

## Теоретический материал по топологиям вычислительных систем

(IV курс, дисциплина «Вычислительные системы и комплексы»)

В основе архитектуры любой многопроцессорной вычислительной системы лежит способность к обмену данными между компонентами этой ВС. Коммуникационная система ВС представляет собой сеть, узлы которой связаны трактами передачи данных — каналами. В роли узлов могут выступать процессоры, модули памяти, устройства ввода/вывода, коммутаторы либо несколько перечисленных элементов, объединенных в группу. Организация внутренних коммуникаций вычислительной системы называется *топологией*.

### 1 Основные характеристики топологии и ее классификация

Топологию сети межсоединений (СМС) определяет множество узлов  $N$ , объединенных множеством каналов  $C$ . Связь между узлами обычно реализуется по *двухточечной схеме* (point-to-point). Любые два узла, связанные каналом связи, называют *смежными узлами* или *соседями*. Каждый канал  $c = (x, y) \in C$  соединяет один *узел-источник* (source node)  $x$  с одним *узлом-получателем* (recipient node)  $y$ , где  $x, y \in N$ . Узел-источник, служащий началом канала  $c$ , будем обозначать  $s_c$ , а узел-получатель — второй конец канала —  $r_c$ . Часто пары узлов соединяют два канала — по одному в каждом направлении. Канал  $c = (x, y)$  характеризуется шириной ( $w_c$  или  $w_{xy}$ ) — числом сигнальных линий; частотой ( $f_c$  или  $f_{xy}$ ) — скоростью передачи битов по каждой сигнальной линии; задержкой ( $t_c$  или  $t_{xy}$ ) — временем пересылки бита из узла  $x$  в узел  $y$ . Для большинства каналов задержка находится в прямой зависимости от физической длины линии связи ( $l_c$ ) и скорости распространения сигнала ( $v$ ):  $l_c = vt_c$ . Полоса пропускания канала  $b_c$  определяется выражением  $b_c = w_c l_c$ .

В зависимости от того, остается ли конфигурация взаимосвязей неизменной, по крайней мере пока выполняется определенное задание, различают сети со *статической* и *динамической* топологиями. В статических сетях структура взаимосвязей фиксирована. В сетях с динамической топологией в процессе вычислений конфигурация взаимосвязей с помощью программных средств может быть оперативно изменена.

Узел в сети может быть терминальным, то есть источником или приемником данных, коммутатором, пересылающим информацию с входного порта на выходной, или совмещать обе роли. В сетях с *непосредственными (прямыми) связями* (direct networks) каждый узел одновременно является как терминальным узлом, так и коммутатором, и сообщения пересылаются между терминальными узлами напрямую. В сетях с *косвенными связями* (indirect networks) узел может быть либо терминальным, либо коммутатором, но не одновременно, поэтому сообщения передаются опосредованно, с помощью выделенных коммутирующих узлов. Существуют также такие топологии, которые нельзя однозначно причислить ни к прямым, ни к косвенным. Любую прямую СМС можно изобразить в виде косвенной, разделив каждый узел на два — терминальный узел и узел коммутации. Современные прямые сети реализуются именно таким образом — коммутатор отделяется от терминального узла и помещается в выделенный маршрутизатор. Основное преимущество прямых СМС в том, что коммутатор может использовать ресурсы терминальной части своего узла. Это становится существенным, если учесть, что, как правило, последний включает в себя вычислительную машину или процессор. Три важнейшими атрибутами СМС являются:

- стратегия синхронизации;
- стратегия коммутации;
- стратегия управления.

Две возможных стратегии синхронизации операций в сети — это *синхронная* и *асинхронная*. В синхронных СМС все действия жестко согласованы во времени, что обеспечивается за счет единого генератора тактовых импульсов (ГТИ), сигналы которого одновременно транслируются во все узлы. В асинхронных сетях единого генератора нет, а функции синхронизации распределены по всей системе, причем в разных частях сети часто используются локальные ГТИ.

В зависимости от выбранной стратегии коммутации различают *сети с коммутацией соединений* и *сети с коммутацией пакетов*. Как в первом, так и во втором варианте информация пересылается в виде пакета. *Пакет* представляет собой группу битов, для обозначения которой применяют также термин *сообщение*.

В сетях с коммутацией соединений путем соответствующей установки коммутирующих элементов сети формируется тракт от узла-источника до узла-получателя, сохраняющийся, пока весь доставляемый пакет не достигнет пункта назначения. Пересылка сообщений между определенной парой узлов производится всегда по одному и тому же маршруту.

Сети с коммутацией пакетов предполагают, что сообщение самостоятельно находит свой путь к месту назначения. В отличие от сетей с коммутацией соединений, маршрут от исходного пункта к пункту назначения каждый раз может быть иным. Пакет последовательно проходит через узлы сети.

Очередной узел запоминает принятый пакет в своем буфере временного хранения, анализирует его и делает выводы, что с ним делать дальше. В зависимости от загруженности сети принимается решение о возможности немедленной пересылки пакета к следующему узлу и о дальнейшем маршруте следования пакета на пути к цели. Если все возможные тракты для перемещения пакета к очередному узлу заняты, в буфере узла формируется очередь пакетов, которая «рассасывается» по мере освобождения линий связи между узлами (если очередь также насыщается, согласно одной из стратегий маршрутизации может произойти так называемый «сброс хвоста» (tail drop), отказ от вновь поступающих пакетов).

СМС можно также классифицировать по тому, как в них организовано управление. В некоторых сетях, особенно с переключением соединений, принято *централизованное управление* (рисунок 1). Процессоры посылают запрос на обслуживание в единый контроллер сети, который производит арбитраж запросов с учетом заданных приоритетов и устанавливает нужный маршрут. К данному типу следует отнести сети с шинной топологией. Процессорные матрицы также строятся как сети с централизованным управлением, которое осуществляется сигналами от центрального процессора. Приведенная схема применима и к сетям с коммутацией пакетов. Здесь тег маршрутизации, хранящийся в заголовке пакета, определяет адрес узла назначения. Большинство из серийно выпускаемых ВС имеют именно этот тип управления.

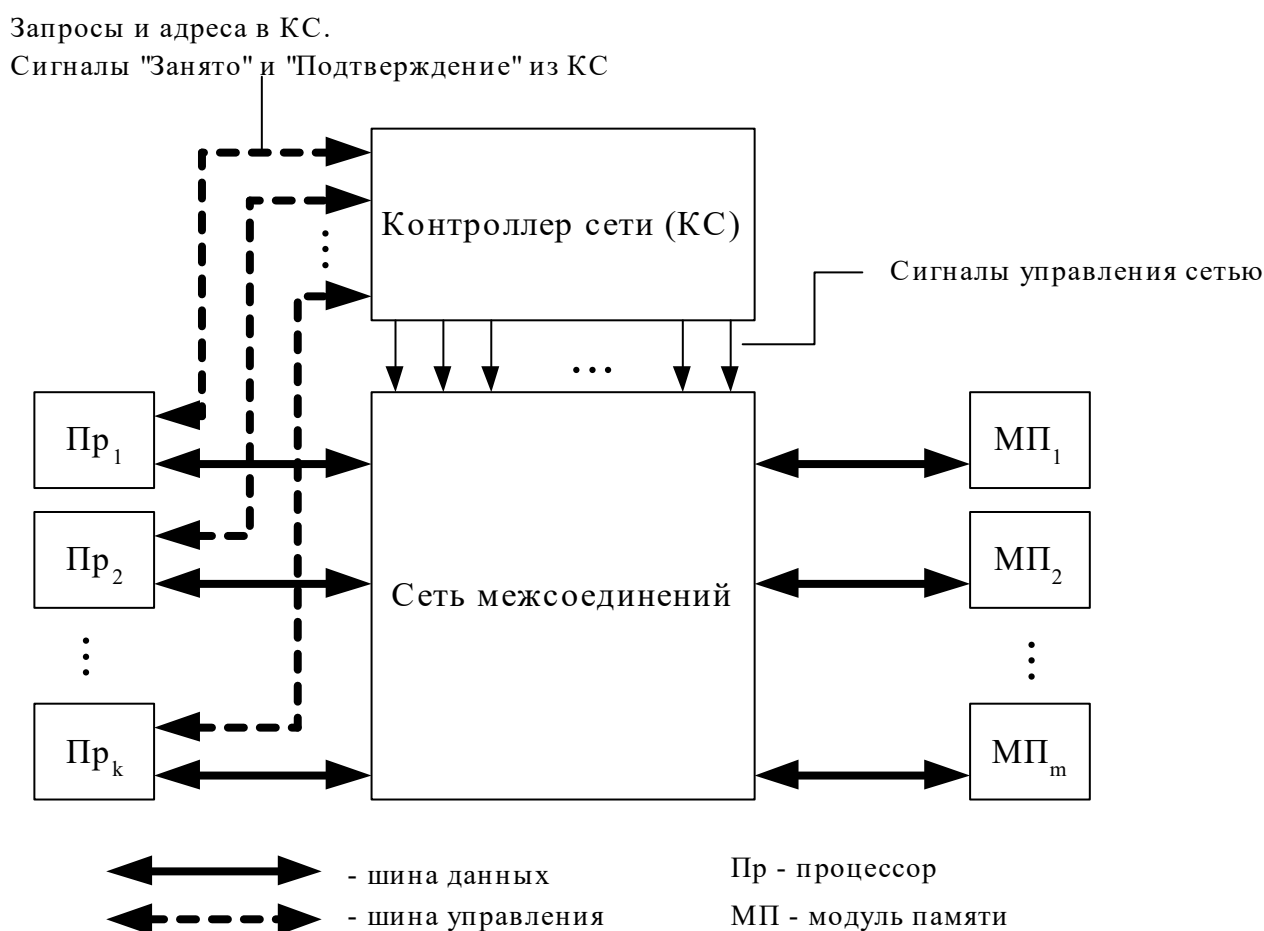


Рисунок 1 – Структура сети с централизованным управлением

В схемах с *децентрализованным управлением* функции управления распределены по узлам сети. Вариант с *централизацией* проще реализуется, но расширение сети в этом случае связано со значительными трудностями. Децентрализованные сети в плане подключения дополнительных узлов значительно гибче, однако взаимодействие узлов в таких сетях существенно сложнее. В ряде сетей связь между узлами обеспечивается посредством множества коммутаторов, но существуют также сети с одним коммутатором. Наличие большого числа коммутаторов ведет к увеличению времени передачи сообщения, но позволяет использовать простые переключающие элементы. Подобные сети обычно строятся как многоступенчатые.

## 2 Метрики сетевых соединений

При описании СМС их обычно характеризуют с помощью следующих параметров:

- размера сети ( $N$ );
- числа связей ( $I$ );
- диаметра ( $D$ );
- порядка узла ( $d$ );
- пропускной способности ( $W$ );
- задержки ( $T$ );
- связности ( $Q$ );
- ширины бисекции ( $B$ );
- полосы бисекции ( $b$ ).

**Размер сети** (network size) численно равен количеству узлов, объединяемых сетью.

**Число связей** (number of links) — это суммарное количество каналов между всеми узлами сети. В плане стоимости лучшей следует признать ту сеть, которая требует меньшего числа связей.

**Диаметр сети** (network diameter), называемый также *коммуникационным расстоянием*, определяет минимальный путь, по которому проходит сообщение между двумя наиболее удаленными друг от друга узлами сети. *Путь* (path) в сети — это упорядоченное множество каналов  $P = \{c_1, c_2, \dots, c_n\}$ , по которым данные от узла-источника, последовательно переходя от одного промежуточного узла к другому, поступают на узел-получатель. Для обозначения отрезка пути между парой смежных узлов применяют термин *переход* (hop). Минимальный путь от узла  $x$  до узла  $y$  — это путь с минимальным числом переходов. Если обозначить число переходов в минимальном пути от узла  $x$  до узла  $y$  через  $H(x, y)$ , то диаметр сети  $D$  — это наибольшее значение  $H(x, y)$  среди всех возможных комбинаций  $x$  и  $y$ . Так, в цепочке из четырех узлов наибольшее число переходов будет между крайними узлами, и «диаметр» такой цепочки равен трем. С возрастанием диаметра сети увеличивается общее время прохождения сообщения, поэтому разработчики ВС стремятся по возможности обходиться меньшим диаметром.

**Порядок узла.** Каждый узел сети связан с прочими узлами множеством каналов  $C_{ix} = C_{ix} \cup C_{ox}$ , где  $C_{ix} = \{c: r_c = x\}$  — множество входных каналов, а  $C_{ox} = \{c: s_c = x\}$  — множество выходных каналов. Порядок узла  $x$  представляет собой сумму числа входных  $|C_{ix}|$  и выходных  $|C_{ox}|$  каналов узла, то есть равен числу узлов сети, с которыми данный узел связан напрямую. Например, в сети, организованной в виде матрицы, где каждый узел имеет каналы только к ближайшим соседям (слева, справа, сверху и снизу), порядок узла равен четырем. В вычислительной системе СМ-1, построенной по топологии гиперкуба, каждый узел связан с 12 другими узлами, следовательно, порядок узлов равен 12. Увеличение значения этой метрики ведет к усложнению коммутационных устройств сети и, как следствие, к дополнительным задержкам в передаче сообщений. С другой стороны, повышение порядка узлов позволяет реализовать топологии, имеющие меньший диаметр сети, и тем самым сократить время прохождения сообщения. Так, если сеть, состоящая из 4096 узлов ( $2^{12}$  или  $64^2$ ), построена по матричной схеме, ее диаметр составляет 126, а для гиперкуба — только 12. Разработчики ВС обычно отдают предпочтение таким топологиям, где порядок всех узлов одинаков, что позволяет строить сети по модульному принципу.

**Пропускная способность сети** (network bandwidth) характеризуется количеством информации, которое может быть передано по сети в единицу времени. Обычно измеряется в мегабайтах в секунду или гигабайтах в секунду без учета издержек на передачу избыточной информации, например битов паритета.

**Задержка сети** (network latency) — это время, требуемое на прохождение сообщения через сеть. В сетях, где время передачи сообщений зависит от маршрута, говорят о средней, минимальной и максимальной задержках сети.

**Связность сети** (network connectivity) можно определить как минимальное число узлов или линий связи, которые должны выйти из строя, чтобы сеть распалась на две непересекающихся сети. Следовательно, связность сети характеризует устойчивость сети к повреждениям, то есть ее способность обеспечивать функционирование ВС при отказе компонентов сети.

**Ширина бисекции** (bisection width). Для начала определим понятие *среза сети* (cut of network). Срез сети  $C(N_1, N_2)$  — это множество каналов, разрыв которых разделяет множество узлов сети  $N$  на два непересекающихся набора узлов  $N_1$  и  $N_2$ . Каждый элемент  $C(N_1, N_2)$  — это канал, соединяющий узел из набора  $N_1$  с узлом из  $N_2$ . Бисекция сети — это срез сети, разделяющий ее примерно пополам, то есть так, что  $|N_2| < |N_1| < |N_2| + 1$ . Ширину бисекции  $B$  характеризуют минимальным числом каналов, разрываемых при всех возможных бисекциях сети:

$$B = \min |C(N_1, N_2)|$$

Ширина бисекции позволяет оценить число сообщений, которые могут быть переданы по сети одновременно, при условии что это не вызовет конфликтов из-за попытки использования одних и тех же узлов или линий связи.

**Полоса бисекции** (bisection bandwidth) — это наименьшая полоса пропускания по всем возможным бисекциям сети. Она характеризует пропускную способность тех линий связи, которые разрываются при бисекции сети, и позволяет оценить наихудшую пропускную способность сети при попытке одномоментной передачи нескольких сообщений, если эти сообщения должны проходить из одной половины сети в другую. Полоса бисекции  $b$  определяется выражением

$$b = \min_{bisection} B(N_1, N_2).$$

Для сетей с одинаковой шириной полосы во всех  $b$  каналах справедливо соотношение:  $b = b_c \times B$ . Малое значение полосы бисекции свидетельствует о возможности конфликтов при одновременной пересылке нескольких сообщений.

### 3 Статические топологии

К статическим топологиям СМС относят такие, где между двумя узлами возможен только один прямой фиксированный путь, то есть статические топологии не предполагают наличия в сети коммутирующих устройств. Если же такие устройства имеются, то используются они только перед выполнением конкретной задачи, а в процессе всего времени вычислений топология СМС остается неизменной.

Из возможных критериев классификации статических сетей чаще всего выбирают их размерность. С этих позиций различают:

- одномерные топологии (линейный массив);
- двумерные топологии (кольцо, звезда, дерево, решетка, систолический массив);
- трехмерные топологии (полносвязная топология, хордальное кольцо);
- гиперкубическую топологию.

#### 1) Линейная топология

В простейшей *линейной топологии* узлы сети образуют одномерный массив и соединены в *цепочку* (рисунок 9). Линейная топология характеризуется следующими параметрами  $D = N-1$ ;  $d = 2$ ;  $I = N-1$ ;  $B=1$ .

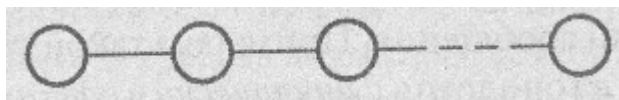


Рисунок 9 - Линейная топология

Линейная топология не обладает свойством полной симметричности, поскольку узлы на концах цепочки имеют только одну коммуникационную линию, то есть их порядок равен 1, в то время как порядок остальных узлов равен 2. Время пересылки сообщения зависит от расстояния между узлами, а отказ одного из них способен привести к невозможности пересылки сообщения. По этой причине в линейных СМС используют отказоустойчивые узлы, которые при отказе изолируют себя от сети, позволяя сообщению миновать неисправный узел.

#### 2) Кольцевые топологии

Стандартная *кольцевая топология* представляет собой линейную цепочку, концы которой соединены между собой (рисунок 10а). В зависимости от числа каналов между соседними узлами (один или два) различают *однонаправленные* и *двунаправленные* кольца. Кольцевая топология характеризуется следующими параметрами:  $D = \min [N/2]$ ;  $d = 2$ ;  $I = N$ ;  $B = 2$ .

Кольцевая топология, по сравнению с линейной, традиционно была менее популярной, поскольку добавление или удаление узла требует демонтажа сети.

Один из способов разрешения проблемы большого диаметра кольцевой сети - добавление линий связи в виде хорд, соединяющих определенные узлы кольца. Подобная топология носит название *хордальной*. Если хорды соединяют узлы с шагом 1 или  $N/2 - 1$ , диаметр сети уменьшается вдвое. На рисунке 10б показана *хордальная кольцевая сеть* с шагом 3.

В качестве практических примеров для топологии кольца следует назвать сети Token Ring, разработанные фирмой IBM, а также вычислительные системы KSR1 и SCI.

Дальнейшее увеличение порядка узлов позволяет добиться еще большего сокращения тракта передачи сообщения. Примером такой топологии может служить показанная на рисунке 10в топология с *циклическим сдвигом связей*. Здесь стандартная кольцевая топология с  $N$  узлами дополнена соединениями между всеми узлами  $i$  и  $j$ , для которых  $|i - j|$  совпадает с целой степенью числа 2. Алгоритмы маршрутизации для подобной сети чрезвычайно эффективны, однако порядок узлов по мере разрастания сети увеличивается.

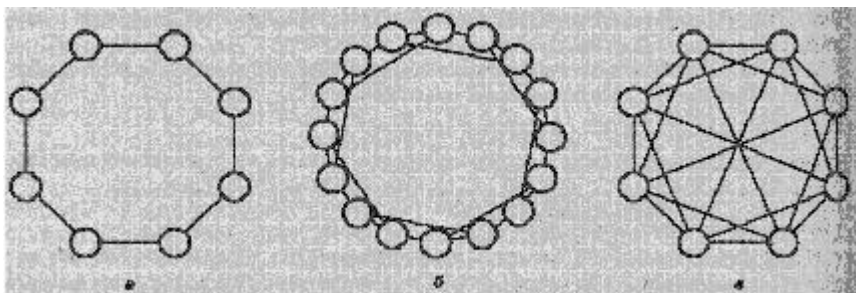


Рисунок 10 - Кольцевые топологии: а — стандартная; б — хордальная; в — с циклическим сдвигом связей

### 3) Звездообразная топология

*Звездообразная сеть* объединяет множество узлов первого порядка посредством специализированного центрального узла — концентратора (рисунок 11). Топология характеризуется такими параметрами:  $D = 2$ ;  $d = N - 1$ ;  $I = N - 1$ ;  $B = 1$ .

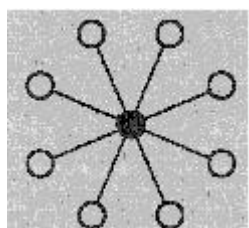


Рисунок 11 - Звездообразная топология

Звездообразная организация узлов и соединений редко используется для объединения процессоров многопроцессорной ВС, но хорошо работает, когда поток информации идет от нескольких вторичных узлов, соединенных с одним первичным узлом, например при подключении терминалов. Общая пропускная способность сети обычно ограничивается быстродействием концентратора, аналогично тому, как сдерживающим элементом в однотипной топологии выступает шина. По производительности эти топологии также идентичны. Основное преимущество звездообразной схемы в том, что конструктивное исполнение узлов на концах сети может быть очень простым.

### 4) Древоподобные топологии

Еще одним вариантом структуры СМС является *древовидная топология* (рисунок 12а). Сеть строится по схеме так называемого строго двоичного дерева, где каждый узел более высокого уровня связан с двумя узлами следующего по порядку более низкого уровня. Узел, находящийся на более высоком уровне, принято называть *родительским*, а два подключенных к нему нижерасположенных узла — *дочерними*. В свою очередь, каждый дочерний узел выступает в качестве родительского для двух узлов следующего более низкого уровня. Отметим, что каждый узел связан только с двумя дочерними и одним родительским. Если  $h$  — высота дерева (количество уровней в древовидной сети), определяемая как  $\max \lceil \log_2 N \rceil$ , то такую сеть можно охарактеризовать следующими параметрами:  $D = 2(h - 1)$ ;  $d = 3$ ;  $I = N - 1$ ;  $B = 1$ . Так, вычислительная система из 262 144 узлов при решетчатой топологии будет иметь диаметр 512, а в случае строго бинарного дерева — только 36. Топология двоичного дерева была использована в мультипроцессорной системе DADO из 1023 узлов, разработанной в Колумбийском университете.

При больших объемах пересылок между несмежными узлами древовидная топология оказывается недостаточно эффективной, поскольку сообщения должны проходить через один или несколько промежуточных звеньев. Очевидно, что на более высоких уровнях сети вероятность затора из-за недостаточно высокой пропускной способности линий связи выше. Этот 5

недостаток устраняют с помощью топологии, называемой «толстым» деревом (рисунок 12б).

Идея «толстого» дерева состоит в увеличении пропускной способности коммуникационных линий на прикорневых уровнях сети. С этой целью на верхних уровнях сети родительские и дочерние узлы связывают не одним, а несколькими каналами, причем чем выше уровень, тем больше число каналов. На рисунке это отображено в виде множественных линий между узлами верхних уровней. Топология «толстого» дерева реализована в вычислительной системе СМ-5.

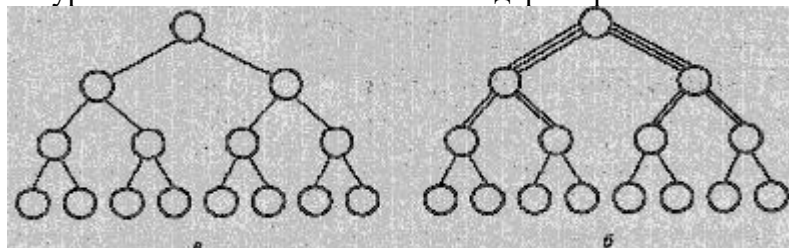


Рисунок 12 - Древоподобная топология: а — стандартное дерево; б — «толстое» дерево

### 5) Решетчатые топологии

Поскольку значительная часть научно-технических задач связана с обработкой массивов, вполне естественным представляется стремление учесть это и в топологии ВС, ориентированных на подобные задачи. Такие топологии относят к решетчатым (mesh), а их конфигурация определяется видом и размерностью массива.

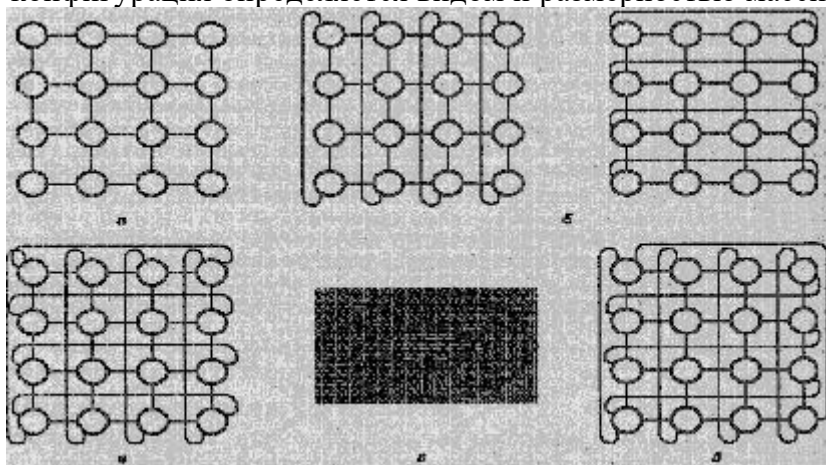


Рисунок 13 - Решетчатые топологии: а — плоская; б — цилиндрическая;  
в-г — тороидальная; д — витая тороидальная

Простейшими примерами для одномерных массивов могут служить цепочка и кольцо. Для двумерных массивов данных наиболее подходит топология плоской прямоугольной матрицы узлов, каждый из которых соединен с ближайшим соседом (рисунок 13а). Такая сеть размерности  $m \times m$  ( $m = \sqrt{N}$ ) имеет следующие характеристики:  $D = 2(m - 1)$ ;  $d = 4$ ;  $I = 2N - 2m$ ;  $B = m$ .

Если провести операцию свертывания (wraparound) плоской матрицы, соединив информационными трактами одноименные узлы левого и правого столбцов или одноименные узлы верхней и нижней строк плоской матрицы, то из плоской конструкции получаем топологию типа цилиндра (рисунок 13б). В топологии цилиндра каждый ряд (или столбец) матрицы представляет собой кольцо. Если одновременно произвести свертывание плоской матрицы в обоих направлениях, получим тороидальную топологию сети (рисунок 13в). Двумерный тор на базе решетки  $m \times m$  обладает следующими параметрами:  $D = 2 \min [m/2]$ ;  $d = 4$ ;  $I = 2N$ ;  $B = 2m$ .

Объемный вид тороидальной топологии для массива размерности  $4 \times 8$  показан на рисунке 13г. Помимо свертывания к плоской решетке может быть применена операция *скручивания* (twisting). Суть этой операции состоит в том, что вместо колец все узлы объединяются в разомкнутую или замкнутую спираль, то есть узлы, расположенные с противоположных краев плоской решетки, соединяются с некоторым сдвигом. Если горизонтальные петли объединены в виде спирали, образуется так называемая сеть типа ILLIAC. На рисунке 13д показана подобная конфигурация СМС, соответствующая хордальной сети четвертого порядка и характеризующаяся следующими метриками:  $D = m - 1$ ;  $d = 4$ ;  $I = 2N$ ;  $B = 2m$ .

Следует упомянуть и трехмерные сети. Один из вариантов, реализованный в архитектуре суперЭВМ Cray T3D, представляет собой трехмерный тор, образованный объединением

процессоров в кольца по трем координатам:  $x$ ,  $y$ ,  $z$ .

Примерами ВС, где реализованы различные варианты решетчатых топологий, могут служить: ILLIAC IV, MPP, DAP, CM-2, Paragon и др.

#### 6) Полносвязная топология

В *полносвязной топологии* (рисунок 14), известной также под названием топологии «максимальной группировки» или «топологии клика», каждый узел напрямую соединен со всеми остальными узлами сети. Сеть, состоящая из  $N$  узлов, имеет следующие параметры:  $D = 1$ ;  $d = N - 1$ ;  $I = (N*(N - 1))/2$ ;  $B = N^2/4$

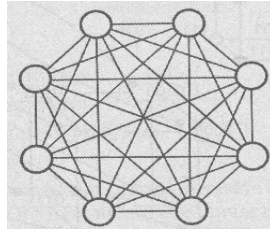


Рисунок 14 - Полносвязная топология

Если размер сети велик, топология становится дорогостоящей и трудно реализуемой. Более того, топология максимальной группировки не дает существенного улучшения производительности, поскольку каждая операция пересылки требует, чтобы узел проанализировал состояние всех своих  $N - 1$  входов. Для ускорения этой операции необходимо, чтобы все входы анализировались параллельно, что, в свою очередь, усложняет конструкцию узлов.

#### 7) Топология гиперкуба

При объединении параллельных процессоров весьма популярна топология гиперкуба, показанная на рисунке 15. Линия, соединяющая два узла (рисунок 15а), определяет одномерный гиперкуб. Квадрат, образованный четырьмя узлами (рисунок 15б) — двумерный гиперкуб, а куб из 8 узлов (рисунок 15в) — трехмерный гиперкуб и т. д. Из этого ряда следует алгоритм получения  $m$ -мерного гиперкуба: начинаем с  $(m - 1)$ -мерного гиперкуба, делаем его идентичную копию, а затем добавляем связи между каждым узлом исходного гиперкуба и одноименным узлом копии. Гиперкуб размерности  $m$  ( $N = 2^m$ ) имеет следующие характеристики:  $D = m$ ;  $d = m$ ;  $I = mN/2$

Увеличение размерности гиперкуба на 1 ведет к удвоению числа его узлов, увеличению порядка узлов и диаметра сети на единицу.

Обмен сообщениями в гиперкубе базируется на двоичном представлении номеров узлов. Нумерация узлов производится так, что для любой пары смежных узлов двоичное представление номеров этих узлов отличается только в одной позиции. В силу сказанного, узлы 0010 и 0110 — соседи, а узлы 0110 и 0101 таковыми не являются. Простейший способ нумерации узлов при создании  $m$ -мерного гиперкуба из двух  $(m - 1)$ -мерных показан на рисунке 15д. При копировании  $(m - 1)$ -мерного гиперкуба и соединении его с исходным  $(m - 1)$ -мерным гиперкубом необходимо, чтобы соединяемые узлы имели одинаковые номера. Далее к номерам узлов исходного гиперкуба слева добавляется бит, равный 0, а к номерам узлов копии — единичный бит.

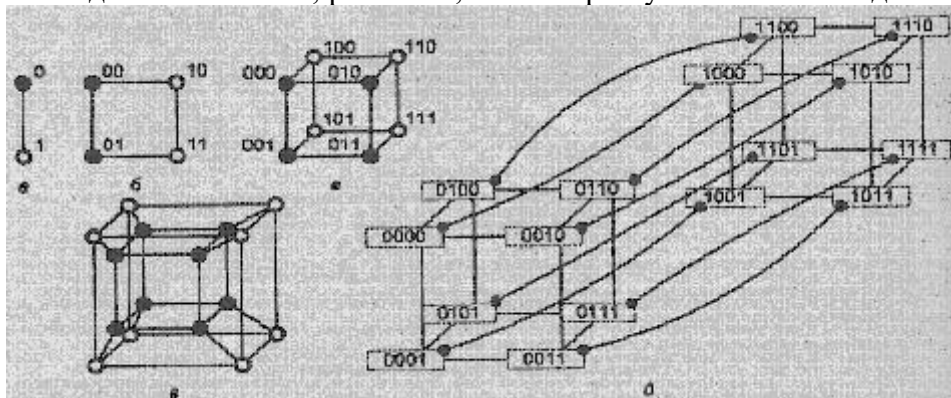


Рисунок 15 - Топология гиперкуба: а — одномерная; б — двумерная; в — трехмерная; г — четырехмерная; д — схема формирования четырехмерного гиперкуба из двух трехмерных

Номера узлов являются основой маршрутизации сообщений в гиперкубе. Такие номера в  $m$ -мерном гиперкубе состоят из  $m$  битов, а пересылка сообщения из узла  $A$  в узел  $B$  выполняется за  $m$  шагов. На каждом шаге узел может либо сохранить сообщение и не пересылать его дальше до следующего шага, либо отправить его дальше по одной из линий. На шаге  $i$  узел, хранящий сообщение, сравнивает  $i$ -й бит своего собственного номера с  $i$ -м битом номера узла назначения. Если они совпадают, продвижение сообщения прекращается, если нет — сообщение передается

вдоль линии  $i$ -го измерения. Линией  $i$ -го измерения считается та, которая была добавлена на этапе построения  $i$ -мерного гиперкуба из двух  $(i - 1)$ -мерных.

Создание гиперкуба при большом числе процессоров требует увеличения порядка узлов, что сопряжено с большими техническими проблемами. Компромиссное решение, несколько увеличивающее диаметр сети при сохранении базовой структуры, представляет собой куб из циклически соединенных узлов (рисунок 16). Здесь порядок узла равен трем при любом размере сети.

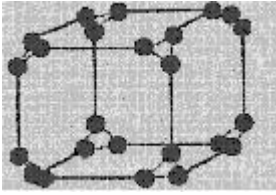


Рисунок 16 - Куб из циклически соединенных узлов третьего порядка

#### 8) Топология $k$ -ичного $n$ -куба

Название топологии означает, что в ней реализуется куб, имеющий  $n$  измерений, причем каждое измерение содержит  $k$  узлов ( $N = k^n$ ). Каждому узлу назначен  $n$ -разрядный номер в системе счисления с основанием  $k$ , и он связан с узлом, номер которого отличается только в одной цифре и только на единицу.  $k$ -ичный  $n$ -куб может быть построен путем объединения  $k$  экземпляров  $k$ -ичных  $(n - 1)$ -кубов в кольцо.

Многие ранее рассмотренные топологии представляют собой варианты топологии  $k$ -ичного  $n$ -куба:

- $k$ -ичный 1-куб — кольцо;
- $k$ -ичный 2-куб — двумерный тор;
- $k$ -ичный 3-куб — трехмерный тор;
- 4-ичный 2-куб — плоская решетка  $4 \times 4$ ;
- 2-ичный  $n$ -куб — гиперкуб.

Доказано, что эффективность топологии, а также ее масштабируемость улучшаются с ростом значения  $k$  и уменьшением количества измерений  $n$ .