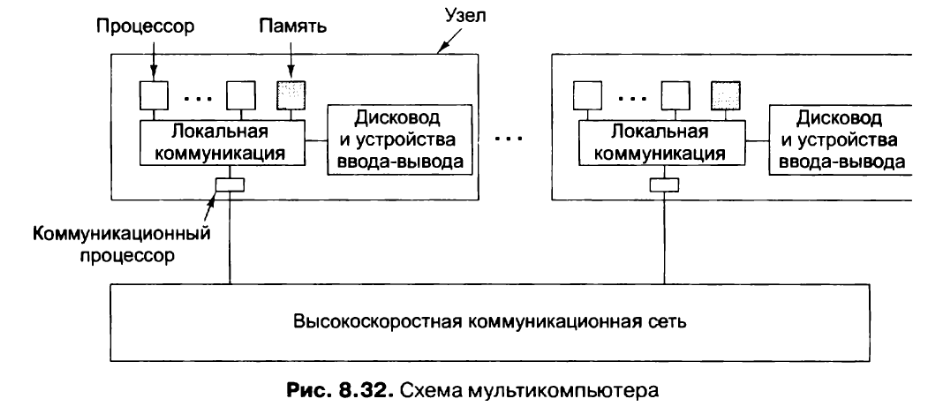
Отличительная особенность мультикомпьютера в том, что каждый процессор имеет собственную память, в которую другие компьютеры не имеют прямого доступа. Программы на разных процессорах взаимодействуют друг с другом с помощью примитивов *send* и *receive*, которые используются для передачи сообщений. Это различие полностью меняет модель программирования.

Каждый узел в мультикомпьютере состоит из одного или нескольких процессоров, ОЗУ (общие для процессоров данного узла), диска и (или) других узлов ввода/вывода, а также процессора передачи данных. Процессоры передачи данных связаны между собой по высокоскоростной коммуникационной сети.

Итак, мультикомпьютеры связываются через сети межсоединений. Мультикомпьютеры и мультипроцессоры очень сходны в этом отношении, т.к. мультипроцессоры также содержат модули памяти, которые также должны быть связаны между собой и с процессором. Основная причина сходства коммуникационных связей в мультипроцессоре и мультикомпьютере заключается в том, что в обоих случаях применяется передача сообщений.

Топология коммуникационной сети (кольцо, решетка, дерево и т.д.) определяет, как расположены линии связи (дуги) и коммутаторы(узлы).

Коэффициент разветвления сети – число линий связей узла.

Расстояние между двумя узлами - число дуг, которые нужно пройти, чтобы попасть из одной вершины в другую. Диаметр сети - максимальное расстояние между двумя вершинами. Диаметр определяет самую большую задержку при передаче пакетов от одного процессора к другому или от процессора к памяти.

*Пропускная способность сети* – количество данных, которое она может передать в секунду. *Бисекционная* пропускная способность – минимальная из всех возможных (минимальный разрез сети).

Сети можно характеризовать по их *размерности* - числу возможных вариантов перехода.

*Коммутация каналов* – перед тем, как послать пакет, весь путь от начала до конца резервируется заранее. Все пор­ты и буферы затребованы заранее и поэтому биты на полной скорости перемещают­ся от источника к потреби­те­лю.  На рис. 6.25 – резерви­ру­ет­ся канал с использованием трех входных и трех выходных портов.

*Коммутация с проме­жу­то­чным хранением* – здесь не требуется предварительного резервирования. Из исходного пункта посылается целый пакет к первому коммутатору, где он хранится целиком. Затем он передается следующему коммутатору и так до тех пор, пока не прибудет к месту назначения. П

Коммутаторы с промежуточным хранением должны отправлять пакеты в буфер. Если бы не было *буферизации*, входящие пакеты, которым нужен занятый в данный выходной порт, пропадали бы.  Применяется три метода буферизации.

*При буферизации на входе* один или несколько буферов связываются с каждым входным портом в форме очереди типа FIFO. Если пакет нельзя передать, то он просто ждет своей очереди. Но если пакет ожидает, то следующий пакет также ожидает очереди, даже если требуемый ему порт свободен. Такая ситуация называется *блокировкой начала очереди*.

Проблема может быть устранена с помощью *буферизации на выходе*. В этой системе буферы связаны с выходными портами. И при буферизации на входе и при буферизации на выходе с каждым портом связано определенное количество буферов. Если место недостаточно для хранения всех пакетов, то пакеты пропадают. Для разрешения этой проблемы используют *общую буферизацию*, при которой один буферный пул динамически распределяется по портам по мере необходимости. Однако такой метод требует достаточно сложного управления.

Хотя метод коммутации с промежуточным хранением гибок и эффективен, здесь возникает проблема с возрастающей задержки при передаче данных по сети межсоединений.

Существует идея разработки гибридной сети межсоединений, объединяющую в себе коммутацию каналов и коммутацию пакетов. Например, каждый пакет можно разделить на части и как только первая часть пакета поступила в коммутатор, ее можно передавать дальше, даже если оставшиеся части пакета еще не прибыли.

Такой подход отличается от коммутации каналов тем, что ресурсы не резервируются заранее. Следовательно, возможна конфликтная ситуация в соревновании за право обладания ресурсами. При коммутации *без буферизации пакетов*, если первый блок пакета не может двигаться дальше, оставшаяся часть пакета продолжает поступать в коммутатор. В худшем случае эта схема превращается в коммутацию с промежуточным хранением. При другом типе маршрутизации, т.н. «wormhole routing» (червоточина), если первый блок не может двигаться дальше, то в исходный пункт подается сигнал остановить передачу и пакет может растянуться на несколько коммутаторов. После освобождения ресурсов продвижение пакета продолжается.

В любой сети соединений с размерностью от один и выше можно выбирать, по какому пути передавать пакеты от одного узла к другому. Правило, определяющее, какую последовательность узлов должен пройти пакет при движении от исходного пункта к пункту назначения, называется *алгоритмом выбора маршрута*.

**Маршрутизация: 1.**от источника **2.** распределенная. При *маршрутизации от источника* источник определяет весь путь по сети заранее. Этот путь выражается списком из номеров портов, которые нужно будет использовать в каждом коммутаторе по пути к пункту назначения. Если путь лежит через k коммутаторов, то первые k байтов в каждом пакете будут содержать kномеров выходных портов, по 1 байту на каждый порт. Когда пакет доходит до коммутатора, первый байт отсекается и используется для определения выходного порта. Оставшаяся часть пакета используется для направления в соответствующий порт.

При *распределенной маршрутизации* каждый коммутатор сам решает, в какой порт отправить каждый приходящий пакет. Если выбор одинаков для каждого пакета, направленного к одному и тому же конечному пункту, то маршрутизация является *статической*. Если коммутатор при выборе принимает во внимание текущий трафик, то маршрутизация является *адаптивной*.

Популярным алгоритмом маршрутизации, который применяется для прямоугольных решеток с любым числом измерений, является *пространственная маршрутизация*. В соответствии с этим алгоритмом пакет сначала перемещается вдоль оси x до нужной координаты, и т.д.