема, передачи и обработки поступающих на его вход пакетов. Основное достоинство параллельных систем заключается в том, что они могут динамически настраиваться под конкретные внешние условия, в частности, за счет перераспределения своей мощности в зависимости от приоритета решаемых задач.

Разнообразие названных групп центров сопряжения (ЦС) и архитектурных решений внутри каждой группы обеспечивают выбор ЦС под конкретные условия объединения разнородных сетей при требуемом качестве обслуживания и заданном либо прогнозируемом трафике. Выбор в основном реализуется из сообразительности цены, количества портов ввода-вывода, требуемой производительности, набора интеллектуальных функций ЦС.

В учебном пособии рассматривается методическое обеспечение выбора ЦС с заданным набором свойств, а также моделей и схем расчета оценивания свойств ЦС в зависимости от степени разнородности взаимодействующих сетей.

1. ОБЪЕДИНЕНИЕ РАЗНОРОДНЫХ СЕТЕЙ

Объединение разнородных сетей в единую интересеть осуществляется на основе территориальных сетей. Территориальные сети решают проблему формирования глобальных сетей из локальных сетей, сетей регионов и целых стран и даже наднациональных сетей (например, E-BONE для Европы, Internet для Мира). Как правило, эти сети строятся с использованием протоколов ATM, ISDN, Frame Relay или X.25, TCP/IP [4].

Сети, объединяемые в единую интересеть, могут существенно различаться по протоколам, адресации (плоская или иерархическая), по размерам пакетов, по параметрам, например таким как продолжительность тайм-аутов, время жизни пакета или максимальный размер пакета и по многим другим.

Стандартным решением, при котором разнородные сети функционируют как единая сеть, все узлы которой имеют возможность обмениваться информацией, называется трансляцией.

Трансляция протоколов обеспечивает согласование двух протоколов путем преобразования (трансляции) сообщений, поступающих от одной сети, в формат другой сети. Транслирующий элемент, в роли которого выступает программный или аппаратный шлюз, мост, коммутатор или маршрутизатор размещается между взаимодействующими сетями и служит посредником в их "диалоге". Часто транслятор протоколов называют шлюзом в широком смысле, независимо от того, какие протоколы он транслирует. Этим подчеркивается тот факт, что трансляция осуществляется выделенным устройством, соединяющим две разнородные сети.

Трансляция протоколов представляет собой сложный процесс, включающий преобразование форматов сообщений, отображение адресов сетей и узлов, различным образом трактуемых в этих протоколах, согласование величин тайм-аутов, времен жизни пакета, максимальных размеров пакетов, структуры пакетов и многое другое [2].

При использовании техники трансляции не требуется устанавливать дополнительное программное обеспечение на рабочих станциях, сохраняется привычная среда пользователей и приложений транслятор полностью прозрачен для них. Проблемы межсетевого взаимодействия локализованы, следовательно, упрощается администрирование, поиск неисправностей, обеспечение безопасности.

Однако, транслятор замедляет работу из-за относительно больших временных затрат на сложную процедуру трансляции, а

также из-за ожидания запросов в очередях к центру сопряжения, через который проходит весь межсетевой трафик. При увеличении сетевой нагрузки резко снижается производительность центра сопряжения.

1.1. Соответствие функций ЦС модели OSI

На рис. 1.1 представлено распределение известных групп ЦС по уровням модели OSI.

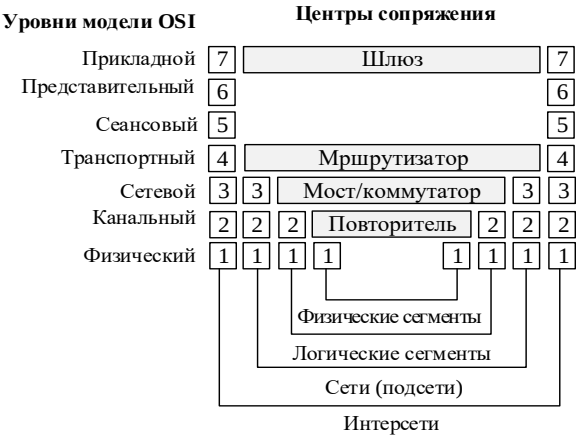


Рис. 1.1. Соответствие функций ЦС модели OSI

На физическом уровне сети объединяются повторителями или концентраторами, которые просто переносят бит из одной сети в другую такую же сеть, и преобразовывают сигналы при сопряжении различных физических сред.

Мосты и коммутаторы работают на канальном уровне. Они могут принимать блоки данных, анализировать их физические адреса, направлять их в другие сети, осуществляя по ходу минимальные преобразования протоколов.

На сетевом уровне маршрутизаторы соединяют две сети. Если сетевые уровни у них разные, маршрутизатор может обеспечить перевод пакета из одного формата в другой. Маршрутизатор, поддерживающий несколько протоколов называется мультипротокольным маршрутизатором.

На транспортном уровне существуют транспортные шлюзы, предоставляющие интерфейсы для соединений сетей, у которых протоколы на этом уровне разные. Например, транспортный шлюз

реализует передачу пакетов из одной сети в другую, склеивая одно соединение с другим.

На прикладном уровне шлюзы занимаются преобразованием семантики сообщений.

Многоуровневая трансляция протоколов порождает необходимость модифицировать на каждом уровне передаваемые сообщения применительно к функциям, реализуемым на этом уровне. Данные, передаваемые в форме сообщения, снабжаются *заголовком* (З) и *концевиком* (К) (рис. 1.2), в которых содержится информация, необходимая для обработки сообщения на соответствующем уровне: указатели типа сообщения, адреса отправителя, получателя, канала, порта и т. д.

Средства управления нижнего уровня оперируют с данными, указанными в оформлении, как с данными на конверте. При передаче на вышестоящий уровень сообщение «освобождается от конверта», в результате чего на следующем уровне обрабатывается очередная «конверт». Таким образом, каждый уровень управления оперирует не с самим сообщением, а только с «конвертами», в которые «упаковано» сообщение.

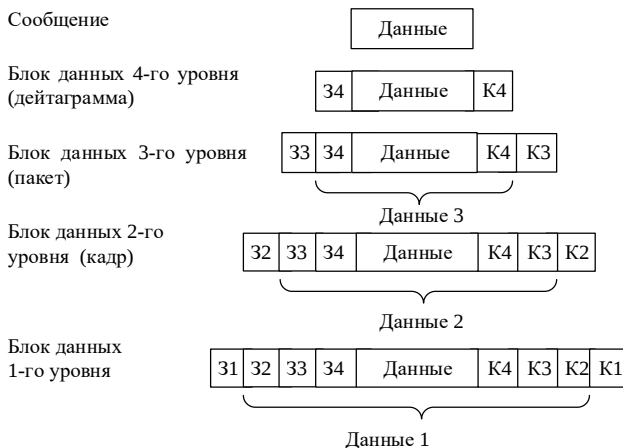


Рис. 1.2. Вложенность протокольных блоков данных различных уровней

Рассмотрим типичный сценарий работы ЦС при объединении двух разных сетей, например Ethernet и Internet. Работу ЦС можно представить, как работу узла, состоящего из двух половин, каждая из которых работает под управлением стека протоколов сопрягаемой сети

– левая как узел сети 1, а правая как узел сети 2 (рис. 1.3). Во входной порт 1 центра сопряжения приходят данные сети 1. Они передаются последовательно на физический, канальный, сетевой и транспортный уровни согласно протоколам сети 1, претерпевая. Когда все конверты будут убраны, то формируются новые блоки данных, формат которых принят в сети Internet. Таким образом передача данных в выходной порт 2 происходит по протоколам сети 2: на транспортном уровне добавляется заголовок TCP, на сетевом – заголовок IP, в результате чего формируется IP-пакет, который отправляется на канальный уровень, где обрывается еще одним заголовком и контрольной суммой (CRC) и наконец в формате кадра спускается на физический уровень для передачи в сеть 2.

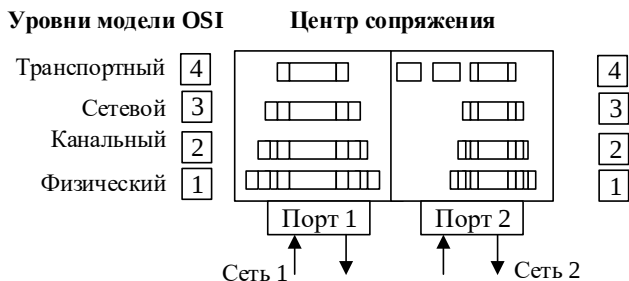


Рис. 1.3. Сопряжение двух разных сетей через ЦС

Все необходимые преобразования по сопряжению выполняются средствами программного обеспечения ЦС, реализующего согласование форматов пакетов, скоростей передачи данных и протоколов, принятых в сопрягаемых сетях [5].

1.2. Объединение сетей на физическом уровне

На физическом уровне пакет представляет собой счетную последовательность импульсов, распространяющихся по кабелю. За счет частичного отражения от точек подключения и поглощения в среде импульсы в пакете «расплываются» и искажаются (ухудшается отношение сигнал/шум), что является одной из причин ограничения длин кабельных сегментов. Для преодоления этих ограничений и вводятся сетевые повторители (рис. 1.4).

Повторитель принимает сигналы из одного сегмента кабеля и побитно синхронно повторяет их в другом сегменте, улучшая форму и мощность импульсов, а также синхронизируя импульсы. Таким

образом, повторитель используется для физического соединения сегментов кабеля сети с целью увеличения общей длины шины. Например, в классическом Ethernet допускается установка четырех повторителей, что позволяет расширять максимальную длину кабеля с 500 до 2500 м [3].

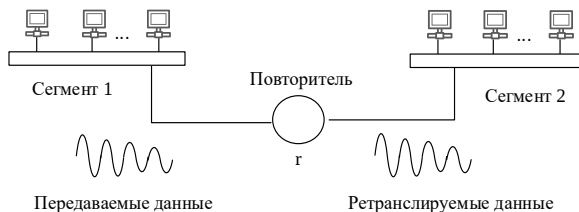


Рис. 1.4. Усиление и ретрансляция сигналов повторителем

Задержка сигнала повторителем не должна превышать 7,5 тактов (750 нс для Ethernet). Все входы/выходы повторителя с точки зрения пакетов эквивалентны.

Многопортовый повторитель называют *концентратором* (hub). Концентраторы образуют из отдельных физических отрезков кабеля общую среду передачи данных – логический сегмент. Когда разные компьютеры, подключенные к концентратору, начинают передачу с небольшим сдвигом во времени, сигналы – переносчики пакетов, в общей физической среде «сталкиваются» и искажаются, возникает конфликт (коллизия – collision). Пакеты, участвующие в коллизии – уничтожаются. Компьютеры должны подождать произвольное количество времени и заново передать уничтоженные пакеты. По мере увеличения числа компьютеров, подключенных к концентратору, увеличивается и число столкновений, а по мере увеличения столкновений снижается эффективность сети. Именно по этой причине ЦС канального уровня (мосты и коммутаторы) практически полностью вытеснили концентраторы.

Таким образом, на физическом уровне сети объединяются повторителями и концентраторами, которые являются аналоговыми устройствами, сопрягающими сегменты кабеля [2]. И те, и другие не анализируют поступающие данные и не используют адреса. Функцией ЦС физического уровня является простой перенос бит из одной сети в другую такую же сеть, и преобразование сигналов при сопряжении различных физических сред.

1.3. Объединение сетей на канальном уровне

На канальном уровне сопряжение реализуется средствами мостов и коммутаторов [4].

Мост (bridge) – это устройство, которое обеспечивает взаимосвязь двух (реже нескольких) локальных сетей (сегментов) посредством передачи кадров из одной сети (сегмента) в другую(ой) с помощью их промежуточной буферизации (рис. 1.5).

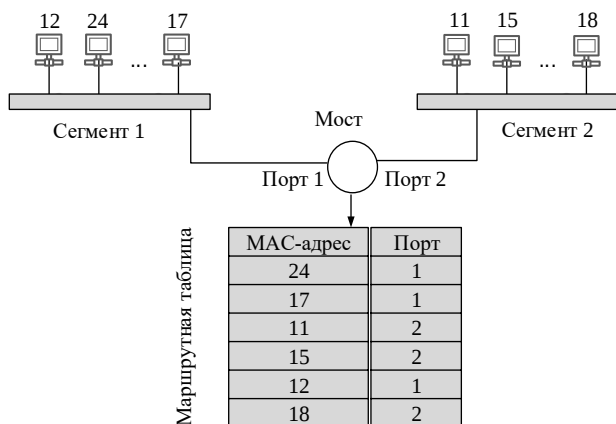


Рис. 1.5. Мост как коммуникационное устройство канального уровня

Мост по адресу источника составляет списки рабочих станций (PC) сети (сегмента) и формирует маршрутную таблицу. Она содержит для каждого MAC-адреса соответствующий номер порта моста, к которому подключены PC сети (сегмента). Мост выступает по отношению к каждой из сетей (сегментов) как конечный узел, который: принимает кадр, буферизует его, по полю адреса приемника проверяет список.

Когда пакет поступает во входной порт моста, то из его заголовка извлекается и анализируется адрес назначения. Адрес сопоставляется с содержимым маршрутной таблицы и определяется номер порта, куда пакет должен быть передан. Возможны три варианта:

1. Если адрес отправителя и получателя совпадают, то пакет игнорируется.
2. Если адрес отправителя и получателя различаются, то пакет передается в соответствующий порт ЦС.
3. Если сеть получателя неизвестна, то пакет уничтожается.

Этот алгоритм применяется для каждого входящего в мост кадра. Процессор отслеживает и обновляет записи в таблице каждые несколько микросекунд.

Мосты обеспечивают выполнение функций канального уровня путем поддержки протоколов канального уровня (сети Ethernet, Token Ring, FDDI).

Благодаря функции фильтрации пакетов мост изолирует трафик одного сегмента от трафика другого сегмента, при этом коэффициент загрузки сегментов уменьшается.

Мост не только снижает нагрузку в объединенной сети, но и уменьшает возможности несанкционированного доступа благодаря изолированности сегментов. Производительность моста должна превышать среднюю интенсивность межсегментного трафика. Буферная емкость моста рассчитывается, исходя из пиковой нагрузки.

Коммутаторы были разработаны с использованием мостовых технологий. Разница между мостом и коммутатором состоит в том, что мост в каждый момент времени может осуществлять передачу кадров только между одной парой портов, а коммутатор одновременно поддерживает потоки данных между всеми своими портами. Другими словами, мост передает кадры последовательно, а коммутатор – параллельно.

Коммутатор (switch – переключатель) – центр сопряжения, предназначенный для соединения нескольких узлов в пределах одного или нескольких сегментов сети и/или нескольких сетей. Характерный пример применения коммутаторов представлен на рис. 1.6

Коммутатор – мультипроцессорная система. В каждом порту коммутатора свой процессор и буферная память для временного хранения пакетов.

В ассоциативной памяти коммутатор хранит таблицу коммутации, в которой указывается соответствие MAC-адреса узла порту коммутатора. При включении эта таблица пуста, и коммутатор работает в режиме обучения. В этом режиме поступающие на какой-либо порт данные передаются на все остальные порты. При этом коммутатор анализирует кадры и, определив MAC-адрес хоста-отправителя, заносит его в таблицу на некоторое время.

Впоследствии, если на один из портов коммутатора поступит кадр, предназначенный для хоста, MAC¹-адрес которого уже есть в таблице, то этот кадр будет передан только через порт, указанный в таблице. Если MAC-адрес хоста-получателя не ассоциирован с каким-

¹ MAC-адрес – уникальный идентификатор PC в локальной сети

либо портом коммутатора, то кадр будет отправлен на все порты за исключением того, с которого он был получен. Со временем коммутатор строит таблицу для всех активных MAC-адресов, в результате чего трафик локализуется.

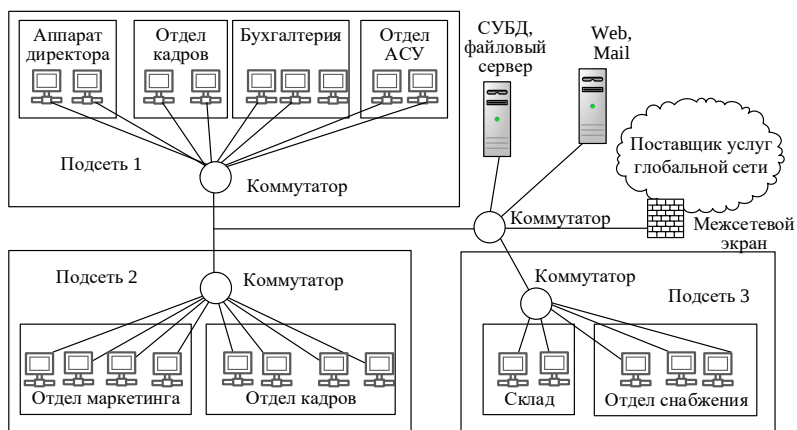


Рис. 1.6. Схема локальной вычислительной сети организации

Коммутаторы характеризует малая задержка и высокая скорость пересылки на каждом порту интерфейса.

Существуют три архитектурных решения аппаратной реализации коммутаторов, различающиеся способами комплексирования его функциональных модулей. Это коммутаторы на основе матрицы, общей шины и с общей памятью.

Коммутаторы на основе матрицы. Коммутатор матричного типа обеспечивает самый быстрый способ взаимодействия входных портов с выходными. Построение таких коммутаторов осуществляется на основе двоичных коммутационных элементов с двумя входами и двумя выходами.

Пример реализации коммутационной матрицы для восьми портов дан на рис. 1.7. Во входном порту по адресу назначения, записанному в служебной части информационного кадра, на основании просмотра адресной таблицы определяется номер выходного порта. Эта информация добавляется к байтам исходного кадра в виде специального ярлыка – тега (tag). Для данного примера тег представляет собой 3-разрядное двоичное число, соответствующее номеру выходного порта.

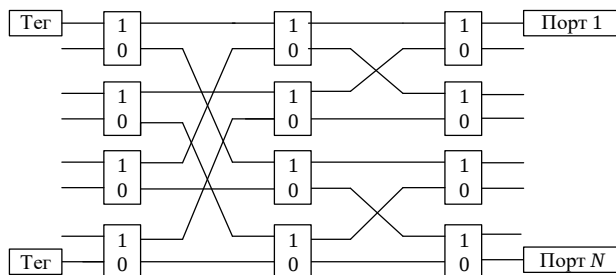


Рис. 1.7. Вариант реализации коммутационной матрицы

Матрица состоит из трех каскадов двоичных переключателей – коммутационных элементов, которые соединяют свой вход с одним из двух выходов в зависимости от значения бита тега.

Коммутационный элемент (КЭ) может работать в одном из двух режимов: «транзит» или «кросс» (рис. 1.8).

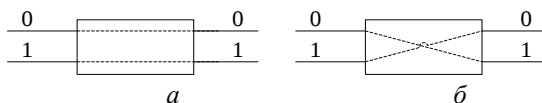


Рис. 1.8. Режимы работы коммутационного элемента:
а) —«транзит» б) —«кросс»

Переключатели первого каскада управляются первым битом тега, второго – вторым, а третьего – третьим. Известным недостатком этой технологии является отсутствие буферизации данных внутри коммутационной матрицы – если составной канал невозможно построить из-за занятости выходного порта или промежуточного КЭ, то данные должны накапливаться в буферных запоминающих устройствах порта коммутатора.

Коммутаторы на базе общей шины. Коммутаторы с общей шиной для связи входных портов с выходными применяют высокоскоростную шину, используемую в режиме разделения времени. В этой архитектуре шина (моноканал) пассивна, а активную роль выполняют специализированные процессоры портов. Пример такой архитектуры приведен на рис. 1.9.

Кадр передается по шине в псевдопараллельном режиме небольшими частями (по несколько байт). Размер такой ячейки данных определяется производителем коммутатора.

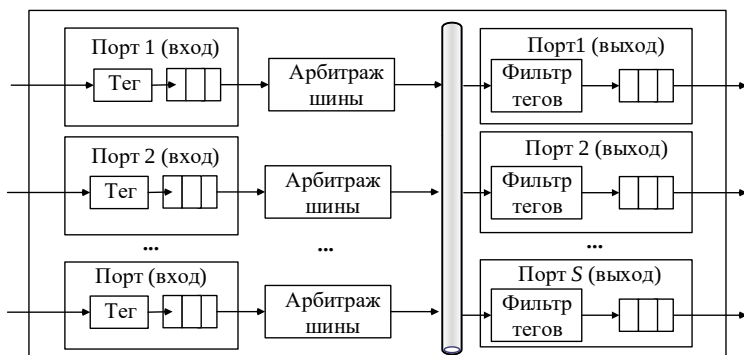


Рис. 1.9. Структура коммутатора на базе общей шины

Во входном порту формируется тег, в котором указывается номер порта назначения, и который добавляется к информационной ячейке, переносимой по шине. Каждый выходной порт содержит фильтр тегов, который выбирает только те теги, которые предназначены данному порту

Шина не может осуществлять промежуточную буферизацию. Однако поскольку доступ портов к шине осуществляется циклически, задержка при доступе накапливается за счет промежутков ожидания между частями транспортируемого по шине пакета. Для того чтобы шина не была узким местом коммутатора, ее производительность должна быть в несколько раз выше скорости поступления данных на входные порты.

Коммутатор с общей разделяемой памятью. В коммутационной схеме с общей разделяемой памятью входные и выходные порты коммутатора соединены между собой не через шину, а через общую память. Пример такой архитектуры приведен на рис. 1.10.

Входные порты (точнее специализированные процессоры этих портов) соединяются с переключаемым входом разделяемой памяти, а выходные – с переключаемым выходом этой памяти. Переключением входа и выхода разделяемой памяти управляет менеджер очередей. Он организует в разделяемой памяти несколько очередей данных, по одной для каждого выходного порта.

Входные порты передают менеджеру запросы на запись данных в очередь того порта, который соответствует адресу назначения пакета. Менеджер по очереди подключает вход памяти к одному из входных портов, и тот переписывает данные в очередь определенного выходного порта. По мере заполнения очередей менеджер производит также поочередное подключение выхода разделяемой памяти к

выходным портам, и данные из очереди переписываются в выходной буфер соответствующего порта.

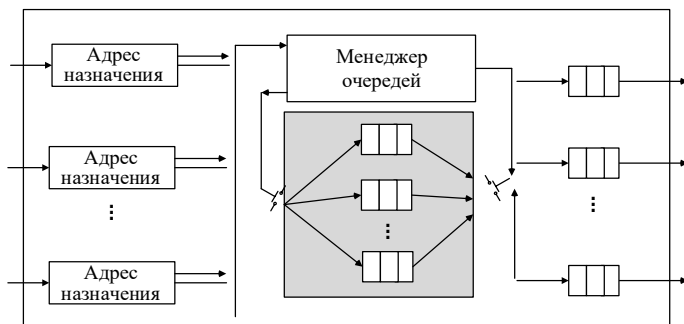


Рис. 1.10. Структура коммутатора на базе общей памяти

К недостаткам коммутаторов этого типа относят их сложность и высокую стоимость.

Функции коммутаторов постоянно расширяются. Современные коммутаторы по своим возможностям приближаются к маршрутизаторам, а по некоторым параметрам корпоративные коммутаторы даже опережают их. Так, например, классификация трафика на транспортном уровне позволяет интеллекту коммутаторов различать уже не только отдельные IP-пакеты с различными установленными классами обслуживания, но и различные типы вышестоящих протоколов (например, HTTP, FTP, SMTP) благодаря анализу заголовков TCP-пакетов. Таким образом коммутаторы стали основным типом ЦС на первом, втором и третьем уровнях OSI, заменяя концентраторы, мосты и маршрутизаторы при объединении локальных сетей.

1.4. Объединение сетей на сетевом уровне

Трансляция сетевых протоколов является более сложной задачей, чем трансляция канальных протоколов, хотя бы потому, что в отличие от канального уровня, на котором имеется единая система уникальных адресов узлов, каждый сетевой протокол имеет собственный, свойственный только ему, формат адресов. Так, преобразование протокола Ethernet в протокол Token Ring сводится к нескольким несложным действиям, потому что в обоих протоколах используется единая адресация узлов. А вот трансляция протоколов сетевого уровня

IP и IPX – более сложный процесс, включающий не только преобразование форматов сообщений, но и отображение адресов сетей и узлов, различным образом трактуемых в этих протоколах [2].

Кроме различий в системе адресации, в каждом сетевом протоколе имеется еще много других специфических особенностей, которые могут выражаться в различии как количественных параметров (например, для разных протоколов могут быть определены разные величины тайм-аутов, времен жизни пакета или максимальных размеров пакетов), так и в структуре пакетов. Протоколы могут отличаться и функциональными возможностями, например, одни из них реализованы с установлением соединений, а другие – без установления соединений, в одних – предусмотрена возможность фрагментации, в других – нет.

Итак, сопряжение на сетевом уровне имеет свои особенности и отличается от объединения сетей на канальном уровне. На рис. 1.11 источник *S* пытается послать пакет приемнику *D*. Эти две машины работают в разных сетях Ethernet, соединенных коммутатором. Источник *S* вставляет пакет в кадр и отправляет его. Кадр прибывает на коммутатор, который по MAC-адресу определяет, что его надо переслать в ЛВС 2. Коммутатор просто снимает кадр с ЛВС 1 и передает его в ЛВС 2.

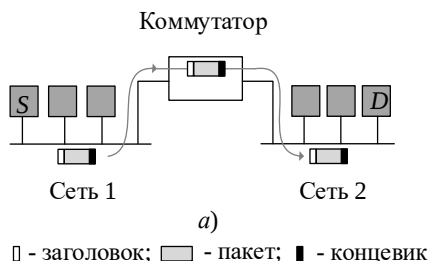


Рис. 1.11. Две сети Ethernet, объединенные коммутатором

Допустим, две сети Ethernet объединены не коммутатором, а парой маршрутизаторов (рис. 1.12). Маршрутизаторы между собой соединены выделенной двухточечной линией. Кадр принимается маршрутизатором, из его поля данных извлекается пакет. Маршрутизатор анализирует содержащийся в пакете адрес (например, IP-адрес). Этот адрес нужно отыскать в таблице маршрутизации. В соответствии с ним принимается решение об отправке пакета (возможно, упакованного в кадр нового вида – это зависит от протокола, используемого линией) на удаленный маршрутизатор. На противоположном конце пакет вставляется в поле данных кадра Ethernet и поступает в ЛВС 2.

В чем заключается основная разница между случаем коммутации (установки моста) и маршрутизации? Коммутатор (мост) пересылает весь кадр, обосновывая свое решение значением MAC-адреса. При применении маршрутизатора пакет извлекается из кадра, и для принятия решения используется адрес, содержащийся именно в пакете. Коммутаторы не обязаны вникать в подробности устройства протокола сетевого уровня. А маршрутизаторы обязаны.

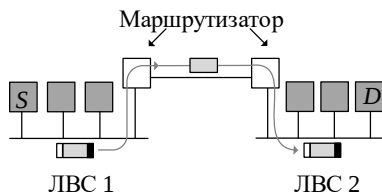


Рис. 1.12. Две сети Ethernet, объединенные маршрутизатором

Маршрутизаторы сетевого уровня соединяют две сети. Если сетевые уровни у них разные, маршрутизатор может обеспечить перевод пакета из одного формата в другой. Маршрутизатор, поддерживающий несколько протоколов называется мультипротокольным маршрутизатором.

Маршрутизаторы образуют логические сегменты посредством явной адресации, поскольку используют не плоские аппаратные, а составные числовые адреса (рис. 1.13). В этих адресах имеется поле номера сети, так что все компьютеры, у которых значение этого поля одинаково, принадлежат к одному сегменту, называемому в данном случае *подсетью*.

Маршрутизатор позволяет организовывать в сети избыточные связи. Он «видит» всю картину связей подсетей друг с другом, на основе чего составляет маршрутную таблицу и может выбрать наиболее подходящий маршрут при наличии нескольких альтернативных.

Маршрутизаторы не только объединяют сети, но и надежно защищают их друг от друга – маршрутизатор отказывается передавать «неправильный» пакет дальше, изолируя дефектный узел от остальной сети.

Чтобы составить карту связей в сети, маршрутизаторы обмениваются специальными служебными сообщениями.

Функции маршрутизатора могут быть разбиты на три группы в соответствии с уровнями модели OSI: уровень интерфейсов, уровень сетевого протокола, уровень протоколов маршрутизации (рис. 1.14).

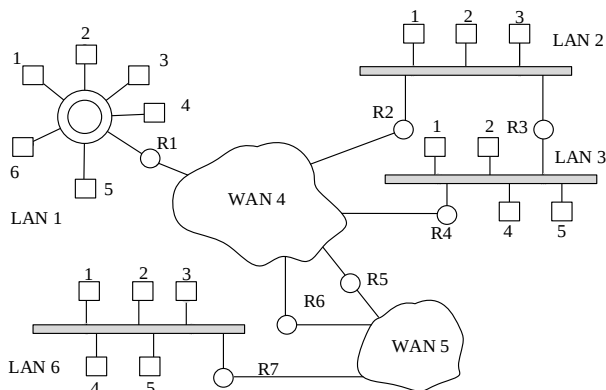


Рис. 1.13. Структура интерсети, построенной на основе маршрутизаторов:
R1, R2, ..., R7 – маршрутизаторы; LAN1, LAN2, LAN3, WAN4, WAN5, LAN6 – уникальные номера сетей в едином формате; PC1, PC2, ... – локальные номера узлов (рабочих станций)

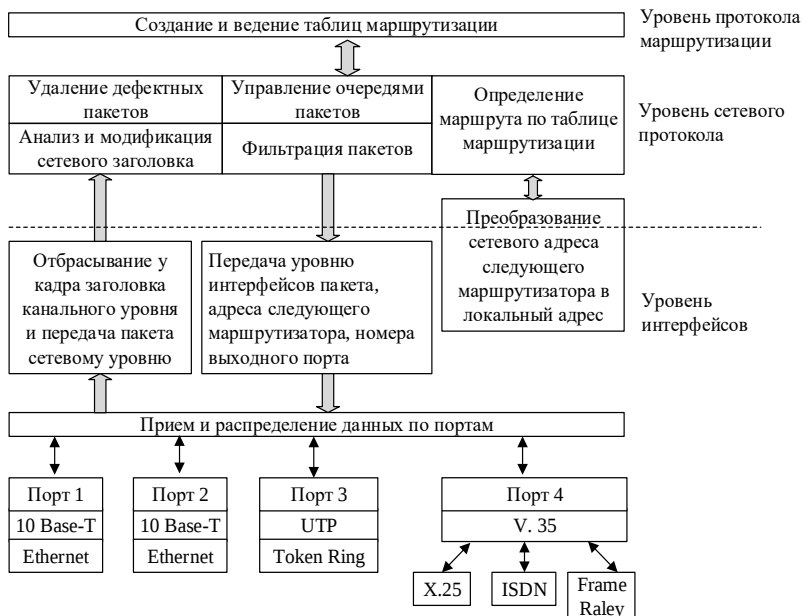


Рис. 1.14. Функциональная модель маршрутизатора

Построением и поддержанием таблицы маршрутизации занимаются *протоколы маршрутизации* (RIP, OSPF, ICMP). Они обеспечивают обмен между маршрутизаторами информацией о топологии сети, анализ этих сведений, определение наилучших маршрутов. Результаты заносятся в таблицы маршрутизации.

Различают три класса маршрутизаторов: магистральные, региональные, офисные. Все региональные и офисные маршрутизаторы практически заменяются быстродействующими управляемыми коммутаторами.

1.5. Объединение сетей на транспортном и выше уровнях

Специфические особенности сопряжения разнородных сетей делают задачу трансляции сетевых протоколов нетривиальной, требующей привлечения программных средств. Узел, реализующий трансляцию одного сетевого протокола в другой, называется шлюзом. Некоторые шлюзы решают и более сложную задачу согласования стеков протоколов, включающих протоколы всех уровней.

Шлюзы поддерживают взаимодействие сетей, которые используют несовместимые протоколы. Шлюзы обеспечивают соединение и необходимые преобразования в терминах как аппаратуры, так и программного обеспечения.

Шлюз – это точка сети, которая служит выходом в другую сеть. Например, сервер, контролирующий трафик между локальной сетью компании и сетью Интернет – это шлюз.

Основная задача шлюза – конвертировать протокол между сетями. Шлюз должен понимать все протоколы, используемые в объединенной сети.

В каждой из сетей, образующих интернет, сохраняются присущие им принципы адресации узлов и протоколы обмена информацией, поэтому шлюзы могут объединять не только локальные сети с различной технологией, но и локальные сети с глобальными сетями. Эти особенности делают шлюз сложным интеллектуальным устройством, построенным на базе нескольких мощных процессоров (рис. 1.15). Такой мультипроцессор работает, как правило, под управлением специализированной операционной системы [5].

Когда пакеты прибывает в шлюз сети, то в порту от них отрезаются заголовки и концевики и остаются только поля данных, которые передаются в общее поле памяти шлюза. Далее анализируется заголовок пакета и в соответствии с записанным в нем заданием строится последовательный алгоритм (цепочка команд) обработки

пакета протокольными процессами. В шлюзе одновременно выполняется несколько заданий, так как протоколы могут иметь свои копии по уровням OSI, а общая память разделена на секции. Это обеспечивает параллельную обработку пакетов в шлюзе.

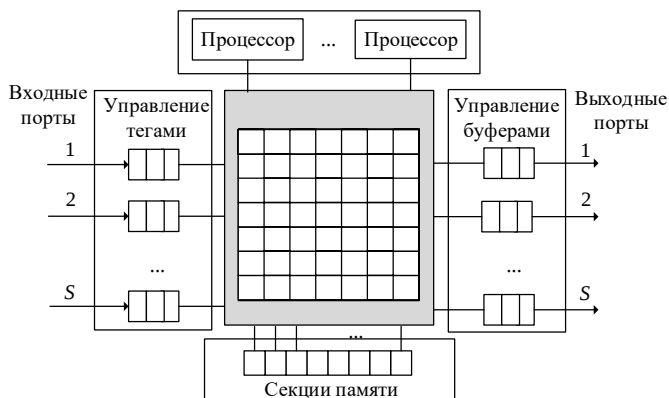


Рис. 1.15. Архитектура шлюза

Шлюз может принять пакет, сформированный под один протокол (например, Apple Talk) и конвертировать его в пакет другого протокола (например, TCP/IP) перед отправкой в другой сегмент сети (другую сеть). Шлюзы работают медленнее, чем коммутаторы.

В крупных сетях сервер, работающий как шлюз, обычно интегрирован с прокси-сервером и межсетевым экраном.

На транспортном уровне шлюз интегрированы с маршрутизатором. Транспортный шлюз позволяет, к примеру, передавать пакеты из сети TCP в сеть SNA (протоколы транспортного уровня у них различаются).

На прикладном уровне шлюзы преобразуют семантику сообщений. Например, шлюзы между электронной почтой Интернета (RFC 822) и электронной почтой X.400 должны анализировать содержимое сообщений и изменять различные поля электронного конверта.

1.6. Обобщение функций ЦС

Обобщим функции рассмотренных центров сопряжения корпоративной сети.

Функционирование ЦС – это многоэтапный процесс буферизации

данных, поступающих на вход ЦС, который включает следующие этапы:

- прием блоков данных: входящие в ЦС данные записываются в буферную память входного порта со скоростью канала передачи данных;

- обработка блоков данных: данные обрабатываются процессором (процессорами) согласно функциям трансляции ЦС и определение исходящего порта;

- передача блоков данных: данные передаются по каналу в выходящем из ЦС направлении.

Копия принятого в ЦС сообщения хранится в буферной памяти до получения квитанции АСК (логического подтверждения) о безошибочной доставке сообщения адресату или истечении time-out.

ЦС типа маршрутизатор или шлюз содержат таблицы маршрутизации, в которые записываются номера портов и идентификаторы соединений, присутствующие в заголовке каждого пакета.

ЦС принимают решение о том, на какой порт нужно передать кадр, анализируя адрес назначения, помещенный в заголовке кадра.

Быстродействие МУ обеспечивается параллельной пересылкой пакетов между портами.

2. СТРУКТУРНО-ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ ЦС

Исходя из анализа функциональных схем разных видов коммутаторов, маршрутизаторов и шлюзов можно выделить структурные элементы (модули), на которых строится ЦС:

- канальный модуль (КМ) – для выполнения процессов приема-передачи кадров;

- процессорный модуль – для выполнения уровневых протоколов ЦС. Программный модуль состоит из совокупности процессоров обработки пакетов, назовем их обрабатывающими модулями (ОМ);

- модуль памяти – общее поле памяти (ОПП) – для хранения общих информационных ресурсов, необходимых при выполнении протоколов и хранения блоков данных, передаваемых по ЦС;

- схема комплексирования (СК) – для объединения модулей ЦС;

- виртуальный канал (ВК) – для отображения входов/выходов ЦС и поступающей на него нагрузки.

Рассмотрим перечисленные структурные модули и выполняемые ими функции [5].

2.1. Канальный модуль

КМ – это прежде всего обрабатывающее устройство. В его структуру входят процессор, порт входа-выхода – два блока буферной памяти по одному на вход и выход, а также два контроллера прямого доступа (КПД).

Процессор канального модуля выполняет протоколы и функции второго уровня OSI, то есть реализует операции генерации и выделения межкадровых разделителей, битстаффинга, циклического кодирования-декодирования, побитовой передачи и формирование кадра длиной L бит.

КПД – контроллер прямого доступа – осуществляет прием/передачу данных в/из буферной памяти КМ.

Структура КМ приведена на рис. 2.1.

КМ предназначен для накопления блоков данных по каждому каналу и записи их в общую память ЦС с последующей обработкой и передачей в исходящий ВК.

За каждым КМ в общей памяти закреплены две кольцевые очереди – одна на прием (входящая), другая на передачу (исходящая), которые формируются операционной системой, под управлением которой работает ЦС. Кольцевая очередь представляет собой таблицу логических ссылок, связывающих адреса хранения кадров.

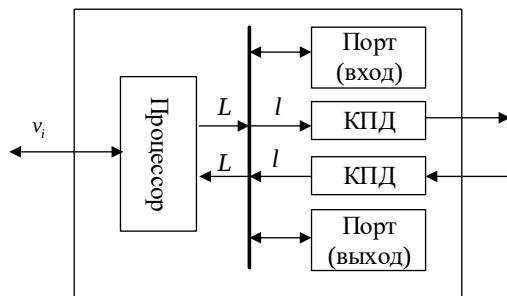


Рис. 2.1 Структура КМ

Приемный КПД при регистрации процессором первого бита кадра определяет в кольцевой очереди свободное место. Если свободных мест нет, то кадр теряется. Поступление новых кадров вызывает новые обращения КПД в кольцевую очередь КМ.

В исходящую кольцевую очередь КМ записываются адреса тех кадров, которые должны быть переданы в соответствующий исходящий канал. Таким образом в некоторую исходящую кольцевую очередь записываются адреса кадров из любых входящих кольцевых очередей. Передающий КПД выбирает из исходящей кольцевой очереди адрес кадра, подлежащего передаче и переписывает его в порт, который затем побитно выводится процессором КМ в исходящий канал.

Если исходящая кольцевая очередь в данный момент времени не содержит кадров для передачи, то процессор КМ генерирует в исходящий канал межкадровые разделители. Проверка наличия появления кадра выполняется путем повторных обращений в кольцевую очередь.

2.2. Процессорный модуль

Процессорный модуль состоит из набора процессоров обработки пакетов (обрабатывающих модулей (ОМ)).

Каждый ОМ реализует логическую обработку блока данных, поступающего в ЦС согласно протоколу трансляции, реализованному в нем и состоит из процессора, оперативного запоминающего устройства (ОЗУ) для хранения рабочих данных и локальной постоянной памяти для хранения программ выполняемых им протоколов.

Все процессоры, входящие в состав ОМ идентичны и имеют номинальное быстродействие B_n^{OM} операций/с. Объем и цикл ОЗУ всех ОМ одинаков. Объем ЛП зависит от реализуемых протоколов трансляции. Структура ОМ приведена на рис. 2.2.

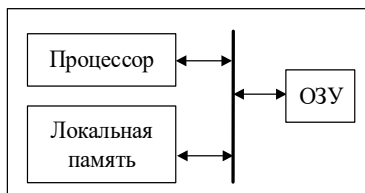


Рис. 2.2. Структура ОМ

Каждый ОМ выполняет строго определенные протоколы: ОМ 2 реализует протокол 2-го уровня, ОМ3 – протокол 3-го уровня и т. д. до 7-го уровня модели OSI. Здесь возможна параллельная обработка пакетов на разных ОМ как одного и того же уровня, так и разных уровней. При чем количество ОМ разных уровней не обязательно должно быть одинаковым. Например, если на выполнение процессов ОМ $(n+1)$ -го уровня необходимо больше времени чем ОМ n -го уровня, то для исключения роста очереди к ОМ $(n+1)$ -го уровня в работу включается еще один (или несколько) таких ОМ.

Таким образом разделение ОМ по функциям уровней модели OSI позволяет настраивать ЦС под конкретные взаимодействующие сети путем добавления или удаления структурных элементов – масштабировать структуру ЦС.

Каждый кадр, записанный процессором КМ в общее поле памяти ЦС, генерирует задание, которое выполняется на ОМ по некоторому алгоритму, определяющим последовательность обработки, и задается управляющей информацией, содержащейся в их заголовках. Информационная связь процессов, выполняемых на ОМ представлена на рисунке 2.3.

В ЦС в реальном масштабе времени выполняется ряд процессов, реализующих функции трансляции протоколов. Каждый кадр, записанный КМ в общее поле памяти, генерирует задание, которое обрабатывается протокольными процессами по некоторому последовательному алгоритму.

Одновременно, если позволяет число ОМ может обрабатываться несколько заданий различными ОМ. Если задание, приписанное некоторому протокольному процессу, не находит свободного ОМ, то

оно ставится в очередь заданий. По освобождении ОМ, он приписывается первому ожидающему его заданию.

При выполнении процессов может возникнуть необходимость доступа к ресурсам, находящимся в общем поле памяти. Если при запросе некоторого ресурса, он оказывается занятым, то процесс ставится в очередь к данному ресурсу и его дальнейшее выполнение блокируется. ОМ, на котором выполнялась обработка задания, освобождается. При освобождении ресурса он приписывается первому ожидающему его заблокированному процессу, и обработка задания продолжается с того места алгоритма обработки, где она была прервана. Временная диаграмма развития этого процесса приведена на рисунке 2.3.

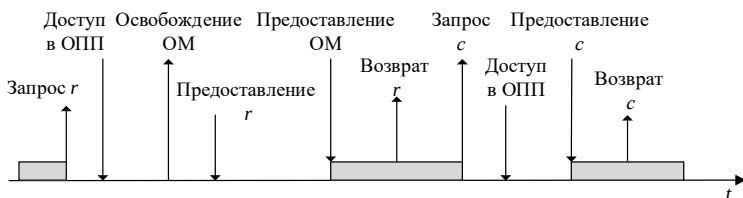


Рис. 2.3. Диаграмма развития процесса на ОМ

Передача задания от одного процесса к другому, распределение активных процессов по ОМ, регулирование доступа процессов к ресурсам осуществляется ОС. Последнее состоит из набора макрокоманд, позволяющих выполнять все необходимые системные операции. Обращение прикладного процесса к любой макрокоманде, инициирует на ОМ соответствующий системный процесс. В свою очередь, при выполнении любого системного процесса осуществляется доступ к системным ресурсам, находящимся в общем поле памяти. Если при запросе некоторого системного ресурса он оказывается занятым, то запрос ставится в очередь к системным ресурсам. При этом ОМ, выполняющий системный процесс, освобождается.

На вход ОМ поступают данные двух видов: сообщения и команды. Команды служат для управления согласованной работой уровней протоколов, а сообщения представляют собой пользовательские данные.

Согласованность в работе ОМ осуществляется командами. Обмен командами между двумя модулями происходит независимо и одновременно в обоих направлениях по инициативе стороны, передающей команду. Выдача команд производится путем ее записи в специальное поле команд. Каждой паре взаимодействующих модулей

выделено статически два поля команд, по одному в каждом направлении. В поле команд выделено специальное место для признака блокировки, наличие этого признака указывает, что там находится необработанная команда, и следующую команду туда писать нельзя.

Передача команды происходит следующим образом:

- модуль, передающий команду, читает поле команды и ждет сброса признака блокировки;
- как только признак блокировки будет сброшен, модуль записывает в поле команду, устанавливает признак блокировки и инициирует прерывание;
- по прерыванию модуль, принимающий команду, читает поле команд в свою память;
- по завершении чтения команды, модуль, принимающий команду, снимает признак блокировки.

Таким образом, модуль, передающий команду, будет ждать, пока принимающий модуль не обработает предыдущую команду. По существу, в командах передаются адреса буферов, содержащих данные. Обработка команды заключается в том, что адрес буфера переписывается в ОЗУ и ставится в кольцевую FIFO очередь на обработку в ОМ.

Обмен данными между ОМ разных уровней происходит без физической передачи данных между модулями, в командах передаются только адреса тех буферов, которые содержат данные.

2.3. Схема комплексирования

По топологии связей между модулями ЦС выделяются три схемы комплексирования (рис. 2.4):

- «матрица»;
- «общая шина»;
- «разделяемая память».

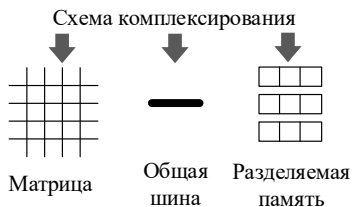


Рис. 2.4. Варианты комплексирования функциональных модулей ЦС

В ЦС с общей шиной все функциональные модули (процессоры, модули памяти, каналные модули) подсоединены к общей шине межмодульных связей, ширина которой может быть от одного бита до нескольких байтов (рис. 2.5).

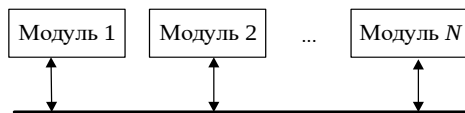


Рис. 2.5. Схема комплексирования «общая шина»

Между модулями системы нет постоянных связей, информация между ними передается в режиме разделения времени, то есть реализуется временной способ коммутации. В каждый момент времени по шине передается лишь один кадр от какого-то одного источника. Другие источники кадров должны ожидать, пока освободится шина. В каждом кадре, кроме данных, подлежащих передаче, должен содержаться адрес модуля, куда эти данные направляются. Для распознавания адресов каждый модуль имеет соответствующую аппаратуру. При такой организации межмодульных связей существует проблема конфликтов между несколькими кадрами, поступающих к одному модулю. Для разрешения конфликта предусмотрены специальные процедуры.

Отметим основные преимущества системы с общей шиной межмодульных связей:

- система характеризуется относительно низкой функциональной сложностью и невысокой стоимостью;
- в системе легко осуществляется реконфигурация структуры путем добавления или удаления функциональных модулей.

Недостатки таких систем:

- ограничение производительности системы пропускной способностью общей шины;
- ухудшение общей производительности системы при ее расширении путем добавления модулей;
- потери производительности системы, связанные с разрешением конфликтов, которые возникают в случае, когда несколько модулей одновременно претендуют на занятие общей шины для передачи данных. Эти потери можно снизить, если кроме общей памяти, доступной всем процессорам, каждый из них снабдить местной (локальной, процессорной) памятью для хранения информации, участвующей в ближайшем ряду вычислений. При наличии местной

памяти частота обращения процессоров к общей памяти уменьшается, следовательно, уменьшается частота возникновения конфликтов и потери времени на их разрешение.

- отказ общей шины приводит к выходу из строя всей системы.

Организация связей между модулями системы на основе общей шины является одним из распространенных способов построения многопроцессорных вычислительных комплексов, какими являются ЦС.

В ЦС с коммутацией на основе топологической «матрицы» реализуется метод пространственного разделения. Схема образует межмодульные связи путем установления соединений в точках пересечений прямоугольной сетки шин и таким образом представляет собой матрицу электронных переключателей, размещенных в этих точках. Каждый переключатель обеспечивает связь определенной пары модулей. С помощью «матрицы» можно устанавливать физическую связь любого процессора с любым модулем памяти или канальным модулем, а также связь любого канального модулем с любым модулем памяти или другим канальным модулем. Физический конфликт сохраняется на время передачи данных. Возможен одновременный обмен между несколькими парами модулей, при этом блоки данных могут передаваться между парой модулей по различным маршрутам.

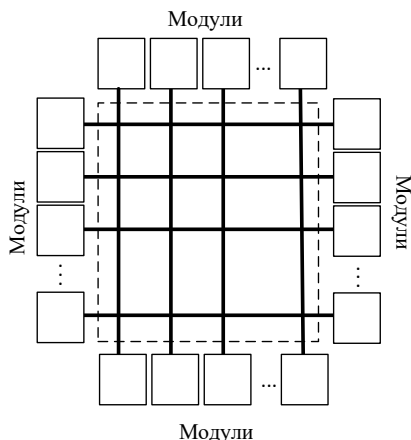


Рис. 2.6. Схема комплексирования «матрица»

Схема «матрица» не порождает конфликтов. В ЦС с этой схемой можно обеспечить высокую скорость передачи данных.

Преимущества схемы матричной схемы комплексирования модулей ЦС:

- высокая пропускная способность при передаче данных;
- возможность повышения производительности системы за счет добавления функциональных устройств (реализуется без перепрограммирования операционной системы);
- возможность повышения надежности взаимодействия модулей путем введения избыточности в структуру схемы комплексирования;
- обеспечение достаточно высокого уровня параллелизма в работе системы и коэффициента загрузки основных устройств;
- существенное уменьшение (по сравнению со схемой «общая шина») числа конфликтов в системах, так как последние возникают только при запросах от нескольких устройств на связь с одним и тем же модулем.

Недостатки матричной схемы комплексирования:

– сложность и высокая стоимость коммутатора межмодульных связей. Аппаратная сложность схемы «матрица» пропорциональна n^2 , где n – число взаимодействующих через схему модулей. В связи с этим расширение системы ограничивается возможностями схемы комплексирования;

– организация и управление процессом обработки сложнее по сравнению со схемой комплексирования «общая шина» вследствие более высокого уровня параллелизма.

В схеме комплексирования «разделяемая память» взаимные соединения модулей выполняются с помощью индивидуальных шин, подключающих каждый процессор и каждый КМ к отдельному входу. Для этого необходимо, чтобы общее поле памяти имело несколько входов, работа которых управляется операционной системой для разрешения конфликтов. Конфликт возможен в случае доступа двух и более модулей к одному и тому же разделу памяти в пределах одного временного цикла. Число подключаемых модулей ЦС к одному разделу памяти ограничивается числом входов.

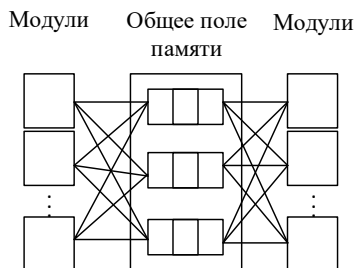


Рис. 2.7. Схема комплексирования «разделяемая память»

Преимущества схемы комплексирования «разделяемая память»:

- реализация межмодульных связей проще и обходится дешевле, чем в схеме «матрица» связей;
- скорость передачи информации значительно выше, чем в схеме «общая шина»;
- система может работать и в режиме однопроцессорной конфигурации.

Недостатки схемы комплексирования «разделяемая память» следующие:

- ограничение резервов и числа возможных конфигураций, обусловленное ограниченностью количества и типов входов в общее поле памяти;
- большое число линий связи и разъемов, усложняющих конструкцию системы и снижающих ее надежность;
- память, составленная из многовходовых модулей, является дорогостоящей.

2.4. Общее поле памяти ЦС

Центры сопряжения на базе мультипроцессорной организации используют для приема, передачи и обработки сообщений ограниченную память. Структура памяти, в которую поступают сообщения от многочисленных одновременно работающих канальных модулей, в значительной мере определяет пропускную способность ЦС.

В ЦС с большим числом портов память разбивается на секции (СОП – общей памяти) для параллельного обслуживания нескольких канальных и обрабатывающих модулей. Способы разбиения разнообразны и зависят от метода организации памяти. Различают статические и динамические методы. В первом случае за каждым каналом жестко закрепляется один или несколько буферов, в каждый из которых можно записать сообщение максимальной длины. Этот метод отличается неэффективным использованием памяти. При динамическом методе память является общим ресурсом для всех взаимодействующих с ней модулей. Поиск свободного места в памяти осуществляется лишь в момент появления запроса на него, что характеризует рациональность динамического метода. Место в памяти может быть выделено для любого канального модуля. Размер места может определяться сообщением максимальной длины или быть меньше его. В последнем случае для приема сообщения выделяется

несколько связанных в кольцевой очереди мест и память при большом разбросе длин сообщений используется эффективнее [5].

2.5. Виртуальный канал

В глобальных сетях обмен сообщениями между двумя ЦС осуществляется через один или несколько промежуточных узлов.

Если несколько процессов передачи данных хотят одновременно получить доступ к ЦС, то возможно образование нескольких логических (виртуальных) каналов, которые функционируют на базе единственной физической линии. В связи с этим одной из функций ЦС является организация и поддержание за счет разделения физической линии требуемого количества логических каналов, которые имеют возможность одновременно обмениваться данными.

Рассмотрим виртуальный канал, проходящий по пути от источника к получателю по сети с K центрами сопряжения, которые накапливают, обрабатывают и передают дальше по пути следования данные. Типичный пример показан на рис. 2.8.

В этом примере небольшой сети показаны три виртуальных канала, все исходящие из одного и того же ЦС1. Канал ВК1 проходит $K=3$ ЦС (включая исходящий), тогда как ВК2 и ВК3 проходят через ЦС2. В этих линиях инфокоммуникационной сети виртуальные каналы накладываются.

Любой виртуальный канал, может быть смоделирован последовательностью из K систем обслуживания с ожиданием (рис. 2.9). Сообщения, посылаемые пользователями виртуального канала, называются сообщениями, принадлежащими линии (или просто выделенный поток). Предполагается, что они образуют пуассоновский поток с параметром λ^B сообщений/с. Когда эти сообщения проходят по линии от узла к узлу вдоль линии следования, они в каждом буфере встречаются с так называемыми внешними сообщениями, с которыми разделяют логическую линию. Внешние сообщения, каждое из которых также следует по своему собственному логическому пути, могут поступать в рассматриваемый узел от других узлов сети или других внешних источников. Назовем поток внешних сообщений транзитным с параметром λ^T .

Разделение потоков на выделенные и транзитные при расчете характеристик сетей или центров сопряжения отдельных элементов в литературе употребляется сравнительно недавно, так как прозрачность, пропускная способность и доступность гарантируется для отдельного потока, а не для всей совокупности.

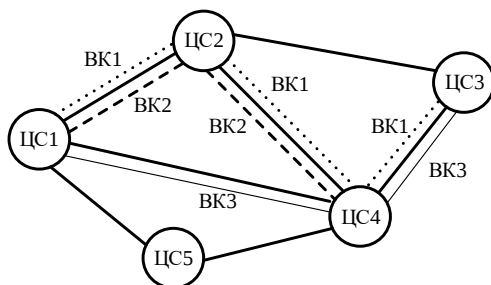


Рис. 2.8. Виртуальные каналы в сети

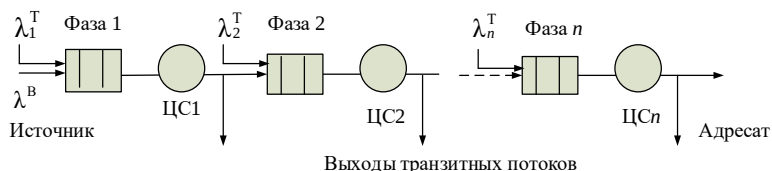


Рис. 2.9. Модель виртуального канала

Как только интенсивность потока сообщений, принадлежащих линии увеличивается, большее число сообщений скапливается в буфере вдоль линии и для внешних сообщений соответственно увеличивается время задержки. Это увеличение времени задержки свидетельствует о возникновении перегрузки. Средства управления перегрузками, которые также реализуются в ЦС, позволяют либо предотвратить увеличение выше определенной величины, либо поддерживать число сообщений, одновременно находящихся на линии, в определенных пределах. Это указывает на компромисс, который достигается введением ограничений. Так, механизм управления, реализованный в ЦС с помощью системы квитиования, ограниченного объема буферной памяти КМ в действительности уменьшает перегрузки по всей сети и поддерживает нагрузку между ЦС вдоль виртуального канала в определенных границах.

2.6. Обобщенная модульная структура МУ

Технология построения высокопроизводительных систем, какими являются ЦС, основана на модульной технологии, что позволяет масштабировать его структуру, адаптируя под необходимые условия взаимодействия разнородных сетей.

С увеличением сетевой нагрузки обеспечение необходимой производительности ЦС решается масштабированием процессорного модуля – добавлением обрабатывающих модулей. Подключение новой сети решается добавлением нового КМ. Выбор топологии связей между модулями задается схемой комплексирования. Структурно ЦС должен быть выполнен в виде стойки с отсеками для вставки модулей (рис. 2.10). Для каждого конкретного случая взаимодействия сетей подбирается достаточная (не избыточная по производительности и минимальная по стоимости) комплектация ЦС [5].



Рис. 2.10. Центр сопряжения

Согласно рассмотренной модели центра сопряжения можно выделить его общую структуру (рис. 2.11), которая синтезируется из функциональных модулей в зависимости от сложности задачи сопряжения разнородных сетей.

Модули выполняют следующие функции:

- каналный модуль накапливает блоки информации по каждому каналу и реализует функции и протоколы второго уровня модели OSI;
- обрабатывающий модуль реализует логическую обработку информации, поступающей на его вход согласно алгоритму, реализованному в нем начиная с третьего уровня модели OSI;
- схема комплексирования соединяет модули ЦС в единую систему мутации структурных элементов ЦС – выполняет функцию комплексирования модулей ЦС;

– память, разбитая на секции, предоставляет места для хранения ресурсов, необходимых для выполнения протоколов, например маршрутных таблиц, адресов, идентификаторов портов и других.

– виртуальный канал, представляющий собой цепочку промежуточных узлов от источника к адресату.

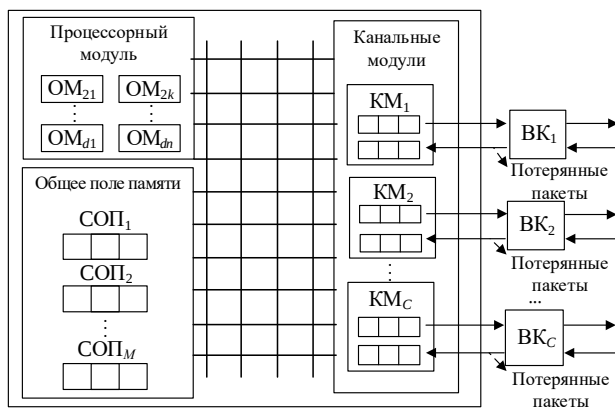


Рис. 2.11. Структурная схема ЦС

Учитывая общий вид модели ЦС структурная схема моста представлена на рис. 2.12.

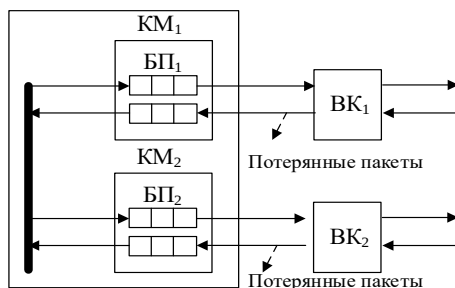


Рис. 2.12. Структурная схема моста

Структурные схемы коммутаторов с разными схемами комплексирования из набора функциональных модулей представлены соответственно на рисунках 2.13а, 2.13 б, 2.13в.

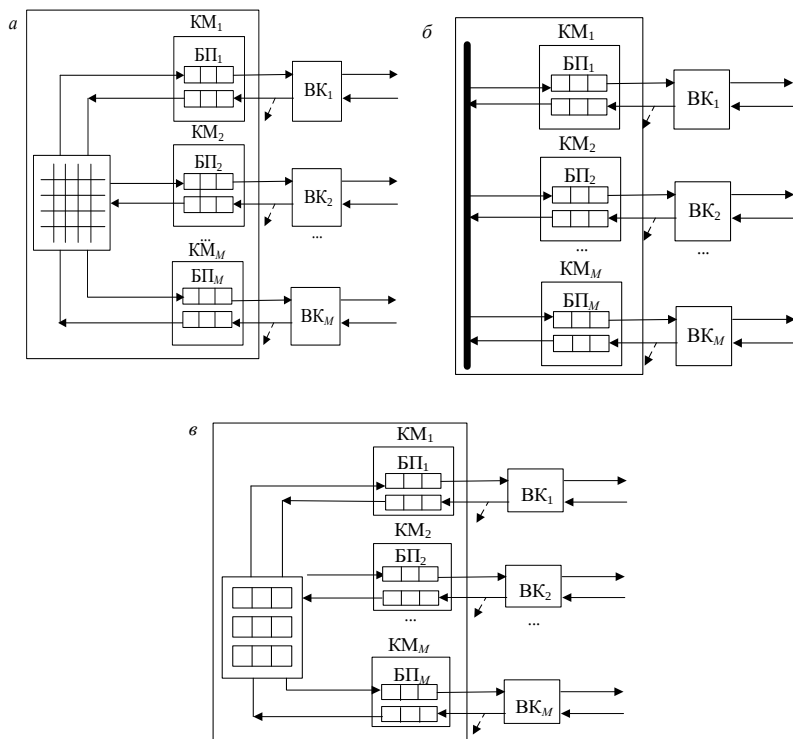


Рис. 2.13. Структурные схемы коммутаторов с топологией межмодульных связей: а – «матрица», б – общая шина, в – разделяемая память

Функций, реализованных в канальных модулях достаточно только для выполнения протоколов 2-го уровня модели OSI, что объясняет отсутствие в структурах, изображенных на рис. 2.12 и рис. 2.13 общей памяти и процессорного модуля.

Количественно мост может быть представлен множеством параметров

$$B = \{C, d, M, k\},$$

где C – число портов ввода-вывода или сопрягаемых мостом сетей, то есть $C=2$;

d – число уровней трансляции протоколов, $d=2$;

M – число СОП, $M=0$;

схема комплексирования есть «общая шина»;

k – число ОМ, определяется исходя из требований производительности моста.

Коммутатор может быть представлен множеством параметров

$$S=\{C, d, M, k\},$$

где C – число портов ввода-вывода или сопрягаемых коммутатором сетей;

d – число уровней трансляции протоколов, $d=2$ для коммутаторов канального уровня и $d=3$ для коммутатора сетевого уровня;

M – число СОП, $M=0$;

схема комплексирования «матрица» или «общая шина» или «разделяемая память»;

k – число ОМ, определяется исходя из требований производительности коммутатора.

Структурная схема маршрутизатора, соответствующая общей концепции ЦС приведена на рис. 2.11, в которой $d=3$ для маршрутизатора. Основная функция МУ данного типа – определение оптимальных маршрутов, выстраивание виртуальных каналов со сложной иерархической адресацией. Сказанное говорит о необходимости появления таблицы маршрутизации, которая является общим ресурсом для всех портов, следовательно, необходимо определить место ее хранения и обеспечить доступ к ней всех КМ.

Маршрутизатор может быть представлен множеством параметров

$$R=\{C, d, M, k\},$$

где C – число портов ввода-вывода или сопрягаемых маршрутизатором сетей;

d – число уровней трансляции протоколов, $d=3$ для маршрутизатора сетевого уровня и $d=4$ для маршрутизатора транспортного уровня;

M – число СОП, определяется исходя из требований производительности маршрутизатора;

схема комплексирования «матрица» и «разделяемая память»;

k – число ОМ, определяется исходя из требований производительности коммутатора.

Чем больше уровней трансляции реализует ЦС, тем сложнее задачи, решаемые сетевой операционной системой и сложнее программное обеспечение, реализующее протоколы высоких уровней, для выполнения которых необходимы системные ресурсы, которые так же должны быть доступны процессорам (ОМ), осуществляющим

логическую обработку по согласованию протоколов разнородных сетей, форматов пакетов, скоростей и других функций. Это объясняет появление общей памяти для хранения общих ресурсов и необходимость появления ОМ, выполняющих протоколы, не входящие в функции КМ.

Шлюз включает все составные модули общей модели ЦС, так как решает задачи взаимодействия сетей максимально возможной степени разнородности вплоть до 7-го уровня модели OSI.

Шлюз может быть представлен множеством параметров

$$G=\{C, d, M, k\},$$

где C – число портов ввода-вывода или сопрягаемых шлюзом сетей;

d – число уровней трансляции протоколов, $d=4$ для транспортного шлюза и $d=7$ для интеллектуального шлюза;

M – число СОП, определяется исходя из требований производительности шлюза;

схема комплексирования «матрица» и «разделяемая память»;

k – число ОМ, определяется исходя из требований производительности шлюза.

3. МОДЕЛИ СОГЛАСОВАНИЯ РАЗНОРОДНОСТЕЙ В ЦС

Рассмотрим некоторые модели согласования разнородностей, которые решаются средствами программного обеспечения ЦС – согласование форматов пакетов и скоростей передачи данных.

3.1. Модель фрагментации/сборки пакетов

К задачам ЦС, которую необходимо решить при обеспечении межсетевого взаимодействия, относится фрагментация/сборка пакетов. Например, если длина пакета данных, поступающих от локальной сети меньше максимальной длины IP-пакета, то IP-пакет должен быть собран в выходном порту шлюза.

Сборка (агрегация) представляет собой последовательный процесс накопления пакетов до необходимой длины. Данный процесс, увеличивает время обработки в ЦС и требует специального рассмотрения. Кроме того, агрегация тесно связана с особенностями управления входным потоком. Из-за большого несоответствия скоростей (2400 бит/с ... 56 Кбит/с для глобальных и 1...10 000 Мбит/с и более для локальных сетей) необходимо собрать пакет нужной длины и ограничить поток поступлений в ЦС до тех пор, пока собранный пакет не будет передан в исходящий канал.

Обозначим число пакетов, которые выходят из одной сети и накапливаются для сборки в один пакет для передачи в другую сеть через W ($W > 1$). Рассмотрим механизм, при котором пакет считается собранным только тогда, когда получен последний из W пакетов.

Предположим, что источник пакетов от локальной сети имеет счетчик S и представим системой обслуживания, на которую поступает пуассоновский поток с интенсивностью λ пак./с. Его показатель уменьшается на единицу, когда пакет передается на обслуживание. Первоначально счетчик устанавливается в положение W , а когда он достигает нулевого значения, указывая, что переданы W пакетов, их поступление в сеть блокируется. Модель, изображающая этот механизм, показана на рис. 3.1. Блок W представляет собой накапливающий буфер, в котором и происходит сборка пакета. Буфер накапливает до $(W - 1)$ пакетов и затем при поступлении W -го пакета вновь устанавливает счетчик S , который к этому времени достиг нуля ($j = W, i = 0$), в положение W .

В связи с добавлением в схему блока W состояние рассматриваемой сети задается парой чисел i и j , где i – значение счетчика; j – число запросов в накопителе ($0 \leq i \leq W, 0 \leq j \leq W, 0 \leq i + j \leq W$). Обозначим через $n = W - (i + j)$ число циркулирующих в сети

пакетов. Тогда может быть выписана двумерная система уравнений равновесия для вероятностей состояний p_{ij} ; с помощью этих уравнений все интересующие параметры могут быть найдены численными методами вычислений.

Однако вместо решения этой общей системы уравнений обратимся к частному случаю, когда интенсивность поступления пакетов равна пропускной способности системы обслуживания μ ($\lambda = \mu$), который подходит для нашего случая. В этом случае двумерная система уравнений состояния, включающая i и j , приводится к системе, содержащей только j . Это следует из рис. 2.1. При $\lambda = \mu$ система обслуживания C и обслуживающий прибор по сути одно и то же обслуживающее устройство. Поэтому достаточно определить стационарные вероятности p_j состояния накопителя W (или, что равносильно $p_n = p_{w-j}$, для верхней системы обслуживания). На основании рис. 2.1, при удалении системы обслуживания C соответствующая система уравнений для блока W запишется в виде

$$\begin{aligned} u(w)p_0 &= u(1)p_{w-1}, \\ u(w-1)p_1 &= u(w)p_0, \\ &\vdots \\ &\vdots \\ u(1)p_{w-1} &= u(2)p_{w-2}, \end{aligned} \quad (3.1)$$

где $u(n) = \mu n / (n + 1)$ – пропускная способность сети, равная интенсивности обслуживания μ умноженной на вероятность того, что устройство занято обслуживанием запроса.

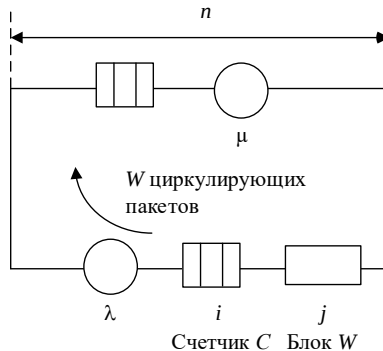


Рис. 3.1. Модель системы обслуживания сборки пакета

Решая систему уравнений для p_j , найдем

$$p_j / p_0 = u(w) / u(w - j). \quad (3.2)$$

Суммируя по всем состояниям и приравнявая сумму к единице найдем

$$p_j = \frac{(w - j) + 1}{(w - 1)(w + T_w)},$$

где параметр T_w равен сумме

$$T_w = \sum_{j=1}^w \frac{1}{j}.$$

Производительность системы обслуживания Λ вычисляется с помощью равенства (3.2). Производительность получается путем усреднения $u(n)$ в верхней системе на рис. 3.1 по всем значениям $n=(W-j)$ с учетом уравнений равновесия (3.1), откуда следует, что $u(j)p_{w-j}$ постоянно. Производительность находится в виде

$$\Lambda = \frac{\mu W}{W + T_w},$$

а среднее число пакетов $M(n)$ в системе

$$M(n) = \sum_{n=1}^w np(n) = \frac{\Lambda}{\mu} \left(\frac{1+W}{2} + 1 \right).$$

Тогда на основании формулы Литтла нормированная задержка равна

$$\mu M(T) = \left(\frac{1+W}{2} + 1 \right).$$

Таким образом, время в секундах необходимое на сборку пакета равно

$$T = W \left(\frac{1+W}{2} + 1 \right) / \mu.$$

Пусть через ЦС взаимодействуют две разнородные сети: у одной длина пакета L_1 байт, у другой L_2 байт, производительность процессора порта – μ , пакетов в секунду (пак./с). Размер накопителя $W = \lceil L_1/L_2 \rceil$, где $\lceil \cdot \rceil$ – означает операцию округления в сторону большего. По идее необходимо брать в рассмотрение только количество информационных байт пакета, а не всю длину. При определении W предполагалось, что длины служебной информации пакетов компенсируют друг друга.

В табл. 3.1 представлены некоторые результаты по времени сборки пакета для разных сетей, взаимодействующих через ЦС.

Табл. 3.1

Время сборки				
L_1 , байт	L_2 , байт	W	μ , пак./с	$E(T)$, с
128	53	3	10 000	$0.9 \cdot 10^{-3}$
256	53	5	50 000	$0.4 \cdot 10^{-3}$
1024	53	20	50 000	$4.6 \cdot 10^{-3}$

3.2. Модели согласования скорости сетей, взаимодействующих через ЦС

Даже при наилучшей возможной процедуре маршрутизации сообщения будут иногда встречать перегруженные участки и испытывать задержки на пути к месту назначения. Это может случиться из-за естественных изменений нагрузки, в результате которых она превышает средний уровень, исходя из которого проектировался алгоритм маршрутизации, из-за неожиданных сбоев и отказов элементов сети (рис. 3.2). Сказанное говорит о важности выбора механизма управления передачей, которая также реализуется в ЦС. Правая сторона ЦС должна работать как узел сети 1, а левая – как узел сети 2. ЦС принимает пакеты по входным портам со скоростью v_1 , а передает на выход уже со скоростью v_2 .

Целью применения различного рода процедур управления потоками в ЦС является предотвращение снижения эффективности функционирования сети в целом. Под снижением эффективности понимается как падение пропускной способности сети в целом или ее отдельного фрагмента, так и резкое увеличение времени доставки пакетов через сеть вследствие перегрузки.

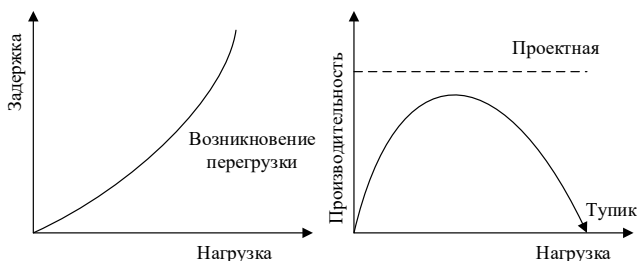


Рис. 3.2. Возникновение перегрузки в сети

В настоящее время разработаны различные методы сглаживания нагрузки, реализованные в ЦС. Одна категория методов основана на введении ограничений на интенсивность потока сообщений между каждыми двумя портами источник–адресат. В этом случае ограничение вводится непосредственно в порту-источнике. Существуют ограничения, которые вводятся для кадров, находящихся в канальных модулях ЦС. Управление, когда в портах ЦС ограничивается емкость буфера для приема кадров от сетей, взаимодействующих через ЦС, называется локальным.

Управление потоками выполняется набором механизмов, посредством которых потоки данных поддерживаются в пределах, определяемых располагаемыми ресурсами. Большинство проблем, связанных с управлением потоками, порождается недостаточной производительностью ЦС, а не перегрузкой системы пакетами данных. Последнее обычно связано с недочетами проектирования или с неустойчивой работой компонентов. Таким образом, главной функцией управления потоками в ЦС является поддержание равномерной нагрузки на различные порты ЦС, а значит на различных участках сети.

3.2.1. Согласование скоростей по технологии скользящего окна

К механизмам управления потоками, реализованным в центрах сопряжения, относится управление с помощью скользящего окна. При этом механизме управления каждый кадр при поступлении в ЦС-получатель подтверждается отдельно. Квитанция о получении данных ЦС-получателем сдвигает окно на единицу, позволяя войти в порт ЦС-источника следующему кадру, если все кадры в окне предварительно переданы. На рис. 3.3 показан пример, где размер окна равен N .

Вследствие управления с помощью скользящего окна пакеты могут передаваться по каналу до тех пор, пока их число меньше N . Если в канал уже поступило N пакетов, дальнейшая передача не разрешается, пока не поступит подтверждающий пакет, который сдвинет окно вперед на единицу.

Предположим теперь, что подтверждения передаются обратно через сеть с высшим приоритетом. При указанном предположении мы по существу пренебрегаем задержкой при возвращении подтверждений со стороны получателя обратно к источнику. Это дает возможность моделирования управления по технологии скользящего окна, накладываемого на канал передачи данных, с помощью модели замкнутой системы (рис. 3.3). Здесь ЦС-источник и ЦС-получатель связаны дополнительной искусственной системой обслуживания, обозначенной как $M + 1$, интенсивность обслуживания которой равна λ , что соответствует интенсивности входного потока в канале. По замкнутой системе циркулирует фиксированное число N пакетов. Отметим теперь, что эта модель замкнутой системы охватывает механизм скользящего окна. Если в канале (верхние M систем обслуживания) находится N пакетов, нижняя система обслуживания $M + 1$ пуста и обслуживать не может. Это моделирует состояние блокировки, которое пакеты встречают при поступлении в момент исчерпанного окна. В момент, когда один из N пакетов в канале поступает к получателю, он появляется в системе обслуживания $M + 1$ (это и есть предположение о нулевой задержке подтверждения, передаваемого обратно источнику), и источник теперь может доставить пакеты с пуассоновской интенсивностью λ . Это имеет место всегда, если в канале находится менее N пакетов; тогда остальные накапливаются в искусственной системе обслуживания $M + 1$.

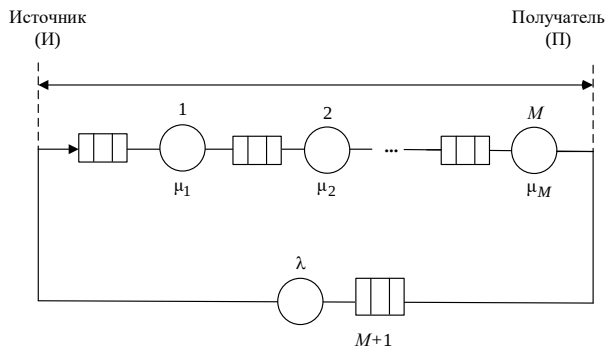


Рис. 3.3. Модель управления с помощью скользящего окна

Применим эту модель для нахождения статистики передачи пакетов по виртуальному каналу из конца в конец, а именно: найдем, какова задержка пакетов при их доставке от источника к получателю и какова производительность этого механизма управления.

Качественное представление о работе управления с помощью механизма скользящего окна можно получить без анализа. Представим, что интенсивность поступления пакетов λ увеличивается. При отсутствии управления системами обслуживания в узлах канала начинается их переполнение, задержка из конца в конец быстро возрастает, и возникает перегрузка. При наличии управления перегрузка ограничивается, поскольку одновременно в канале в состоянии транзита могут находиться не более N пакетов. Чем меньше N , тем меньше задержка из конца в конец. Однако цена, которую надо за это платить, – соответствующее снижение производительности. При увеличении N производительность возрастает, но и время задержки также увеличивается. Таким образом, налицо обмен между задержкой и производительностью, который справедлив для всех механизмов управления перегрузкой. Следовательно, наилучшей является такая схема управления, которая обеспечивает минимальную задержку при данной производительности, или, что равносильно, наибольшую производительность при заданном времени задержки. Нетрудно понять и зависимость от M , т. е. числа переприемов в процессе передачи по виртуальному каналу. При увеличении M растет также минимальная задержка из конца в конец за счет передачи, поскольку в сети накоплением и воспроизведением пакет должен быть записан в каждом центре сопряжения и затем передан. Таким образом, минимальная задержка растет линейно с ростом M и принимает абсолютное значение при $N=1$. Но производительность с ростом M уменьшается, потому что минимальное время задержки по сети увеличивается.

Пусть теперь число N допустимых пакетов в виртуальном канале увеличивается. Это приводит к увеличению как производительности, так и задержки. При не очень больших значениях N влияние на производительность оказывается более наглядным. Однако когда N продолжает расти, производительность достигает насыщения в области ее максимально возможного значения, тогда как задержка продолжает расти. Таким образом, существует оптимальное значение N в смысле получения повышенной производительности при не очень большом увеличении задержки из конца в конец.

Воспользуемся для дальнейшего анализа теоремой Нортон (разложения сетей массового обслуживания). Эта теорема устанавливает, что M последовательных систем обслуживания между точками И и П могут быть заменены одной системой обслуживания,

зависящей от состояния, без изменения статистических характеристик между точками И и П. Эта теорема дает очень ценное упрощение.

Применим теорему Нортона к исследованию механизма управления с помощью скользящего окна, описываемого замкнутой сетью массового обслуживания (рис. 3.4). Сначала упростим анализ, приняв все интенсивности обслуживания $\mu_1, \dots, \mu_M$ равными одному и тому же значению μ . Отсюда следует, что все каналы, составляющие виртуальный канал (путь от источника до получателя) имеют одну и ту же пропускную способность, каждый из них в среднем несет одну и ту же полную нагрузку.

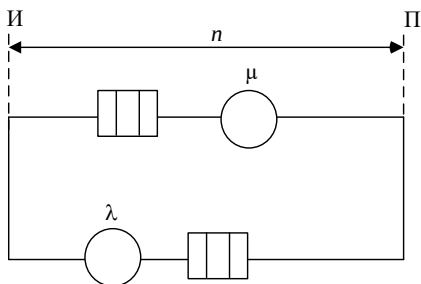


Рис. 3.4. Эквивалентная схема Нортона, циклическая СМО

Очевидно, что вероятность p_n того, что верхняя система находится в состоянии n , равна

$$p_n = \frac{p_0 \lambda^n}{\prod_{i=1}^n u(i)}, \quad (3.3)$$

где p_0 находится из обычного нормирующего условия для вероятностей

$$\sum_{i=1}^n p_n = 1.$$

Производительность $\Lambda(n)$ в случае, когда среди M показанных систем обслуживания распределены n пакетов, всегда меньше или равна μ

$$\Lambda(n) = \frac{n\mu}{n + (M - 1)}. \quad (3.4)$$

Действительно, она должна удовлетворять условию $\Lambda(n) = \mu P\{\text{система не пуста}\}$.

Поскольку все системы обслуживания одинаковы по условию симметрии, вероятность того, что система обслуживания не пуста, одинакова для всех систем. Рассмотрим в качестве примера последнюю из них, M . Вероятность того, что эта система не пуста, равна вероятности того, что, по крайней мере один из n пакетов, циркулирующих по сети, находится в этой системе обслуживания. Но n пакетов распределены среди M одинаковых систем обслуживания случайным образом, поэтому расчет интересующей вероятности (вероятности p_0 того, что система свободна) является комбинаторной задачей расчета вероятности попадания, по крайней мере, одного из n шаров в одну из M урн. В результате получим $n/[n + (M - 1)]$.

Подставляя $\Lambda(n)$ из (3.4) в (3.3), нетрудно найти искомые вероятности состояний верхней системы обслуживания на рис. 3.4:

$$p_n/p_0 = \rho^n \binom{M-1+n}{n},$$

где $\rho \equiv \lambda/\mu$ – нормированная поступающая нагрузка, а p_0 находится из равенства

$$\frac{1}{p_0} = \sum_{n=0}^N \rho^n \binom{M-1+n}{n}.$$

Производительность γ виртуального канала (ВК), управляемого с помощью окна, получается усреднением интенсивности обслуживания по всем N :

$$\gamma = \sum_{n=1}^N \Lambda(n) p_n. \quad (3.5)$$

На основании формулы Литтла задержка $t_{\text{ВК}}$ виртуального канала из конца в конец равна среднему отношению среднего числа пакетов $\bar{n}$ в ВК к γ :

$$t_{\text{ВК}} = \frac{\bar{n}}{\gamma} = \frac{\sum_{n=1}^N n p_n}{\gamma}. \quad (3.6)$$

Для нахождения рабочих характеристик управления с помощью скользящего окна можно построить по равенствам (3.5) и (3.6) графики функций нагрузки при фиксированном N или функции N при фиксированной λ . Однако вместо этого выполним анализ, позволяющий получить интересные результаты непосредственно. Чтобы увидеть, как хорошо работает управление с помощью окна и определить значение N , предлагаемое для соответствующей характеристики управления, рассмотрим область очень большой нагрузки (область перегрузки) $\lambda \geq \mu$. Сначала возьмем наихудший случай $\lambda \gg \mu$ при нагрузке, значительно превышающей пропускную способность. В частности, предположим, что на рис. 3.3 и 3.4 $\lambda \rightarrow \infty$. Тогда как только пакет поступит к получателю и к источнику вернется подтверждение, в виртуальный канал немедленно вводится другой пакет. Таким образом, в этом предельном случае канал всегда находится в состоянии N , $\bar{n} = N$, и производительность γ равна

$$\gamma = \Lambda(N) = N\mu / [N + (M - 1)], \quad \lambda \rightarrow \infty. \quad (3.7)$$

На основании формулы Литтла непосредственно получим

$$t_{\text{БК}} = N/\gamma = [M - 1 + N]/\mu, \quad \lambda \rightarrow \infty. \quad (3.8)$$

График нормированной задержки из конца в конец $\mu t_{\text{БК}}$ в зависимости от нормированной производительности γ/μ при очень большой нагрузке приведен на рис. 3.5 для типового случая с тремя участками $M=3$. Заметим из равенства (3.8), что минимальная задержка $t_{\text{БК}} = M/\mu$ имеет место при $N = 1$. Это соответствует ожиданию. Если в ВК в любой момент времени допускается только один пакет, задержек в очередях не происходит, а задержка из конца в конец равна времени передачи пакета через M последовательных участков. Это показано на рис. 3.5 при минимальной задержке $\mu t_{\text{БК}}$ для $M=3$. При увеличении размера окна N задержка растет пропорционально, как показывает равенство (3.8). В соответствии с (3.7) производительность γ сначала линейно растет с N , но затем выравнивается, приближаясь к максимальному значению μ все более и более медленно с ростом N . Объединив равенства (3.7) и (3.8), получим выражение для характеристики обмена между задержкой и производительностью

$$\mu t_{\text{БК}} = (M - 1) / [1 - (\gamma/\mu)], \quad \lambda \rightarrow \infty.$$

Равенство справедливо только для целых значений N . Оно говорит о том, что по мере приближения γ к μ (и соответственно роста N) малые изменения γ приводят к большим изменениям t_{BK} . Это свойство проявляется на рис. 3.5.

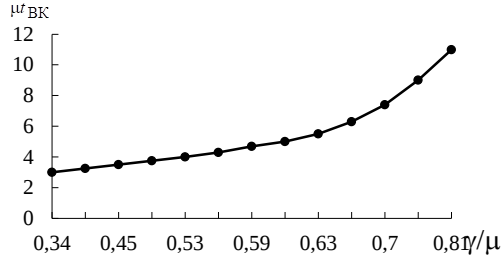


Рис. 3.5. Зависимость задержки от производительности; управление с помощью скользящего окна, $M=3, \lambda \rightarrow \infty$

Рассмотрим другой случай, когда $\lambda = \mu$. Модель имеет такой же вид, что и при $\lambda \rightarrow \infty$, за исключением того, что теперь рассматриваются $M + 1$ обслуживания, а не M . Теперь на основании равенства (3.7) производительность должна быть выражена в виде

$$\gamma = N\mu / (N + M), \lambda = \mu. \quad (3.9)$$

Новое выражение задержки для задержки находится подобным образом с тем лишь отличием, что задержка должна быть уменьшена в $M/(M + 1)$ раз с учетом того, что физический смысл имеет только задержка между И и П на рис. 3.3 (задержка в M узлах ВК). Это позволяет непосредственно записать, используя равенство (3.8)

$$\mu t_{BK} = \left[\frac{M}{M + 1} \right] (M + N), \lambda = \mu. \quad (3.10)$$

3.2.2. Согласование скоростей с помощью механизма управления темпом передачи

Регулирование темпа сеанса на уровне управления передачей применяется для управления интенсивностью прима и выдачи пакетов данных между двумя концами сеанса.

Управление темпом передачи по виртуальному маршруту работает

следующим образом. Первоначально для данного виртуального маршрута назначается фиксированное окно k . Это число устанавливается в счетчике регулирования темпа РС и записывается в локальную память приемопередатчика, с которым связан данный виртуальный канал. При каждом поступлении пакета по виртуальному маршруту показание счетчика РС уменьшается на единицу: $PC \rightarrow PC - 1$.

Когда показание счетчика становится нулевым, подача пакетов в канал задерживается на определенное время. Таким образом работу счетчика РС можно сравнить с работой счетчика, устанавливаемого при управлении с подтверждением в конце окна. Воспользуемся для обозначения размера окна буквой k , чтобы отличить этот случай от уже описанного.

В механизме управления темпом передачи первый пакет в данном окне при поступлении по назначению генерирует специальный ответный пакет (подтверждение). Этот ответ управления темпом виртуального маршрута возвращается к источнику со срочным приоритетом. При поступлении в источник он вызывает увеличение текущего показания счетчика на размер окна $PC \rightarrow PC + k$. Счетчик при стационарной работе изменяется в пределах: $0 \leq PC \leq 2k - 1$.

Покажем, что описанная работа счетчика регулирования темпа виртуального маршрута с фиксированным окном описывается моделью замкнутой сети очередей, показанной на рис. 3.6. Как и прежде, виртуальный маршрут из M участков описывается последовательным соединением M систем обслуживания. Предполагается, что пакеты поступают в источник, как только они возникают в пункте назначения. Введем блок k , накапливающий $(k - 1)$ пакетов. При поступлении k -го пакета, он разгружает блок k , и система обслуживания, моделирующая счетчик РС, мгновенно получает k пакетов. Таким образом, в этой модели будет происходить процесс групповых поступлений.

По модели рис. 3.6 заметим, что когда система обслуживания, моделирующая счетчик РС, пуста, то в фиктивном блоке k уже доставлены и ожидают $j \leq (k-1)$ пакетов, тогда как $(2k-1)-j$ пакетов находятся в процессе транзита по ходу виртуального маршрута. Устройство генерации пакетов λ работать не может, и пакеты от поступления в виртуальный канал блокируются. Если же $PC > 0$, пакеты покидают виртуальный канал (систему обслуживания) в среднем с интенсивностью λ пакетов в единицу времени, причем их поток описывается тем же пуассоновским законом распределения, как и предполагалось в предыдущих моделях.

Предполагается, что виртуальный маршрут из M участков также может быть описан эквивалентной схемой Нортонa с обслуживанием, зависящим от состояния, что приводит к модели на рис. 3.7.

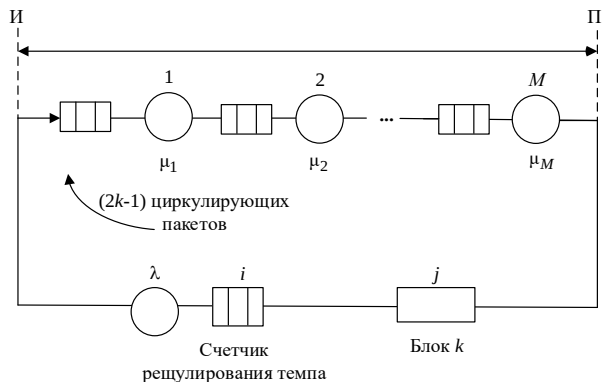


Рис. 3.6. Модель регулирования темпа передачи по виртуальному каналу

Показанные на рис. 3.7 два числа i и j относятся, соответственно, к состояниям счетчика РС и блока k . Тогда можно записать

$$0 \leq i \leq 2k-1, 0 \leq j \leq k-1, 0 \leq i+j \leq 2k-1.$$

Состояние n , описывающее число пакетов, перемещающихся по ВК из M участков, в точности равно $n = (2k-1) - (i+j)$. Как и раньше, при заданной совместной статистике i и j необходимо упростить статистику n , а по ней – требуемые характеристики управляемого виртуального маршрута.

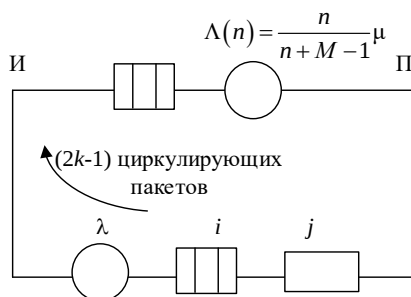


Рис. 3.7. СеМО, моделирующая регулирование темпа передачи по ВК

Следуя процедуре, которая принималась при описании механизма скользящего окна, рассмотрим сначала случай большой нагрузки $\lambda \rightarrow \infty$. Из рис. 3.6 и рис. 3.7 следует, что в этом случае большинство

поступающих пакетов отклоняется ($PC \rightarrow 0$): когда бы блок k не сработал, в источнике всегда имеется достаточно новых пакетов, чтобы направить в виртуальный маршрут k таких пакетов. Число пакетов в виртуальном маршруте колеблется между k и $2k - 1$, как показано на диаграмме состояний (рис. 3.8), двумерная задача сводится к одномерной. Заметим, поскольку блок k на рис. 3.7 не может сработать в промежутке $0 \leq j \leq k - 1$, уравнения, связывающие вероятности состояний в частном случае $\lambda \rightarrow \infty$, запишется в виде

$$\begin{aligned} u(2k-1)p_0 &= u(k)p_{k-1}, \\ u(2k-2)p_1 &= u(2k-1)p_0, \\ &\dots \\ u(k)p_{k-1} &= u(k+1)p_{k-2}. \end{aligned} \quad (3.11)$$

Из (3.11) искомая вероятность состояния p_j , найдется в виде

$$p_j / p_0 = u(2k-1) / u(2k-1-j).$$

Определим p_0 с помощью обычного нормирующего условия

$$\sum_{j=0}^{k-1} p_j = 1,$$

и воспользовавшись выражением для пропускной способности эквивалентной схемы Нортона $\Lambda(n) = \frac{n\mu}{n + (M-1)}$, найдем

$$p_j = \frac{(2k-1-j) + (M-1)}{(2k-1-j)(k + (M-1)S_k)},$$

где параметр S_k равен сумме

$$S_k = \sum_{j=k}^{2h-1} j^{-1}.$$

Пользуясь такой же процедурой, какая применялась в предшествующем анализе, найдем нормированную производительность виртуального маршрута с M участками

$$\frac{\gamma}{\mu} = \frac{k}{k + (M-1)S_k}, \lambda \rightarrow \infty, \quad (3.12)$$

и нормированную задержку из конца в конец равна

$$\mu t_{BK} = M - 1 + \left(\frac{2k - 1 + k}{2} \right), \lambda \rightarrow \infty. \quad (3.13)$$

Сравнивая (3.12) и (3.13) с равенствами (3.9) и (3.10) для управления со скользящим окном, можно видеть, что эти два механизма описываются одинаковыми уравнениями в случае большой нагрузки.

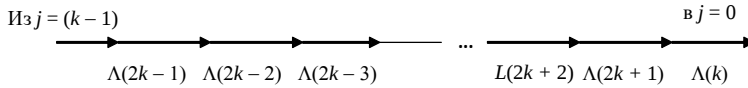


Рис. 3.8. Диаграмма состояний регулирования темпа передачи по ВК

Равенства, описывающие характеристики двух стратегий управления, могут быть записаны в следующей объединенной форме:

1. $\lambda \rightarrow \infty$ (случай большой нагрузки)

$$\mu E(T) = M - 1 + N',$$

$$\frac{\gamma}{\mu} = \left(1 + \frac{M-1}{k} \right)^{-1},$$

где $N' = N$ – для механизма скользящего окна;

$N' = (3k - 1)/2$ – для механизма управлением темпа передачи.

2. Случай $\lambda = \mu$:

$$\mu t_{BK} = \left(\frac{M}{M+1} \right) (M + N'),$$

$$\frac{\gamma}{\mu} = \left(1 + \frac{M}{k} \right)^{-1},$$

где $k = N$ – для механизма скользящего окна;

$k = S_k$ – для механизма управлением темпа передачи.

Сравнение двух схем управления потоками приведено на рис. 3.8.

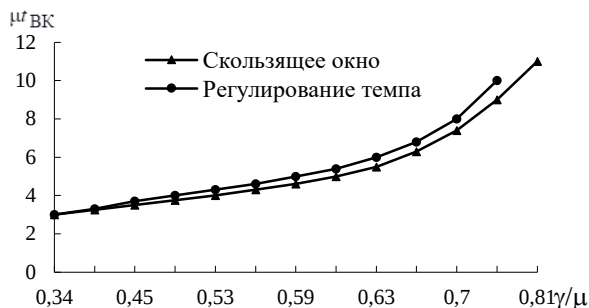


Рис. 3.8. Сравнение механизмов управления с подтверждением в конце окна, со скользящим окном и регулирования темпа передачи

Из сравнительных результатов видно, что механизм управления с помощью скользящего окна наиболее эффективен.

3.2.3 Согласование скоростей с помощью механизма «дырявое ведро»

Представим себе ведро с маленькой дырочкой в днище. Независимо от скорости, с которой вода наливается в ведро, выходной поток обладает постоянной скоростью, когда в ведре есть вода, и нулевой скоростью, когда ведро пустое. Кроме того, когда ведро наполняется, вся лишняя вода выливается через край и теряется (т. е. не попадает в выходной поток под дырочкой).

Та же идея применима к пакетам, как показано на рис. 3.9.

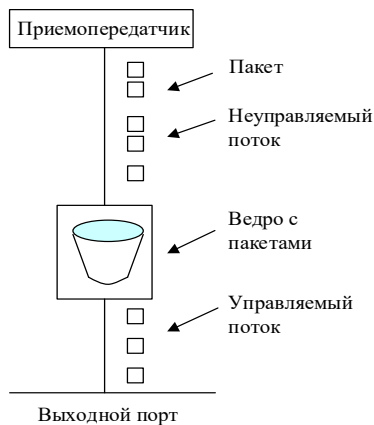


Рис. 3.9. Дырявое ведро с пакетами, реализованное в ЦС

Принцип действия механизма дырявого ведра таков: в каждом порту ЦС есть интерфейс, содержащий механизм «дырявое ведро», то есть конечную внутреннюю очередь порта. Если пакет появляется в очереди, когда очередь полная, пакет игнорируется. Другими словами, если несколько источников пытаются послать пакеты, когда в очереди уже стоит максимально допустимое их число, новый пакет игнорируется. Такой интерфейс может быть реализован как аппаратно, так и программно операционной системой ЦС. Он был предложен Тернером в 1986 г., и называется алгоритмом дырявого ведра. По сути, это не что иное, как однолинейная система массового обслуживания с постоянным временем обслуживания. Этот механизм преобразует неравномерный поток пакетов от разных источников данных, подключенных к шлюзу в равномерный поток пакетов в сети, сглаживая пики и значительно снижая вероятность перегрузки.

Максимальное число обрабатываемых пакетов может быть определено следующим образом. Если I – количество пакетов, поступающих в один момент времени, а T – время в числе интервалов между двумя поступлениями, то к моменту следующего поступления в буфере окажется $(I - T)$ пакетов (предполагаем, что в один интервал времени убывает один пакет).

Для уменьшения числа потерянных пакетов введен буфер ограниченного объема, куда будут помещаться пакеты в случае превышения порогового значения счетчика. Эти ячейки будут повторно поставлены на передачу, когда нагрузка на входы ЦС уменьшится. Тогда потеря заявок будет происходить только в случае переполнения буфера.

Одноканальная экспоненциальная СМО задается параметрами λ , μ . Цель нашего анализа заключается в расчете $\mu t_{вк}$ для сравнения с представленными ранее механизмами согласования скоростей.

Коэффициент загрузки рассчитывается по формуле $\rho = \lambda / \mu$.

Если выполняется условие

$$\rho \leq 1, \quad (3.13)$$

то существует стационарный режим функционирования СМО. В стационарном режиме все вероятностные характеристики системы – постоянные во времени величины. Сами происходящие в СМО события остаются при этом случайными.

В стационарном режиме среднее число N заявок в СМО постоянно. Поэтому среднее число заявок, приходящих в СМО в единицу времени, равно среднему числу заявок, в единицу времени из

СМО уходящих. Следовательно, в стационарном режиме интенсивность потока уходящих заявок равна λ .

Среднее число заявок в очереди в одноканальной экспоненциальной СМО рассчитывается по формуле $L = \frac{\rho^2}{1-\rho}$.

Среднее число N заявок в СМО равно сумме среднего числа L заявок в очереди и среднего числа ρ заявок в канале: $N = \frac{\rho}{1-\rho}$.

Заявка перемещается в очереди в среднем с постоянной скоростью. Среднее число переходов заявки в очереди на одно место вперед за единицу времени равно λ .

При такой скорости перемещения L переходов произойдет за время, равное в среднем

$$\bar{T}_{\text{ож}} = \frac{\rho}{\mu(1-\rho)}. \quad (3.14)$$

Формула (3.14) дает среднее время ожидания.

Среднее время задержки заявки в СМО есть сумма среднего времени ожидания и среднего времени обслуживания заявки: $t_{\text{вк}} = \frac{1}{\mu(1-\rho)}$.

Отсюда нормированная задержка равна $\mu t_{\text{вк}} = \frac{1}{(1-\rho)}$.

Вероятность наличия в системе k требований определяется с помощью геометрического закона распределения в виде $(1-\rho)r^k$, $k = 0, 1, 2, \dots$.

Сравнение всех представленных механизмов приведено на рис. 3.10.

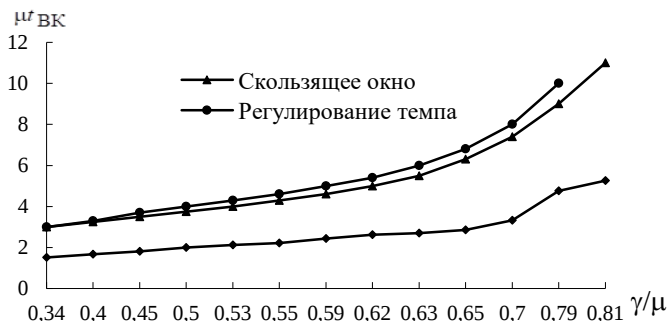


Рис. 3.10. Сравнение механизмов согласования скоростей в ЦС

4. МОДЕЛИ РАСЧЕТА ВЕРОЯТНОСТНО-ВРЕМЕННЫХ ХАРАКТЕРИСТИК ЦЕНТРА СОПРЯЖЕНИЯ

4.1. Вероятностно-временные характеристики ЦС

Анализ пользовательских требований является первым важным шагом при проектировании инфокоммуникационных сетей. На нем, в частности, оцениваются характеристики трафика, обеспечение которых задают требования к качеству обслуживания. Таким образом, сеть гарантирует пользовательскому трафику требуемые параметры обслуживания, а сетевые узлы, от которых это зависит должны обладать необходимой **производительностью** [4].

Основными характеристиками ЦС, измеряющими его производительность, являются:

- Скорость фильтрации (filtering);
- Скорость маршрутизации (forwarding);
- Пропускная способность (throughput);
- Время коммутации кадра.

Кроме того, существует несколько характеристик ЦС, которые в наибольшей степени влияют на указанные характеристики производительности. К ним относятся:

- Размер буфера (буферов) кадров;
- Производительность внутренней коммутационной схемы;
- производительность процессоров;
- размер внутренней адресной таблицы.

1) Скорость фильтрации и продвижения кадров – это две основные характеристики производительности ЦС.

Данные характеристики являются интегральными показателями, они не зависят от того, каким образом технически реализовано ЦС. Скорость фильтрации определяет скорость, с которой ЦС выполняет следующие этапы обработки кадров:

- прием кадра в свой буфер,
- просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра,
- уничтожение кадра, так как его порт назначения совпадает с портом-источником.

Скорость продвижения определяет скорость, с которой ЦС выполняет следующие этапы обработки кадров:

- прием кадра в свой буфер,

- просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра,
- передача кадра в сеть через найденный по адресной таблице порт назначения.

Как скорость фильтрации, так и скорость продвижения измеряются в кадрах в секунду.

Поскольку скорость фильтрации и продвижения не зависят от технической реализации ЦС, то оценивать пригодность устройств сопряжения корпоративной сети будем по следующим, ниже лежащим показателям:

2) Пропускная способность ЦС, измеряется количеством переданных в единицу времени через его порты данных и оценивается в виде:

$$\Lambda = \lim_{\tau \rightarrow \infty} \frac{N(\tau)}{\tau} \text{ пакетов/с,}$$

где $N(\tau)$ – число пакетов, переданных через ЦС за интервал времени $(0, \tau)$.

3) Задержка передачи кадра, измеряется как время, прошедшее с момента прихода первого байта кадра во входной порт ЦС до момента появления этого байта на выходном порту ЦС. Задержка складывается из времени, затрачиваемого на буферизацию байт кадра, а также времени, затрачиваемого на обработку кадра в ЦС, просмотра адресной таблицы (если это необходимо), принятие решения о фильтрации или продвижении и получения доступа к выходному порту.

Среднее время задержки пакета: $\bar{t} = M(\tilde{t})$, с, где $\tilde{t}$ – задержка кадра в ЦС, случайная величина; $M(\tilde{t})$ – математическое ожидание случайной величины $\tilde{t}$.

Центр сопряжения – это как правило многопортовое многопроцессорное устройство, поэтому для него принято все приведенные выше характеристики (кроме задержки передачи кадра) давать в двух вариантах. Первый вариант – суммарная производительность ЦС при одновременной передаче трафика по всем его портам, второй вариант – производительность, приведенная в расчете на один порт.

4) Вероятность потери кадров, поступающих в s -й каналный модуль (процент потерянных пакетов)

$$P_{\text{пот}S} = \lim_{\tau \rightarrow \infty} \frac{N_{\text{пот}S}(\tau)}{N_S(\tau)},$$

где $N_{\text{пот}S}(\tau)$ – число отвергнутых кадров s -го входящего канала за промежуток времени τ ;

$N_S(\tau)$ – общее число кадров, поступивших в ЦС по s -му входящему каналу за промежуток времени τ , $s = \overline{1, S}$.

S – общее число дуплексных каналов связи, подключенных к ЦС.

Существуют несложные расчеты, которые могут дать представление о том, какие значения показателей качества обслуживания обеспечивает ЦС в реальной ситуации. Эти расчеты не учитывают коллизий, возникающих между кадрами, передаваемыми по разным маршрутам коммутационной схемы, потерь на время ожидания доступа к среде и других явлений, которые обусловлены случайными факторами, значительно снижающими реальную производительность ЦС.

Оценка названных характеристики и их оценка требуют соответствующего модельного обеспечения, которое рассматривается ниже.

4.2. Математическое обеспечение расчета ВВХ ЦС

Математической базой методологии структурно-функционального анализа мультипроцессорных систем в частности стали сети массового обслуживания – СеМО. Аппарат экспоненциальных СеМО позволяет получить оценивание производительности и времен задержки в аналитическом виде. Наиболее разработана теория экспоненциальных СеМО. Разработаны практические формы расчета ВВХ экспоненциальных СеМО. Экспоненциальность, однако, довольно сильное ограничение. Поэтому для анализа СеМО используют и алгоритмические методы, реализуемые с помощью имитационных моделей.

Модели для расчета ВВХ предназначены для оценивания наиболее значимых показателей качества обслуживания пользователей и системных характеристик: надежности, живучести, производительности, времени транспортировки пакетов; вероятности потери сообщений и блокировки в узлах; области допустимых значений нагрузки, при которых обеспечивается требуемое качество обслуживания.

В СеМО отдельные СМО отображают функционально

самостоятельные части ЦС, связи между СМО – структуру ЦС, а требования, циркулирующие в СеМО – сообщения или задания в мультипроцессорных системах. Поскольку коммутационная система имеет сетевую структуру, то сказанное в полной мере относится к ее формализованному описанию.

Особенности применения теории СеМО к моделированию сетей вызывают необходимость учитывать в этой теории различные виды взаимодействия отдельных СМО, являющихся узлами СеМО. Взаимодействие узлов является следствием того, что каждое требование подлежит обслуживанию на нескольких узлах. Основной вид взаимодействия узлов – последовательное поэтапное обслуживание требования в разных узлах. Другие виды взаимодействия могут заключаться в зависимости от длительностей пребывания на последовательных этапах, в одновременном занятии требованием на некотором этапе двух или нескольких узлов сети, в блокировках узла и т.д.

Для оценивания характеристик ЦС используем аппарат замкнутых экспоненциальных СеМО, основанный на методе анализа средних значений. Алгоритм анализа средних значений удобен тем, что не требует усилий для борьбы с переполнением и (или) обнулением результатов в памяти ЭВМ.

Как показано во второй главе в Цс имеется конечное число буферов, расположенных в КМ для хранения пакетов, следовательно, циркулирует конечное число заданий, не превышающее

$$b = \sum_{s=1}^S b_s$$

Кроме того, всегда имеется конечное число процессов $A = \sum_{x=1}^X N_x$,

которое соответствует количеству выполняемых параллельно процессов взаимодействия структурных элементов между собой.

Расчет производительности ЦС, работающих по принципу коммутации пакетов могут быть описаны замкнутыми СеМО, в которых циркулирует конечное число заявок (пакетов).

Узел i в замкнутой СеМО представляет собой обслуживающий прибор (возможно многоканальный) с экспоненциальным распределением времени обслуживания и очередью заявок (пакетов), ожидающих обслуживания. После обработки пакета в узле i , он с вероятностью P_{ij} $\left(\sum_{n=1}^N P_{ij} = 1 \right)$, где N – общее число узлов в СеМО)

переходит на обслуживание в другой узел j . Общее число заявок по всей сети постоянно и равно M .

Решающим в анализе средних величин является вычисление среднего времени задержки в узле i , входящего в СеМО. Рассмотрим момент, когда пакет поступает в эту систему. Средняя задержка пакета состоит из времени его обслуживания τ_k и времени обслуживания пакетов, ожидающих перед ним в очереди, включая пакет, находящийся на обслуживании.

Теорема о входящем потоке для замкнутых сетей массового обслуживания показательного типа устанавливает, что состояние узла в момент поступления пакета описывается распределением вероятностей, равным распределению для стационарного состояния сети с числом циркулирующих в ней пакетов, меньшим на один пакет. Эта теорема дает возможность измерить среднее число пакетов, наблюдаемых в момент поступления. Среднее время задержки в k -ом узле при наличии в сети j пакетов связано со средним числом ожидающих при наличии в сети $(j-1)$ пакетов:

$$t_k(j) = \tau_k(j) \left(1 + \frac{n_k(j-1)}{\alpha_k} \right),$$

где $n_k(j-1)$ – среднее число пакетов, ожидающих в узле k при наличии в сети $(j-1)$ пакетов;

α_k – число обслуживающих приборов в k -м узле.

Данное равенство имеет рекуррентную форму. Связь между $\tau_k(j)$ и $n_k(j)$, необходимая для завершения алгоритма, устанавливается на основании формулы Литтла

$$\Lambda(j) = \frac{j}{\bar{t}(j)}.$$

Полная рекуррентная процедура для циклической сети теперь при $j = \overline{1, M}$ и $k = \overline{1, N}$ рассчитывается в виде

$$t_k(j) = \tau_k(j) \left(1 + \frac{n_k(j-1)}{\alpha_k} \right) \quad (4.1)$$

$$\bar{t}(j) = eT(j); \quad (4.2)$$

$$\Lambda(j) = \frac{j}{\bar{t}(j)}; \quad (4.3)$$

$$n_k(j) = \Lambda(j) e_k t_k(j) \quad (4.4)$$

где $t_k(j)$ – среднее время обслуживания пакета в k -ом узле при наличии в сети j пакетов,

$\bar{t}(j)$ – среднее время пребывания пакета в ЦС от момента поступления во входной порт до момента выхода из исходящего порта, при наличии в сети j заявок;

$\Lambda(j)$ – пропускная способность замкнутой СеМО при наличии в ней j заявок;

вектор $e = [e_k]_{k=1, \overline{N}}$ является решением системы линейных уравнений

$$e = eP,$$

которая определяет стационарное распределение цепи Маркова, управляющей переходами заявок в замкнутой СеМО с матрицей вероятностей переходов

$$P = [P_{ij}]_{k=1, \overline{N}; j=1, \overline{N}},$$

где P_{ij} были определены выше. Система (4.10) решается при дополнительном ограничении

$$\sum_{k=1}^N e_k = 1$$

Решение процедуры (4.1) – (4.4) начинается с $n_i(0)=0$, для $k = \overline{1, N}$.

Модели СеМО допускают также и имитационные методы анализа.

Термин "имитационная модель" обозначает математическую модель, которая специально предназначена для исследования в режиме имитации, т.е. для сравнения проектов, планов, конструкций, управлений путем вариантных просчетов. Определение "имитационная" в словосочетании "имитационная модель" характеризует не саму модель, а способ ее использования. Имитационные модели позволяют воспроизводить в некотором масштабе времени процесс функционирования исследуемой системы с

той степенью детализации, которая необходима разработчику, причем элементарные явления, составляющие процесс, имитируются с сохранением их логической структуры и последовательности протекания во времени, т.е. обладает хорошей "физической" наглядностью.

Имитационное моделирование помимо характеристик среднего времени ожидания, пребывания, обработки позволяет получить функции распределения вероятностных и временных характеристик процесса сопряжения разнородных сетей.

Например, все модели механизмов согласования скоростей в ЦС могут быть реализованы в системе мультиагентного моделирования AnyLogic University. Рассмотрим особенности их реализации.

Перед разработкой любой имитационной модели определяются структурные элементы, из которых она состоит. В самом общем виде структуру модели математически можно представить в виде

$$E = f(x_i, y_j),$$

где E – результат действия модели;

x_i – переменные и параметры, которыми может управлять пользователь;

y_j – переменные и параметры, которыми управлять невозможно;

f – функция зависимости между x_i и y_j , определяющая величину E .

Такое упрощение полезно тем, что оно показывает зависимость функционирования системы как от контролируемых параметров, так и от неконтролируемых переменных. Имитационная модель представляет собой комбинацию составляющих: компонентов, переменных, параметров, спецификаций, ограничений, целевых функций.

Под компонентами понимаются составные части (объекты или классы объектов), которые при соответствующем объединении образуют систему.

Параметры – это величины, которые оператор, работающий с моделью, может выбрать произвольно, в отличие от переменных, которые могут принимать только значения, определяемые видом данной функции. Можно сказать, что параметры, после того как они установлены, являются постоянными величинами, не подлежащими изменению.

Спецификации описывают возможные варианты или режимы функционирования системы, которые можно реализовать в моделях и на основе сравнения выбрать подходящий вариант с точки зрения

целевой функции.

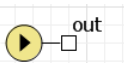
Ограничения представляют собой устанавливаемые пределы изменения значений переменных или ограничивающие условия распределения тех или иных средств. Они вводятся либо разработчиком (искусственные ограничения), либо самой системой вследствие присущих ей свойств (естественные ограничения).

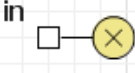
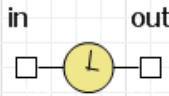
В пособии для оценивания характеристик ЦС использован аналитический аппарат экспоненциальных СеМО, а для оценивания характеристик коммутатора на основе двоичной матрицы и функций согласования скоростей в ЦС – имитационный метод.

Компоненты имитационных моделей приведены в табл. 4.1.

Табл. 4.1

Компоненты имитационной модели

Компонент	Назначение	Параметры компонента
Агент	Пакет	– Время генерации – Время выхода из маршрута – Номер пакета
Source  out – выходной порт	Создает заявки. Обычно используется в качестве начальной точки потока заявок	Параметры: – тип агентов – пакет; – интенсивность прибытия (метод <code>set_rate()</code>). – время между прибытиями, – расписание или табличная функции. Функции: – <code>void inject(int n)</code> – создает заданное количество агентов, равное <code>n</code>). – <code>long count()</code> – возвращает количество агентов, созданных объектом; – <code>long countArrivals()</code> – возвращает количество прибытий, произошедших в данном блоке Source
Sink	Уничтожает поступившие заявки.	Параметры: – тип агентов – пакет. Функции:

 in – входной порт	Используется в качестве конечной точки потока заявок	– long count() – возвращает количество агентов, уничтоженных объектом Sink
Delay  in – входной порт; out – выходной порт	Задерживает заявки на заданный период времени. Время задержки вычисляется динамически, может быть случайным, зависит от текущей заявки или от каких-то других условий.	Параметры: – тип агентов – пакет; – время задержки; – способ задания задержки (определенное время, метод stopDelay()); – вместимость. Переменные: – StatisticsContinuous statsUtilization – статистика использования объекта, высчитывается как size()/capacity). Функции: – void suspend(double dt) – приостанавливает задержку на заданный таймаут dt; – void suspend() – приостанавливает блок Delay; – void resume() – возобновляет приостановленный ранее объект Delay
Queue  in – входной порт; out – выходной порт; outTimeout – порт для	Моделирует очередь заявок, ожидающих обработки	Параметры: – тип агентов – пакет. – вместимость – int capacity. – порядок поступления в очередь (FIFO, LIFO, По приоритету) Функции: – int size() – возвращает количество агентов, находящихся в данный

агентов, покидающих объект по таймауту; outPreempted – порт для вытеснения агентов		момент в очереди; Переменные: – StatisticsContinuous statsSize – статистика длины очереди.
---	--	--

Все множество входных и выходных данных разбиты на группы по назначению: описывающие структуру маршрута передачи, характеристики узлов коммутации, поступающую нагрузку, ограничения на длину окна/темпа передачи/ведра и количество экспериментов.

В табл. 4.2 приведены все группы входных и выходных данных и описывающие их параметры в имитационных моделях ЦС.

Таблица 3.4

Описание входных и выходных данных имитационной модели

Группа параметров	Параметры
Описывающие структуру маршрута от источника к адресату	Число ЦС. Ограничение очереди в ЦС.
Описывающие характеристики шлюза	Пропускная способность ЦС. Функция распределения времени, затрачиваемого шлюзом на обработку одного пакета
Описывающие рабочую нагрузку	Интенсивность поступления пакетов в шлюз. Функция распределения времени поступления пакетов в ЦС. Маршрутная таблица
Описывающие эксперимент	Количество экспериментов. Время, отводимое на эксперимент.
Характеристики времени передачи пакетов	Среднее время задержки. Дисперсия задержки. Гистограмма-функция распределения задержки
Характеристики очередей в шлюзе	Средняя длина очереди. Максимальное значение очереди. Количество потерянных пакетов

Все выходные параметры накапливаются в виде статистик. Для каждого пакета определяется время прохождения по маршруту как разница между моментом времени генерации пакета (появления в программе) и моментом ухода пакета из модели. По истечении времени моделирования накопленная статистика позволяет оценить среднюю задержку пакетов, дисперсию задержки и построить гистограмму распределения задержки. То же самое касается и отдельных статистик по каждому центру сопряжения и очередей.

4.3. Модель коммутатора со схемой комплексирования «матрица»

Схема коммутации на основе «матрицы» представляет собой регулярную решетку, составленную из однотипных двоичных коммутационных элементов (КЭ). Каждый КЭ имеет по два входа и по два выхода и может находиться в одном из двух состояний [1]:

- 1) передача пакета с верхнего (нижнего) входа КЭ на верхний (нижний) выход КЭ – «транзит»;
- 2) передача протокола с верхнего (нижнего) входа КЭ на нижний (верхний) выход КЭ – «кросс».

Процесс передачи кадров от входного порта на выходной состоит в следующем. В заголовке каждого кадра находится маршрутное поле S , представляющее собой последовательность двоичных разрядов, число которых равно числу каскадов в КС. В каждом каскаде происходит декодирование соответствующего разряда маршрутного поля. Если разряд равен 1, то КЭ реализует операцию «кросс», в противном случае (разряд равен 0) – операцию «транзит».

Число каскадов в двоичных КС зависит от числа входов в нее. При числе входов N для обеспечения полнодоступности, то есть наличия хотя бы одного пути между каждым входом, необходимо иметь число каскадов $n = \log N$.

В случае, когда на вход КЭ одновременно поступают два пакета, которые должны быть переданы на один и тот же выход КЭ, возникает конфликт пакетов с возможностью потери одного или обоих пакетов.

Перед тем как пакет будет передан от КЭ i -го каскада к КЭ $(i+1)$ -го каскада, необходимо убедиться, что он может быть принят. Отказ в приеме может возникнуть из-за занятости КЭ. Прежде чем передать данные, необходимо установить сквозное соединение через «матрицу» между входящим и исходящим портами.

Алгоритм коммутации:

1. Прием кадра KM_i , $i = \overline{1, S}$;

2. По адресу назначения и маршрутной таблице (МТ) назначается ТЕГ;

3. Выделенный из кадра пакет сохраняется в порту КМ_i;

4. Первый бит ТЕГа поступает на вход КЭ, непосредственно соединенного с выходом приемного контроллера КМ_i;

5. Происходит переключение КЭ, если соответствующий биты ТЕГа выход не занят;

6. Если соответствующий биты ТЕГа выход КЭ свободен, то он резервируется за прокладываемым соединением, и выставляется соответствующее значение флага. По значению флага поступающий ТЭГ определяет состояние КЭ: свободен требуемый выход или занят;

7. При свободном выходе КЭ i -го каскада передает остаток ТЕГа КЭ $(i+1)$ -го каскада, непосредственно вход которого соединен с выходом КЭ i -го каскада. По первому биту остатка ТЕГа определяется, свободен ли соответствующий выход КЭ $(i+1)$ -го каскада. Если свободен, он резервируется за данным требованием (ТЕГом), если выход занят, то КЭ $(i+1)$ -го каскада выдает КЭ i -го каскада отрицательную квитанцию. Аналогично происходит взаимодействие и между всеми другими непосредственно соединенными парами КЭ каждого маршрута при прокладке соединения.

По отрицательной квитанции осуществляется сброс в «обратной волне» во всех КЭ, в которых были зарезервированы выходы.

8. Если сквозное соединение проложено, то, если в порту выходного КМ есть свободные места, то формируется и передается «обратной волной» положительная квитанция, иначе – отрицательная квитанция и сброс во всех КЭ проложенного соединения;

9. По получении положительной квитанции входящий КМ транслирует через установленное соединение пакет;

10. По окончании приема пакета (перезаписи пакета в порт исходящего КМ), сигналом, аналогичным отрицательной квитанции, осуществляется снятие сквозного соединения (освобождение КЭ маршрута).

Введем временные соотношения, характеризующие работу коммутатора на основе «матрицы». В общем случае, время, затрачиваемое на установление соединения между входящим и исходящим портами, является случайной величиной. Случайность обусловлена возможными коллизиями, что приводит к одной или нескольким попыткам установки соединения, причем, каждая попытка может отличаться количеством переключенных КЭ.

Повторные попытки установления соединения придают стохастичность этому процессу, т.к. возникают вынужденные

возвращения на предыдущие КЭ и число каскадов, пройденных вызовом при его доставке адресату, оказывается случайным числом. Учет данного аспекта позволяет прибегнуть к имитационному моделированию для установления логического канала между входящим и исходящим портами ЦС.

Таким образом, время установления соединения определим как случайную величину, которая может быть найдена выражением

$$t_{y.c} = \sum_{i=1}^{n_k} t_{\kappa_i} + \sum_{i=1}^{n_o} t_{o_i} + n_n t_n, \quad (4.5)$$

где n_k – число каскадов, построенного от источника к адресату;

n_o – число каскадов, на которые вызов вернулся обратно при поиске альтернативного логического канала;

n_{Π} – число попыток в зафиксированной реализации процесса установления соединения; общем случае $0 \leq n_{\Pi} < n_{\text{доп}}$, где $n_{\text{доп}}$ – допустимое число попыток установления соединения;

t_{κ_i} – время прохождения i -го каскада;

t_{o_i} – время обратного прохождения i -го каскада;

t_{Π} – время переключения на другой логический канал.

Каждый эксперимент на имитационной модели дает реализацию трех случайных величин: n_k , n_o и n_{Π} , что позволяет оценить время установления соединения $t_{y.c}$ в соответствии с выражением (4.5). Полученное значение $t_{y.c}$ определяет результат установления соединения:

– если $t_{y.c} \leq t_{\text{доп}}$, то соединение установлено;

– если $t_{y.c} > t_{\text{доп}}$, то соединение установлено, но за время превышающее допустимое, и для данных срочной доставки это имеет критическое значение, т.к. они могли потерять свою актуальность;

– если $n_{\Pi} > n_{\text{доп}}$, то соединение не установлено.

Исходными данными для моделирования являются:

– множество логических каналов, которые могут быть построены от источника i к адресату j – обозначим, как L_{ij} .

– характеристика физических каналов: время передачи тэга по физическому каналу в прямом и обратном направлениях;

– значения вероятностей полной занятости физических каналов;

– допустимое время установления соединения;

– число попыток установления соединения;

– время, выделенное на повторную попытку – переключение на другой логический канал.

В модели накапливаются статистики, позволяющие оценить вероятность установления соединения за время, не превышающее допустимое, а также средние и среднеквадратические значения $t_{y.c}$, характеризующие процесс установления логического канала между портами.

На всем множестве логических каналов L_{ij} производится классификация по числу занятых физических каналов c , т.е. ($c=c_{\min}, \dots, c_{\max}$). В отдельном эксперименте разыгрывается число c номеров занятых физических каналов во множестве L_{ij} и на полученной реализации имитируется процесс прохождения вызова от источника i к адресату j . По факту установления соединения фиксируются значения $n_{тр}$, $n_{от}$ и $n_{п}$. Процедура повторяется столько раз, сколько предусмотрено экспериментом. По результатам экспериментов производятся оценка времени установления соединения.

Реализация количества занятых физических каналов c сводится к «выбору наугад» номеров каналов из d возможных, $c \in d$. Очередной номер занятого физического канала z определяется по формуле $z = \lceil Ud + 1 \rceil$, где U – случайное число, $U \in [0, 1]$, получаемое путем обращения к датчику случайных чисел. Скобки $\lceil \rceil$ означают округление в меньшую сторону. Физическому каналу с номером z присваивается «1» во множестве L_{ij} . Процедура определения z повторяется c раз.

Задача расчета характеристик установления соединения может быть сведена к задаче оценивания математического ожидания $M\xi$ дискретной случайной величины $\xi = f(\alpha)$, где $\alpha = (\alpha_1, \dots, \alpha_d)$ имеет закон распределения вероятностей p (то есть $\alpha \sim p$), который известен. Случайная величина ξ принимает два значения, то есть $\xi \in \{0, 1\}$:

- $\xi = 0$ соответствует установлению соединения,
- $\xi = 1$ – неустановлению соединения.

Значение $M\xi = P\{\xi = 1\}$ имеет смысл вероятности неустановления соединения. Случайная величина ξ невырожденная, то есть $0 < M\xi < 1$.

Интерпретируем случайную величину α как вектор $\alpha = (\alpha_1, \dots, \alpha_d)$, отображающий состояние физических каналов, входящих в моделируемое множество L_{ij} в смысле их наличия или отсутствия. Случайная величина $\alpha_i \in \{0, 1\}$, $i = \overline{1, d}$, d – число физических каналов в моделируемом множестве L_{ij} . Значение $\alpha_i = 1$ имеет смысл отсутствия

свободного i -го физического канала. Случайные величины α_i считаются независимыми, α имеет конечное множество значений $\alpha \in X$, $X = \{x_j; j=1, \dots, n; n=2^d\}$. Для нее распределение $p(x)$, $x \in X$ задается набором вероятностей $p(x_j) = P\{\alpha = x_j\} = p_j \geq 0, j = \overline{1, n}$. С учетом классификации L_{ij} по c оценку $M\xi$ найдем в виде:

$$\tilde{M}\xi = \sum_{c_{\min}}^{c_{\max}} M\xi(c),$$

где $\tilde{M}\xi$ – оценка вероятности неустановления соединения при наличии c физических каналов, непроводящих вызов.

Оценка $\tilde{M}\xi$ находится методом равновзвешенного моделирования.

Для задания состояний $L_{ij}(c)$ на k -м розыгрыше применяется случайный выбор номеров физических каналов c , непроводящих вызов на установление соединения. В результате вектор α получает конкретную реализацию x , содержащую c единиц и $(d-c)$ нулей. В соответствии с правилом прохождения вызова по множеству альтернативных логических каналов L_{ij} и правилом установления соединения вычисляется значение

$$\xi(x|c)_k = f[L_{ij}^k],$$

где $\xi(x|c)_k$ – исход установления (неустановления) соединения при k -м испытании (k -й реализации L_{ij}), $\xi(x|c)_k \in (0,1)$, и вероятность полученной реализации $p(x) = p[L_{ij}^k]$.

Значение $\xi(x|c)_k = 1$ имеет место, если сигнал вызова не дошел до адресата (все логические каналы из множества L_{ij} оказались непроводящими), либо число попыток превысило допустимое значение, либо вызов дошел до адресата и установлено соединение, но за время $t_{y,c} > t_{\text{доп}}$.

Оценка вероятности неустановления соединения между портами i и j коммутатора при c непроводящих физических каналов на множестве всех L_{ij} определяется в виде:

$$\tilde{M}_{\xi}(c) = \frac{C_d^c}{N_{2c}} \sum_{k=1}^{N_{2c}} p(L_{ij}^k | \xi(x|c)=1),$$

где N_c – число разыгранных состояний (реализаций) множества L_{ij} .
Вероятность $p(L_{ij})$ рассчитывается по формуле:

$$p(L_{ij}^k) = \prod_{i=1}^h p(\alpha_i = 1) \prod_{\substack{j=1 \\ j \neq 1}}^{d-h} [1 - p(\alpha_i = 1)].$$

Вероятности $p(\alpha_i = 1)$, $i = \overline{1, d}$ заданы при решении задачи распределения потоков.

Окончательная оценка вероятности неустановления соединения между парой портов i и j примет вид:

$$\tilde{M}_{\xi}(c) = \sum_{c=c_{\min}}^{c_{\max}} \frac{C_d^c}{N_c} \sum_{k=1}^{N_c} p(L_{ij}^k | \xi(x|c)=1). \quad (4.6)$$

Таким образом, выражение (4.6) полностью отражает вероятностный подход к установлению соединения между входящим и исходящим портом коммутатора, построенного на базе «матрицы».

4.4. Модель коммутатора со схемой комплексирования «разделяемая память»

Производительность ЦС инфокоммуникационной сети в большей степени определяется тем, на сколько пересекаются пакеты данных в общем поле памяти ЦС от разных сетей. Пересечение происходит в том случае, если суммарное число запросов на обработку превышает числа запросов, которое может обслуживаться одновременно. В результате возникают временные задержки. Далее приводится аналитический расчет указанных пересечений при симметричном обращении сетей к ЦС [1].

Предполагается, что сети равновероятно обращаются к ЦС (т.е. запросы на сопряжение подчиняются одному и тому же закону распределения. В рассматриваемой модели используются следующие обозначения:

t_1 – отрезок времени на формирование обращения к ЦС;

t_2 – отрезок времени обращения к общему полю памяти ЦС;

t_3 – интервал времени, в течение которого обращение блокируется
 Связь отрезков времен для одного поступающего пакета данных представлена на временной диаграмме рис. 4.1.

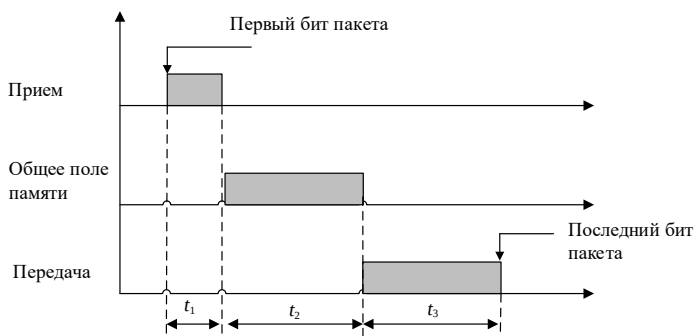
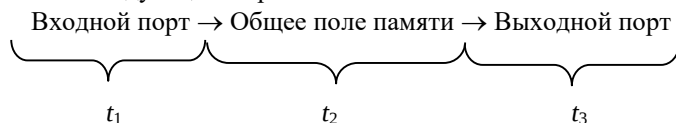


Рис. 4.1. Временная диаграмма прохождения пакета через ЦС

Между элементами ЦС – входными портами для приема кадров, общим полем памяти, выходными портами для передачи кадров существует функциональная связь, которую условно можно представить следующим образом:



Параллельность выполнения процессов в ЦС задается произведением логических показателей использования его модулей $t = t_1 t_2 t_3$.

Вероятность того, что доступ к определенному порту (КМ) будет заблокирован, составит t' , где t' – интервал времени, в течение которого КМ осуществляет передачу, включая пресечения с запросами других КМ. При этом вероятность того, что вход в КМ не заблокирован, составит $(1-t')$.

Следовательно, для N входных портов эта вероятность определяется как $p = (1-t')^N$, где N – число входных портов ЦС.

Вероятность того, что общее поле памяти заблокировано или занято другим КМ, для N входных портов составляет $p' = 1 - (1-t')^N$.

При $N=M=1$ использование общего поля памяти равна $t'=t$. Следовательно, длительность циклов памяти (производительность), нормализованную по системе, можно выразить в виде

$$\Lambda = \left(\frac{1}{t}\right) \left[1 - (1-t')^N\right],$$

где Λ – коэффициент производительности системы.

Положив в уравнении $t' = t$, получим относительно простое выражение для Λ .

Пусть t'' – время цикла КМ или среднее время между обращениями к общей памяти одного КМ; $t_{ож}$ – дополнительное время ожидания освобождения заблокированной памяти. Тогда t и t' определяются как

$$t = \frac{t_2}{t''}; \quad t' = \frac{t_2 + t_{ож}}{t'' + t_{ож}},$$

и окончательно

$$t_{ож} = \frac{t''(t' - t)}{1 - t'}. \quad (4.7)$$

В соответствии с принятыми определениями Λ/N – это отношение скорости исполнения команд процессором КМ при наличии или отсутствия влияния других КМ. Скорость исполнения команд без учета такого влияния равна $1/t''$, а с его учетом – $1/(t'' + t_{ож})$, откуда имеем

$$\frac{\Lambda}{N} = \frac{t''}{t'' + t_{ож}}.$$

Используя выражение (4.7), для $t_{ож}$ находим

$$\frac{\Lambda}{N} = \frac{1-t'}{1-t}$$

или $t' = 1 - \frac{\Lambda}{N}(1-t).$

Подставив значение t' в уравнение и произведя перестановки переменных, получим полиномиальное уравнение N -го порядка, которое может быть решено относительно коэффициента производительности Λ системы:

$$\Lambda - \frac{1}{t} \left(1 - \left\{ 1 - \left[1 - \frac{\Lambda}{N} (1-t) \right] \right\} \right) = 0.$$

Определение t. Показатель эффективного использования представляет собой часть временного цикла КМ, затраченную на конкурентный (непараллельный) цикл памяти. Численное значение t может быть получено перемножением индивидуальных показателей использования элементов, доступ к которым осуществляется последовательно. В связи с этим, определим t_1 , t_2 и t_3 как интервалы времени обращения входных портов к собственной буферной памяти, обращение входных портов к общему выходному порту и времени, в течение которого выходная очередь общего выходного порта заблокирована, т. е. занята другим КМ. Проанализируем каждый из этих показателей.

Определение t_1 . Время формирования обращения к ЦС усредненное по всем входным портам

$$\bar{t}_1 = \frac{L_i}{v_i}, \quad i = \overline{1, S},$$

где L_i – длина пакета данных, передаваемых от i -го КМ;
 v_i – скорость передачи данных от i -го КМ.

Определение t_2 . Время обращения к общему полю памяти определяется как

$$t_2 = \left\lceil \frac{L_i}{l} \right\rceil \tau,$$

где l – длина слова в битах, в соответствии с которым выполняется перезапись пакета в выходной порт;
 τ – время записи одного слова в выходной порт.

Отметим, что t_2 при оценивании времени коммутации через ЦС учитывается дважды – при записи пакета в выходной порт и при считывании их из входного порта.

Определение t_3 . Время t_3 , на которое выходная очередь блокируется $t_3 = t_2 + t_{\text{ож}}$.

Заявка перемещается в очереди в среднем с постоянной скоростью. Среднее число переходов заявки в очереди на одно место вперед за единицу времени равно λ – интенсивности входного потока поступления пакетов данных от N канальных модулей.

При такой скорости перемещения время ожидания освобождения выходной очереди в среднем равно

$$t_{\text{ож}} = \frac{t_2 \Lambda}{1 - \Lambda}.$$

4.5. Математическое моделирование коммутатора с общей шиной

В коммутаторах с общей шиной процессоры портов связывают высокоскоростной шиной, используемой в режиме разделения времени. Чтобы шина не блокировала работу коммутатора, ее производительность должна равняться, по крайней мере, сумме производительности всех портов коммутатора [1].

Для модульных коммутаторов некоторые сочетания модулей с низкоскоростными портами могут приводить к неблокирующей работе, а установка модулей с высокоскоростными портами может приводить к тому, что блокирующим элементом станет, например, общая шина. Кадр должен передаваться по шине небольшими частями, по несколько байт, чтобы передача кадров между несколькими портами происходила в псевдопараллельном режиме, не внося задержек в передачу кадра в целом.

Размер блока данных определяется производителем коммутатора. Некоторые производители, например LANNET или Centillion, выбрали в качестве порции данных, переносимых за одну операцию по шине, ячейку ATM с ее полем данных в 48 байт. Такой подход облегчает трансляцию протоколов локальных сетей в протокол ATM, если коммутатор поддерживает эти технологии.

Входной блок процессора помещает в ячейку, переносимую по шине, тег, в котором указывает номер порта назначения. Каждый выходной блок процессора порта содержит фильтр тегов, который выбирает теги, предназначенные данному порту.

Шина, так же, как и коммутационная матрица, не может осуществлять промежуточную буферизацию, но так как данные кадра разбиваются на небольшие ячейки, то задержек с начальным ожиданием доступности выходного порта в такой схеме нет - здесь работает принцип коммутации пакетов, а не каналов.

Разделяемая шина является простейшей схемой взаимодействия нескольких процессоров портов коммутатора.

Поскольку шина представляет собой разделяемый ресурс, должны быть предусмотрены средства упорядочивания ее использования, например такие схемы, как фиксированная

приоритетность, очередизация по правилу «первым пришел – первым обслужен» и организация цепочек запросов.

Наиболее уязвимыми местами рассматриваемой структуры считается надежность и наложение по времени запросов от различных процессоров портов на владение шиной. Так как существует только единственный путь передачи информации, общая скорость потоков данных в коммутаторе ограничена полосой пропускания магистрали. В этом смысле разделяемая память и коммутационная матрица имеют определенные преимущества. Кроме того, выход из строя магистрали может привести к полному отказу коммутатора.

Здесь несколько процессоров портов связаны через общую шину, осуществляющую арбитраж в целях упорядочения запросов, одновременно поступающих от разных процессоров портов.

Разделяемое оборудование общей шины состоит из арбитра доступа к буферам портов, переключателя шины.

В коммутаторах на базе шин с опросом, подсоединенные к ней порты начинают передачу данных только по запросу контроллера шины. Если в буфере опрашиваемого порта нет подготовленного для передачи сообщения, то специальный логический механизм этого порта формирует сообщение состояния и оповещает о его работоспособности контроллер шины. Пропускная способность такой шины равна сумме пропускных способностей, необходимых для передачи данных от одного порта коммутатора к другому или нескольким другим, и пропускной способности, определяемой интенсивностью потока опрашивающих сообщений между контроллером шины и портом. Время, необходимое для выполнения операции опроса зависит от физической длины шины, поскольку длина определяет задержки распространения. В большинстве случаев задержки распространения опрашивающих сообщений незначительны по сравнению с аналогичными значениями для сообщений с данными.

Требования к ширине полосы пропускания в системах на базе шин с опросом формируется следующим образом:

$$t_s = t_p + \frac{\bar{b}}{C}, \quad (3.45)$$

где t_s – общее время обслуживания передающего порта;

t_p – время, затрачиваемое на сигнал опроса;

b – случайное число, равное числу битов в сообщении, включая служебную информацию;

$\bar{b}$ – среднее значение длины сообщения;

C – пропускная способность шины.

Связь этих переменных представлена на приведенной диаграмме (рис. 4.2).

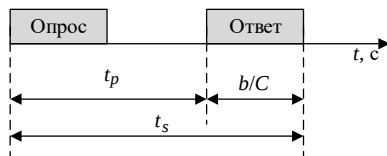


Рис. 4.2. Временная диаграмма связи переменных

Время t_p включает как время обработки, требуемое процессором порта, так и все задержки распространения. Отношение b/C – это время, необходимое для передачи b бит со скоростью C бит в секунду. Если опрашиваемый порт не имеет готового сообщения для передачи, то b – очень мало (стандартное управляющее сообщение о том, что нечего передавать). Если же в буфере сформировано сообщение, то b равно числу бит сообщения; t_s – общая продолжительность цикла опроса, которая обычно представляется случайной величиной, зависящей от числа бит в передаваемом портом сообщении.

С помощью 1, можно оценить среднее время обслуживания шиной одного порта при условии, что дисперсия b значительно меньше, чем $\bar{b}$.

Каждый процессор порта, присоединенный к шине, передает в среднем a сообщений в секунду. При наличии в системе M процессоров портов общая интенсивность потока сообщений составит aM сообщений в секунду, а средний интервал между их поступлениями – $1/aM$ секунд. Следовательно, во избежание неограниченного роста очередей, среднее время обслуживания (t_s) должно быть меньше или равно среднему времени между поступлениями сообщений:

$$\bar{t}_s = t_p + \frac{\bar{b}}{C} \leq \frac{1}{aM}. \quad (4.8)$$

Решая (4.8) относительно C , получаем

$$C \geq \frac{aM\bar{b}}{1 - aMt_p}.$$

При $C \geq aMb/1 - aMt_p$ длина очереди сообщений в каждом порту равна 0 (среднее значение). Это объясняется тем, что любой из портов опрашивается каждые $1/a$ секунд и в то же время каждые $1/a$ секунд в нем формируется очередное сообщение.

4.6. Математическое моделирование ЦС высокого уровня

Основу ЦС типа маршрутизатор транспортного уровня или шлюза составляет многопроцессорный вычислительный комплекс, в структуре которого выделены КМ, ОМ, СОП и схема комплексирования [5].

КМ и ОМ получают доступ к выбранной СОП через схему комплексирования. Если в системе имеется S КМ, N ОМ и M СОП, то одновременно в разные СОП могут обратиться ОМ и КМ в различном сочетании. Если при обращении ОМ и КМ к некоторой СОП, она оказывается занятой, то запрос ставится в очередь. Один и тот же модуль не может выдать новый запрос, если еще не обслужен предыдущий.

Одновременно, если позволяет число ОМ может обрабатываться несколько кадров. Если задание по обработке кадра, приписанное некоторому протокольному процессу, не находит свободного ОМ, то оно ставится в очередь заданий. По освобождении необходимого ОМ, он приписывается первому ожидающему его заданию.

При выполнении процессов может возникнуть необходимость доступа к ресурсам, распределенным в ОПП. Если при запросе некоторого ресурса, он оказывается занятым, то процесс ставится в очередь к данному ресурсу и его дальнейшее выполнение блокируется. ОМ, на котором выполнялась обработка пакета освобождается. При освобождении ресурса он приписывается первому ожидающему его заблокированному процессу, и обработка пакета продолжается с того места алгоритма обработки, где она была прервана.

Каждый тип протокола взаимодействуют с информационными ресурсами, расположенными в общем поле памяти. К информационным ресурсам относятся различные таблицы логических соединений, таблицы локальных адресов, матрица маршрутизации, заголовки очередей, в частности кольцевые очереди и др.)

Протоколы, выполняемые параллельно на различных ОМ могут конкурировать за ресурсы.

Будем считать, что суммарное время ожидания предоставления ресурсов и время их использования не зависит от порядка обращения к ресурсам.

Передача задания от одного протокольного процесса к другому, распределение активных протоколов по ОМ, регулирование доступа процессов протоколов к ресурсам осуществляется ОС. Если при запросе некоторого ресурса он оказывается занятым, то запрос ставится в очередь к ресурсам. При этом ОМ, выполняющий протокольный процесс, освобождается.

Используя рассмотренные модели функционирования КМ и процессов на ОМ, получим выражение для среднего времени задержки пакета в системе. Определим среднее время задержки пакета в маршрутизаторе/шлюзе как средний интервал времени между моментом передачи последнего бита некоторого кадра из КМ в исходящий канал и моментом приема первого бита кадра из входящего канала.

Будем в дальнейшем считать индексом i все параметры, относящиеся к процессам приема КМ, $i = \overline{1, S}$; x – протокольные процессы, выполняющиеся на ОМ, $x = \overline{1, N}$; N – число уровней ОМ (уровней OSI) протокольных процессов; j – процессы передачи КМ; $j = \overline{1, S}$.

Общее число процессов, которые могут быть одновременно активны в системе $K = N + 2S$.

Обозначим $T = [t_k]_{k=1, K}$, где t_k – среднее время нахождения заявки в системе k -го процесса.

Тогда

– $t_k = \tau_i$, $i = \overline{1, S}$ – для процессов приема;

– $t_k = \tau_j$, $j = \overline{1, S}$ – для процессов передачи;

– $t_k = \tau_x$, $x = \overline{1, N}$ – для протокольных процессов, выполняемых на ОМ;

В соответствии с изложенной концепцией имеем время приема пакета в ЦС

$$\tau_i = t_{\text{ож}_i}^{\text{пр}} + \frac{L_i}{v_i} + \tau_{\text{тм}}, \quad i = \overline{1, S},$$

где $t_{\text{ож}_i}^{\text{пр}}$ – время ожидания доступа к СОП при записи пакета из i -го порта в общее поле памяти ЦС;

$\tau_{\text{тм}}$ – время обращения к таблице маршрутизации;

L_i – длина пакета, принятая в сети передающей пакет в i -й порт ЦС;

v_i – скорость канала связи сети передающей пакет в i -й порт ЦС.

Время передачи пакета из ЦС:

$$\tau_j = t_{\text{ож}_j}^{\text{пер}} + \frac{L_j}{v_j}, j = \overline{1, S},$$

где $t_{\text{ож}_j}^{\text{пер}}$ – время ожидания доступа к СОП для чтения пакета из общего поля памяти в j -й порт ЦС;

L_j – длина пакета, принятая в сети принимающей пакет из j -го порта ЦС;

v_j – скорость канала связи сети принимающей пакет из j -го порта ЦС.

Время выполнения протокольных процессов на ОМ определяется суммированием следующих составляющих

$$\tau_x = t^P + t^A + t^W, \quad (4.9)$$

где t^P – суммарное среднее время ожидания предоставления информационных ресурсов, необходимых при выполнении процессов на x -м ОМ;

t^A – суммарное среднее время ожидания пакета в очереди активных пакетов на обработку к x -му ОМ.

t^W – среднее время, необходимое для сборки транспортного блока (только для 4-го уровня).

Среднее время задержки пакета в маршрутизаторе/шлюзе может быть вычислено по формуле

$$\bar{t} = \sum_{k=1}^K e_k t_k,$$

где e_k – коэффициент пропорциональности, учитывающий частоту прохождения заявок через процесс k .

Значения $t_{\text{ож}_j}^{\text{пер}}$ $t_{\text{ож}_j}^{\text{пр}}$ и все составляющие вектора τ_x могут быть найдены с использованием замкнутой СеМО.

При нахождении составляющих вектора τ_x в каждом случае определяется число источников и обслуживающих приборов, и строится замкнутая СеМО. Затем оценивается среднее время ожидания предоставления данного ресурса с использованием рекурсивной процедуры (4.1) - (4.4) и, наконец, суммарное среднее время ожидания предоставления всех ресурсов (4.9), необходимых при обработке пакета.

4.7. Оценка вероятности потерь в ЦС

Из рассмотренной модели КМ следует, что вероятность потери $P_{\text{пот}}$ принимаемых кадров для i -го входящего канала определяется количеством мест в буферной памяти, закрепленных за i -м входящим каналом.

Рассмотрим расчет количество мест в буферной памяти (емкость буферной памяти) в зависимости от вероятности потери кадра.

Входящие в ЦС пакеты данных записываются во входные порты ЦС со скоростью канала передачи данных. Затем пакеты обрабатываются процессором КМ, выполняющим накопление этих пакетов в выходном порту и далее пакет в нужном формате передается по каналу в выходящем из ЦС направлении. Копия принятого в КМ кадра хранится в его буферной памяти до получения квитанции АСК (логического подтверждения) о безошибочной доставке данных адресату или промежуточному узлу. Отсутствие АСК в течение определенного интервала времени (time-out) вынуждает повторить передачу пакета по выходящему каналу в направлении адресата. Это обеспечивает логическую защиту от ошибок в передаче и возможной потери данных из-за отсутствия свободных емкостей в принимающем КМ. При отсутствии места в буферной памяти КМ входящий в него пакет получает отказ. Тогда пакет направляется по другому маршруту или запрос периодически повторяется.

Поступивший в буферную память КМ пакет занимает ее в течение следующих интервалов времени:

- запись блока данных со скоростью канала связи;
- ожидание в очереди на обработку процессором КМ;
- обработка пакета;
- передача пакета в выходной порт ЦС;
- передача из выходного порта ЦС в формате исходящей сети;
- ожидание квитанции об удачной доставке с возможным повторением передачи по истечении интервала time-out.

Поиск свободного буфера (места) в порту КМ осуществляется лишь в момент появления запроса на него. Каждый буфер может быть выделен для любого канала. Размер буфера может определяться пакетом максимальной длины или быть меньше его. В последнем случае для приема пакета выделяется цепочка буферов, и память при большом разбросе длин пакетов используется эффективнее.

Емкость буферной памяти КМ можно определить исходя из допустимой вероятности отказов входящим в ЦС пакетам.

Для определения емкости буферной памяти конкретного КМ воспользуемся системой массового обслуживания $M|M|1|n$, т. е. однолинейной СМО с ограниченной очередью. Запрос из входящего потока, заставший прибор обслуживания занятым, ожидает начала обслуживания в очереди, если в нем имеется свободное место. Если же все n мест для ожидания заняты, запрос покидает систему необслуженным (теряется).

Обозначим, что $i, t \geq 0$ является процессом гибели и размножения и ненулевые параметры λ_i, μ_i определяются следующим образом: $\lambda_i = \lambda, 0 \leq i \leq n-1, \mu_i = \mu, 1 \leq i \leq n$, где λ – интенсивность входного потока заявок и μ – интенсивность обслуживания заявок. Тогда из формулы для стационарных вероятностей процесса гибели и размножения следует, что стационарные вероятности числа запросов в рассматриваемой системе имеют вид

$$\pi_i = \rho^i \frac{1-\rho}{1-\rho^{n+1}}, \quad 1 \leq i \leq n.$$

К важнейшим характеристикам систем, в которых возможна потеря запросов, относится вероятность $P_{\text{пот}}$ того, что произвольный запрос будет потерян. Для рассматриваемой СМО справедливо утверждение о том, что вероятность потери произвольного запроса совпадает с вероятностью того, что в произвольный момент времени все места для ожидания заняты, т. е. справедливо

$$P_{\text{пот}} = \rho^n \frac{1-\rho}{1-\rho^{n+1}}. \quad (4.10)$$

Формула (4.10) может использоваться для планирования необходимой емкости буферной памяти канального модуля ЦС в зависимости от загрузки системы и значения допустимой вероятности потери запроса в системе.

ЗАКЛЮЧЕНИЕ

Общая структура центра сопряжения разнородных сетей строится на модулях, из которых можно синтезировать ЦС любого уровня взаимодействия сетей согласно модели OSI.

Структурные модули включают: канальный модуль – для выполнения процессов приема-передачи кадров; процессорный модуль – для выполнения уровневых протоколов ЦС; общее поле памяти – для хранения информационных ресурсов и поступающих пакетов; схема комплексирования – для объединения всех модулей между собой в рамках одного центра сопряжения; виртуальный канал – для отображения входа/выхода КМ и поступающей на него нагрузки.

В пособии рассматриваются модели, обеспечивающие расчет вероятностно-временных характеристик ЦС разной архитектуры, включая коммутаторы матричного типа, с разделяемой шиной, с общей памятью и ЦС типа маршрутизатор и шлюз.

Имитационная модель коммутатора со схемой комплексирования «матрица» позволяет оценить производительность коммутатора при заданных параметрах входного потока и требований на допустимое время установления соединения между входным и выходным портами коммутатора.

Аналитическая модель коммутатора с общей разделяемой памятью позволяет оценивать производительность коммутатора при заданных параметрах входного потока и количестве секций общего поля памяти.

Математическая модель коммутатора с общей шиной, устанавливающая связь между производительностью шины и количеством портов ввода/вывода и позволяющая оценить эффективность коммутатора.

Предложена модель оценки среднего времени и производительности маршрутизатора/шлюза на уровне обращений к общему полю памяти при реализации протоколов трансляции.

Рассмотрены функции канального модуля центра сопряжения и модели расчета времени фрагментации/сборки пакетов, согласования скоростей взаимодействующих разнородных сетей и анализ буферной памяти центра сопряжения.

ЛИТЕРАТУРА

1. Кутузов О.И., Сергеев В.Г., Татарникова Т.М. Коммутаторы в корпоративных сетях. моделирование и расчет. Под ред. О.И. Кутузова. СПб.: Судостроение, 2003. – 170 с.
2. Куроуз Д., Росс К. Компьютерные сети: нисходящий подход. 6-е изд. – Москва: Издательство «Э», 2016. – 912 с.
3. Руденков Н.А., Долинер Л.И. Основы сетевых технологий: Учебник для вузов. Екатеринбург: Изд-во УФУ, 2011 – 300 с.
4. Олифер В., Олифер Н. 0-54 Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 5-е изд. – СПб.: Питер, 2016. – 992 с.
5. Татарникова Т.М. Модели и методы структурного синтеза центров сопряжения корпоративных информационных сетей. Дисс. на соискание ученой степени доктора технических наук / Санкт-Петербург, 2007.

Учебное издание

Татарникова Татьяна Михайловна, доктор технических наук, доцент
Миклуш Виктория Александровна, старший преподаватель
Краева Екатерина Витальевна, ассистент

КОММУТАТОРЫ

Печатается в авторской редакции.

Подписано в печать 21.05.2020. Формат 60×90 1/16.
Гарнитура Times New Roman. Печать цифровая.
Усл. печ. л. 5,25. Тираж 30 экз. Заказ № 920.
РГТМУ, 192007, Санкт-Петербург, Воронежская, 79