

# Коммутируемые сети

## Ethernet

Современные коммутаторы Ethernet являются наследниками мостов локальных сетей, которые широко использовались в сетях Ethernet и Token Ring на разделяемой среде. Более того, коммутаторы

Ethernet по-прежнему функционально очень близки к вышедшим из употребления мостам, так как базовый алгоритм работы коммутатора и моста является одним и тем же алгоритмом и определяется одним и тем же стандартом **IEEE 802.1D**. По традиции во всех новых стандартах IEEE, описывающих свойства коммутаторов, употребляется термин «**коммутатор**», а не «**мост**». Основное отличие коммутатора от моста состоит в большем количестве портов (мост, как правило, имел два порта, что и послужило поводом для его названия — мост между двумя сегментами) и более высокой производительности.

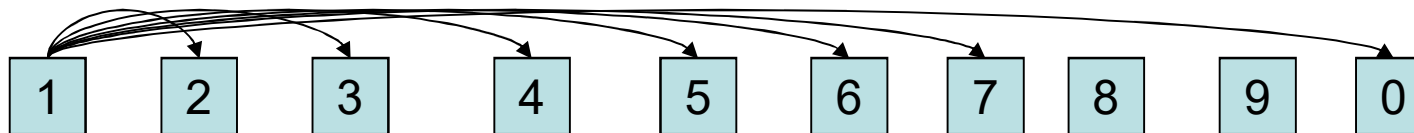
# Сеть с общей разделяемой средой не обеспечивает масштабируемости

- В сети Ethernet требование использовать единую разделяемую среду приводит к нескольким очень жестким ограничениям:
  - общий диаметр сети не может быть больше 2500 м;
  - количество узлов не может превышать 1024 (для сетей Ethernet на коаксиале это ограничение еще жестче).
- На практике из-за главной проблемы разделяемой среды — дефицита пропускной способности — количество узлов даже в сетях 10Base-T и 10Base-F никогда не приближается к 1024.

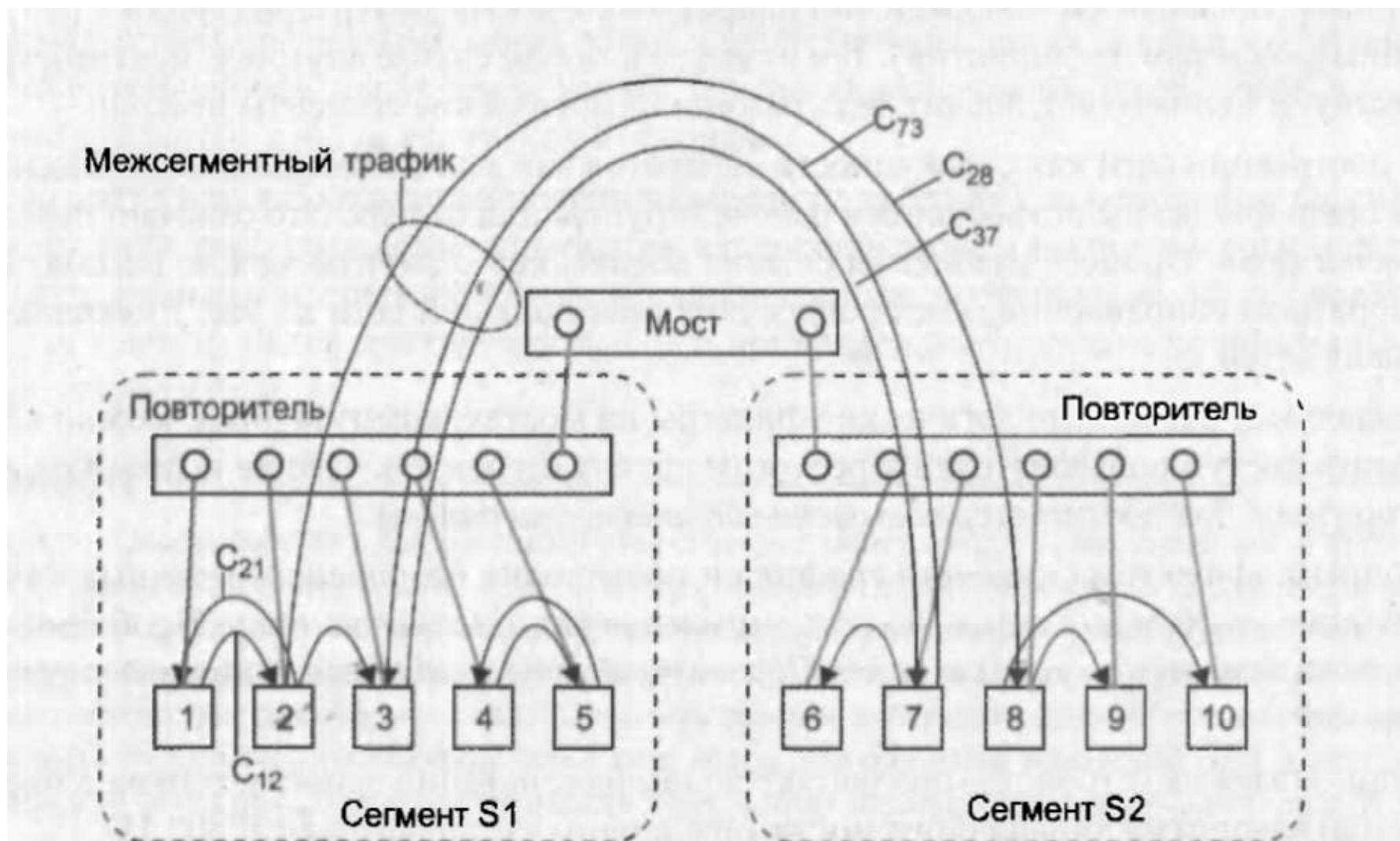
# Сегментация сети

- Ограничения, возникающие из-за использования единой разделяемой среды, можно преодолеть, выполнив *логическую структуризацию сети*, то есть сегментировав единую разделяемую среду на несколько и соединив полученные сегменты сети некоторым коммуникационным устройством, которое не передает данные побитно, как повторитель, а буферизует кадры и передает их затем в тот или иной сегмент (или сегменты) в зависимости от адреса назначения кадра.

## Сеть с общей разделяемой средой без сегментирования



Интенсивность трафика  
 $(T1=10)*9=90$



Интенсивность трафика с сегментированием.

Внутрисегментный трафик  $(T=4)*5 = 20$

Межсегментный трафик:  $5*5+5*5=50$

$20+50 = 70$ , а было 90... интенсивность уменьшилась на внутрисегментную интенсивность трафика. Таким образом снижается трафик внутри сегмента.

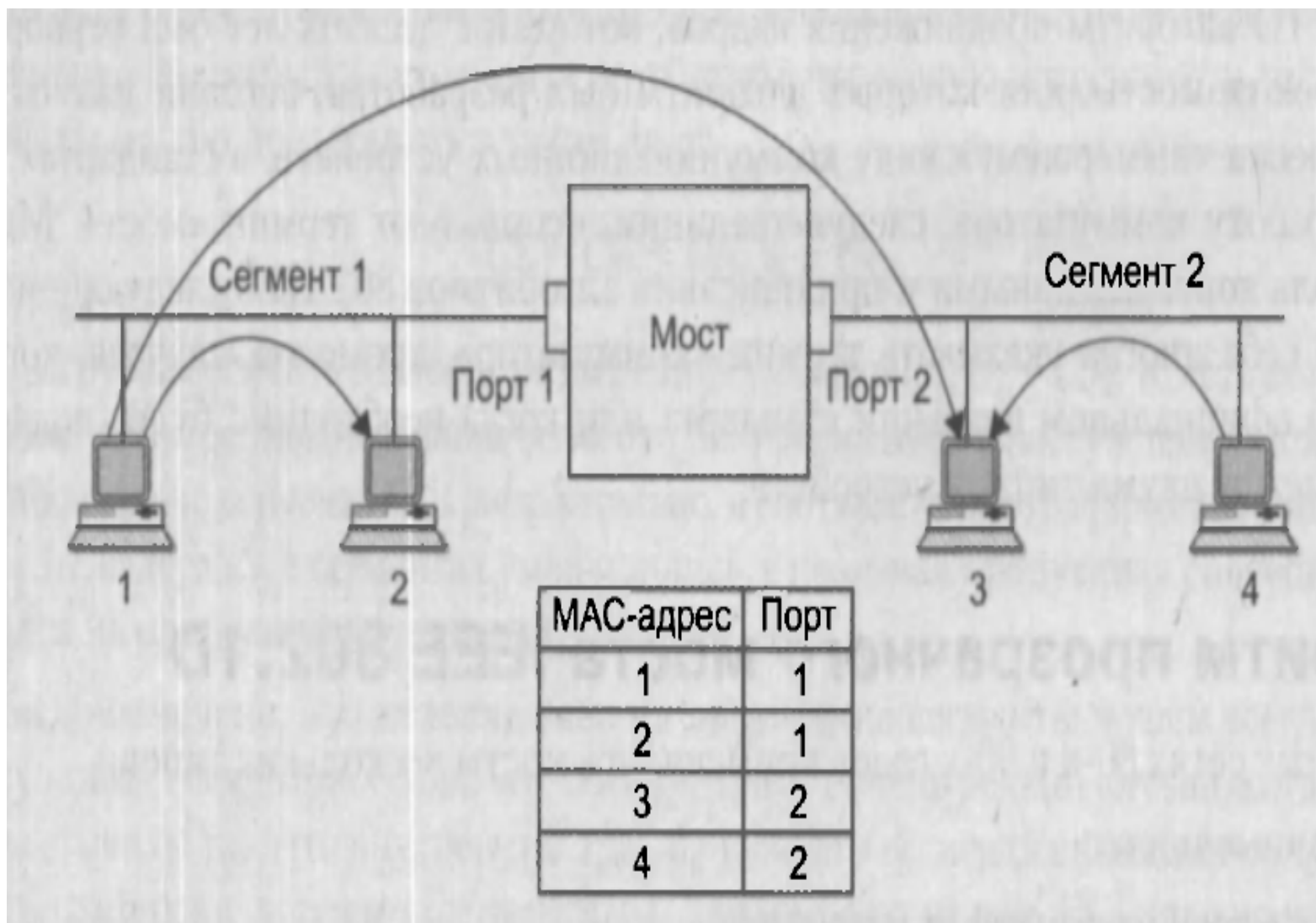
В локальных сетях 80-х и 90-х годов применялись мосты нескольких типов:

- прозрачные мосты;
- мосты с маршрутизацией от источника;
- транслирующие мосты.

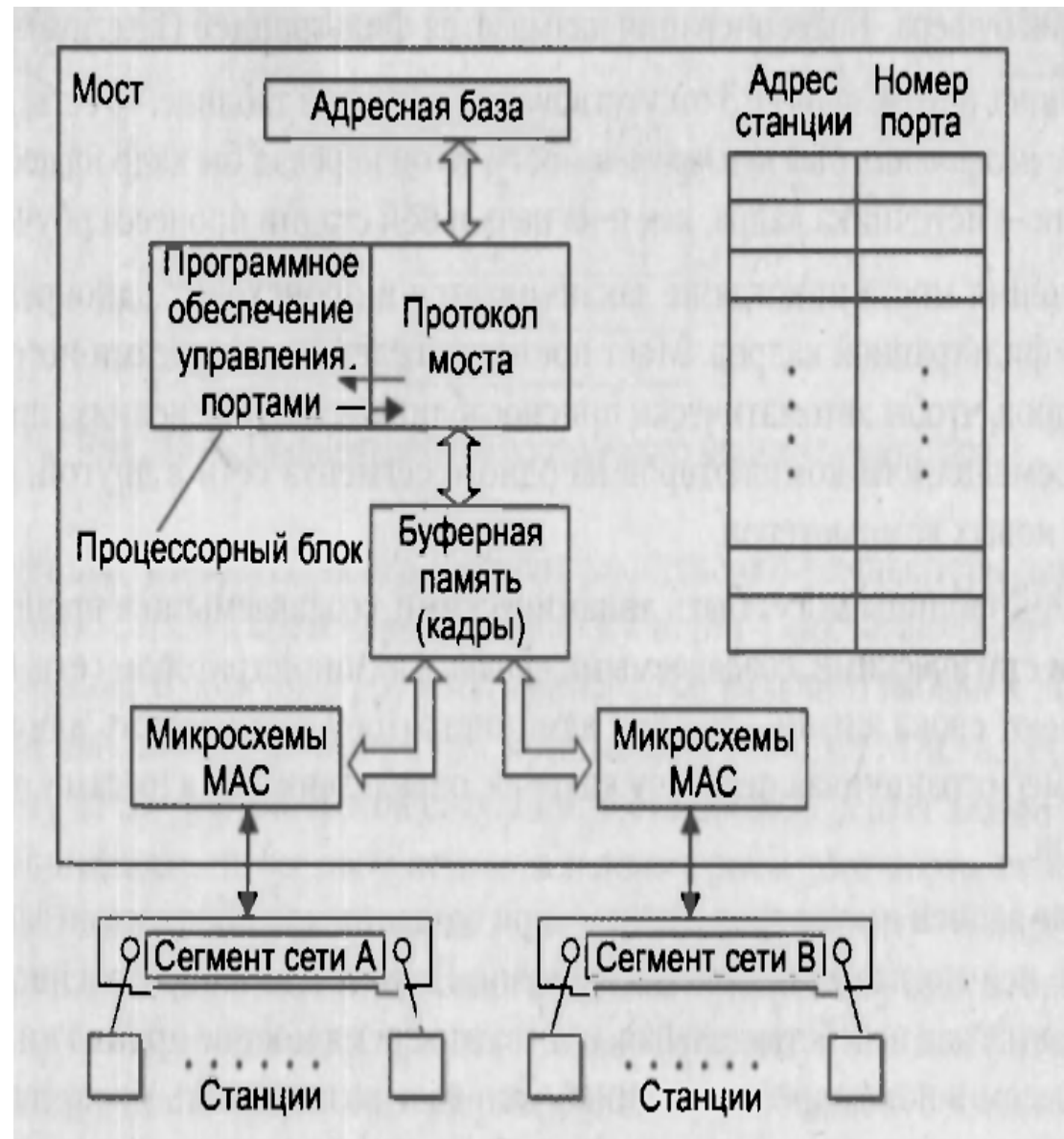
Мосты с маршрутизацией от источника применялись только в сетях Token Ring, а транслирующие мосты были способны соединять сегменты разных технологий, например Ethernet и Token Ring. В результате исчезновения всех технологий локальных сетей, кроме Ethernet, оба этих типа мостов также исчезли, а алгоритм прозрачного моста выжил, найдя свое применение в коммутаторах Ethernet.

Слово «прозрачный» в названии *алгоритм прозрачного моста* отражает тот факт, что мосты и коммутаторы в своей работе не учитывают существование в сети сетевых адаптеров конечных узлов, концентраторов и повторителей. В то же время и перечисленные сетевые устройства функционируют, «не замечая» присутствия в сети мостов и коммутаторов.

Мост строит свою таблицу продвижения (адресную таблицу) на основании пассивного наблюдения за трафиком, циркулирующим в подключенных к его портам сегментах. При этом мост учитывает адреса источников кадров данных, поступающих на его порты. По адресу источника кадра мост делает вывод о принадлежности узла-источника тому или иному сегменту сети.

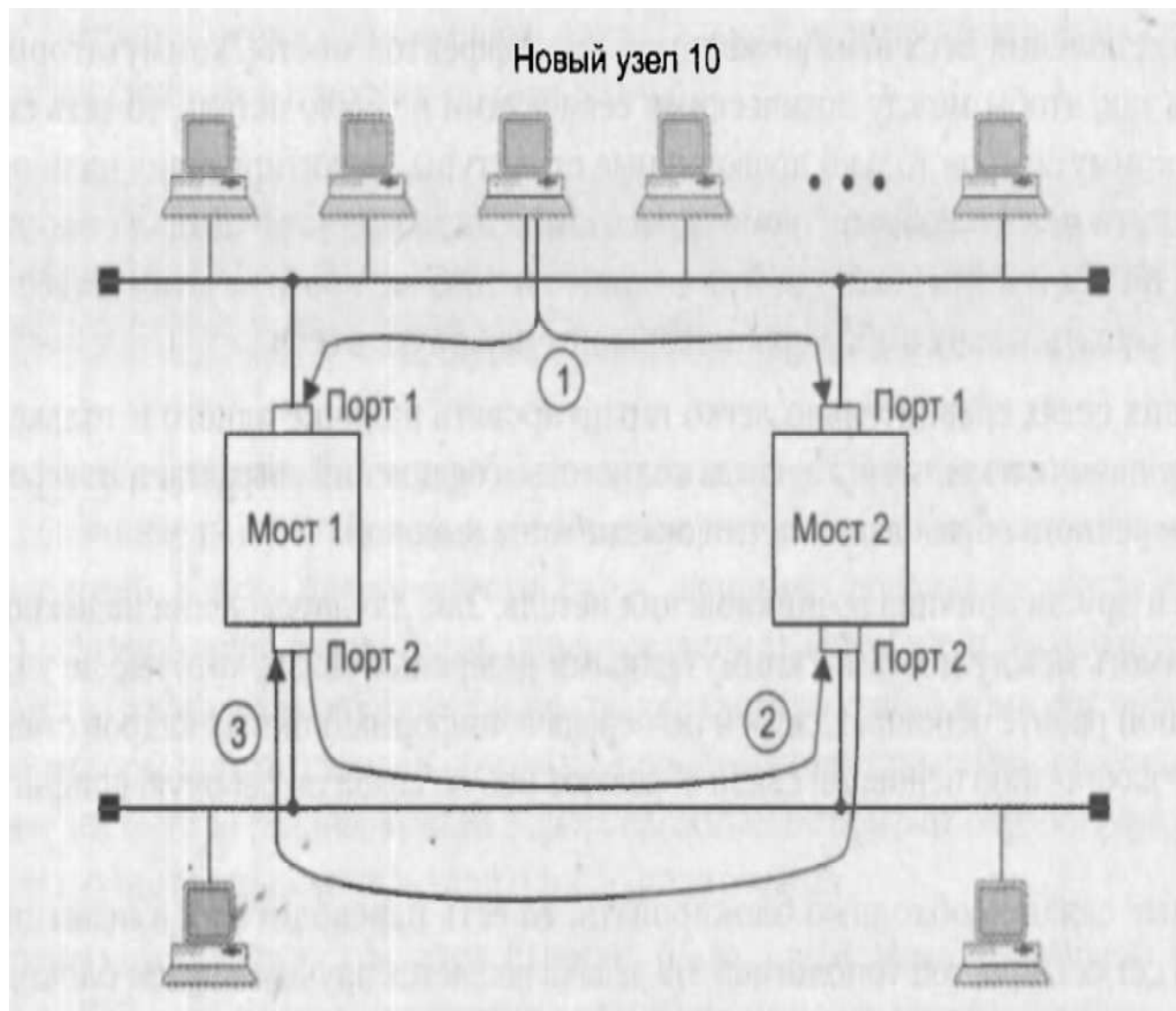


# Структура моста, коммутатора

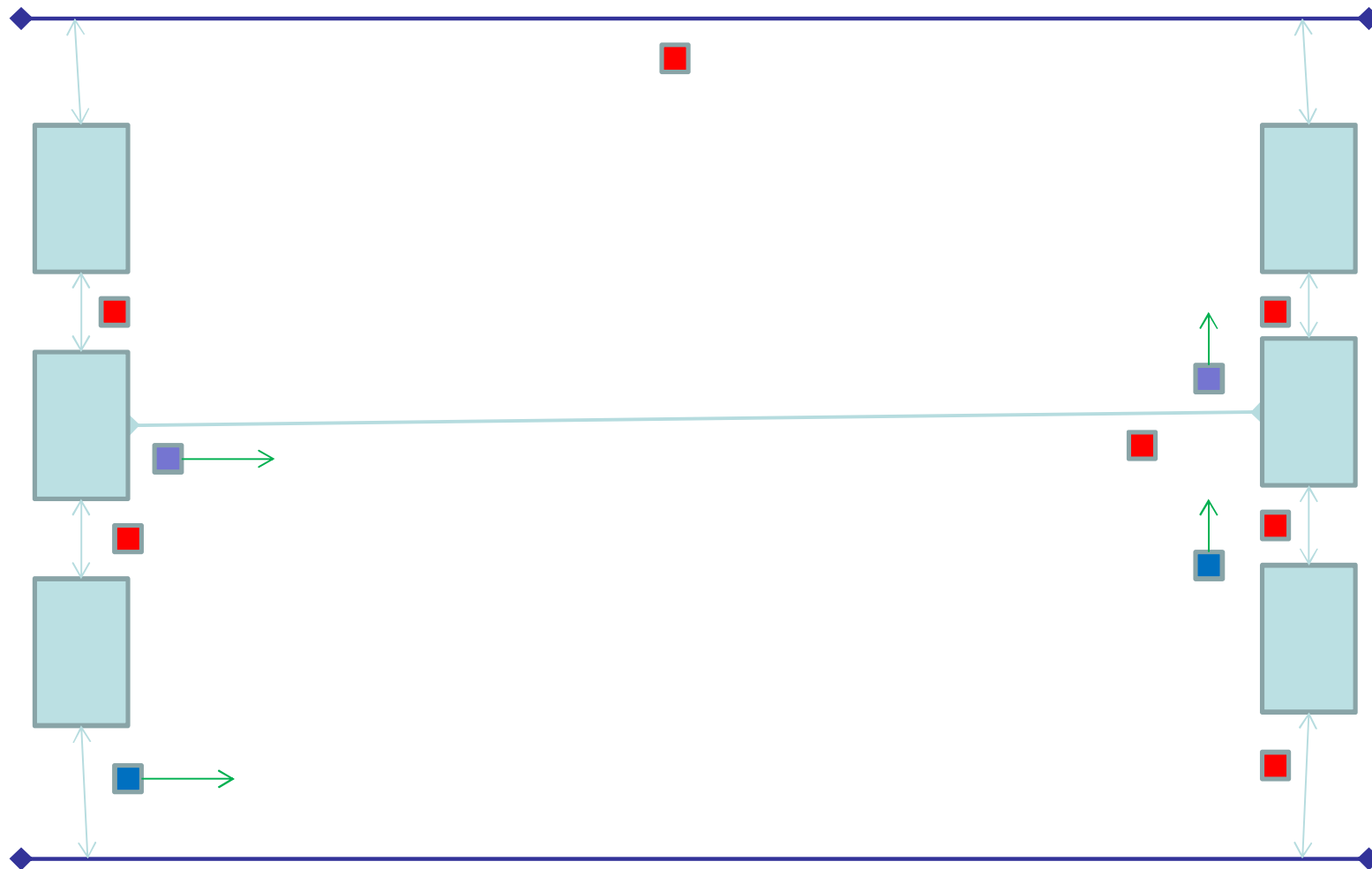




Серьезным ограничением функциональных возможностей мостов и коммутаторов является отсутствие поддержки петлеобразных конфигураций сети.



# Широковещательный шторм



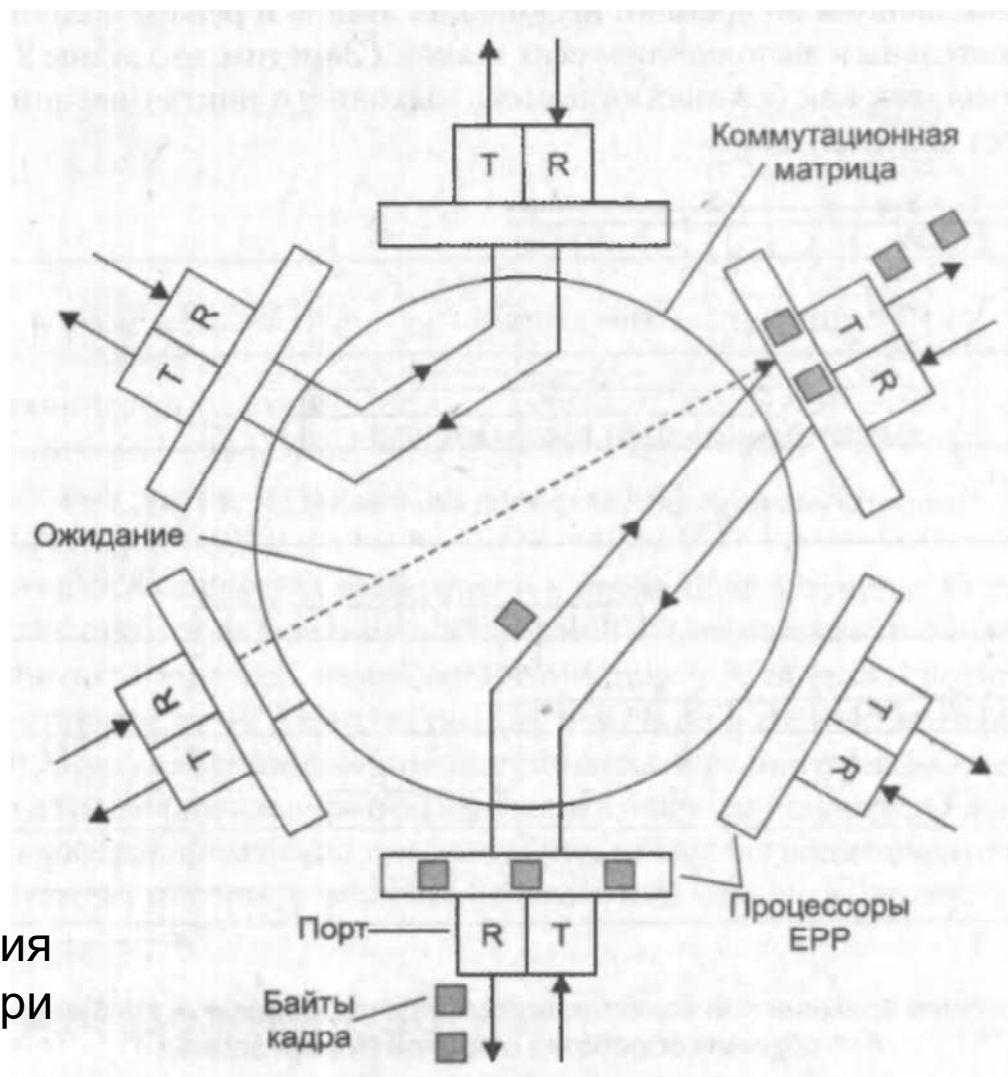
# ПОСЛЕДСТВИЯ НАЛИЧИЯ ПЕТЛИ В СЕТИ

- • «Размножение» кадра, то есть появление нескольких его копий (в данном случае — двух, но если бы сегменты были соединены тремя мостами — то трех и т. д.).
- • Бесконечная циркуляция обеих копий кадра по петле в противоположных направлениях, а значит, засорение сети ненужным трафиком.
- • Постоянная перестройка мостами своих адресных таблиц
- В целях исключения всех этих нежелательных эффектов мосты/коммутаторы нужно применять так, чтобы между логическими сегментами не было петель, то есть строить с помощью коммутаторов только древовидные структуры, гарантирующие наличие единственного пути между любыми двумя сегментами.

# Параллельная коммутация

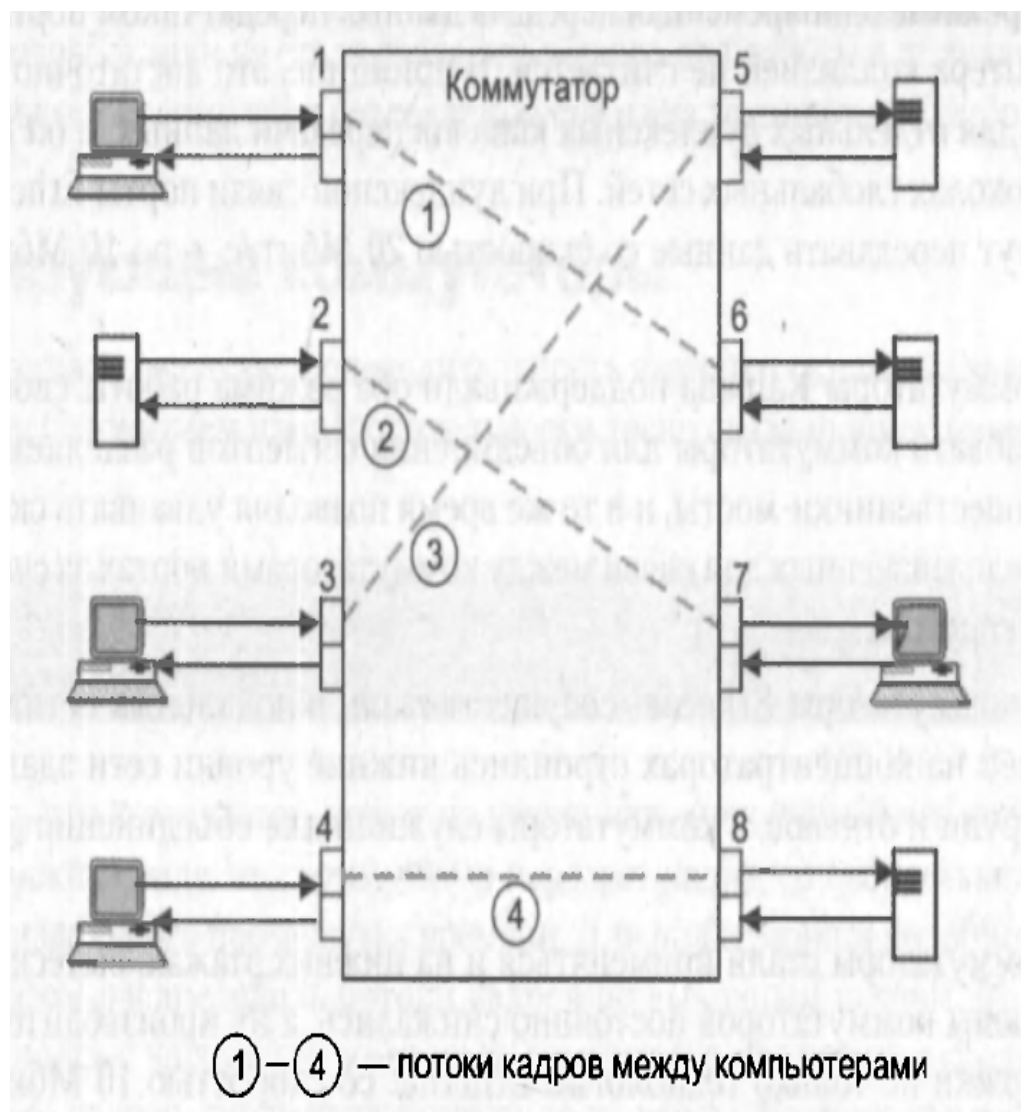
- Обслуживание потоков кадров между теперь уже несколькими портами с помощью одного процессорного блока требовало значительного повышения быстродействия процессора, а это довольно дорогостоящее решение.
- Более эффективным оказалось решение, которое и «породило» коммутаторы: для обслуживания потока, поступающего на каждый порт, в устройство ставился отдельный специализированный процессор, который реализовывал алгоритм прозрачного моста. По сути, **коммутатор — это мультипроцессорный мост**, способный параллельно продвигать кадры сразу между всеми парами своих портов.

# Структурная схема коммутатора EtherSwitch, предложенного фирмой Kalpana (EPP - (Ethernet Packet Processor))



главной причиной повышения производительности сети при использовании коммутатора является *параллельная* обработка нескольких кадров.

## Параллельная передача кадров коммутатором



# Алгоритм

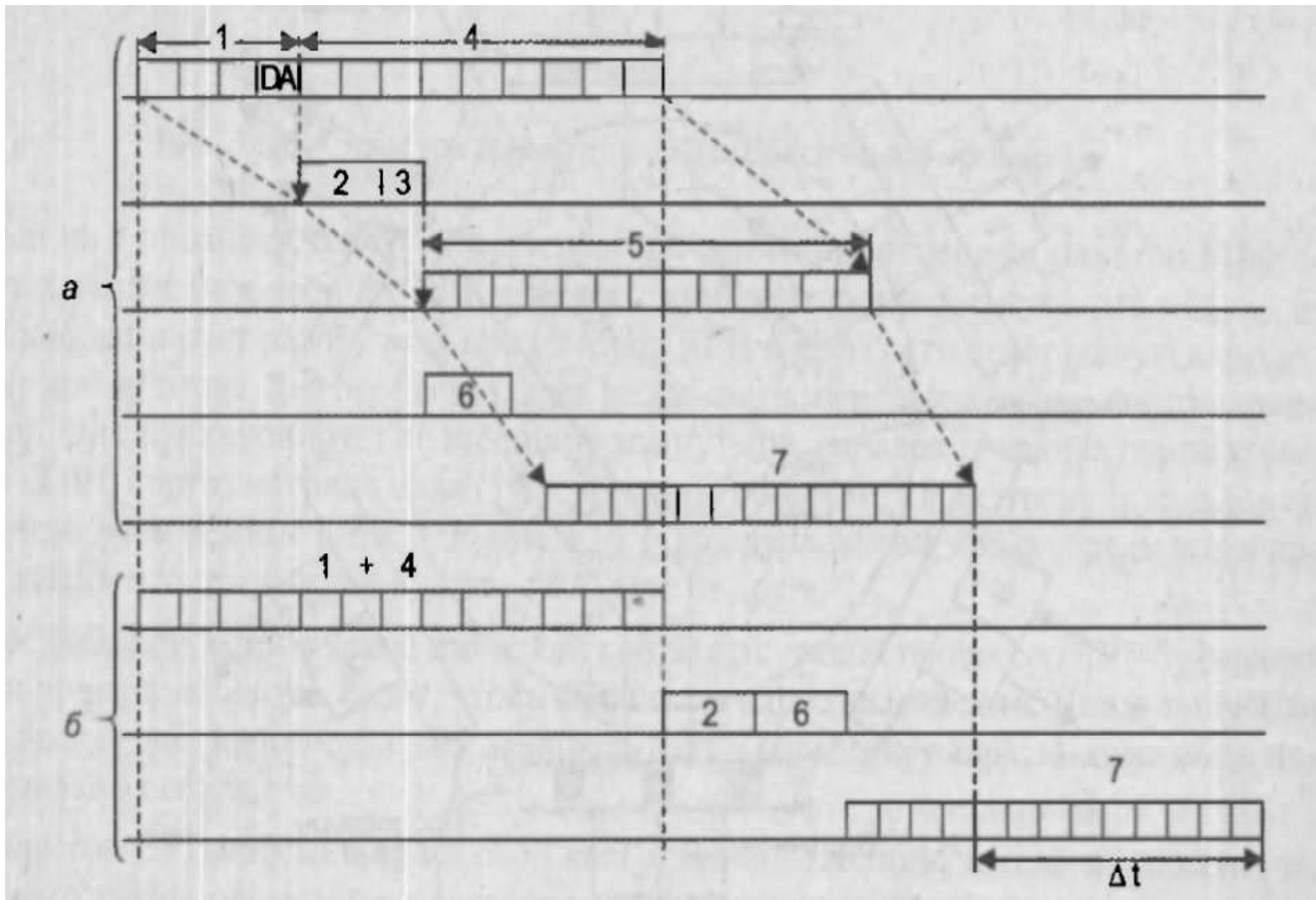
- 1. Процессор **ERP** просматривает свой кэш адресной таблицы, и если не находит там нужного адреса, обращается к системному модулю, который работает в многозадачном режиме, параллельно обслуживая запросы всех процессоров **ERP**. Системный модуль производит просмотр общей адресной таблицы и возвращает процессору найденную строку, которую тот буферизует в своем кэше для последующего использования.
- 2. Если адрес назначения найден в адресной таблице и кадр нужно отфильтровать, процессор просто прекращает записывать в буфер байты кадра, очищает буфер и ждет поступления нового кадра.
- 3. Если же адрес найден и кадр нужно передать на другой порт, процессор, продолжая прием кадра в буфер, обращается к коммутационной матрице, пытаясь установить в ней путь, связывающий его порт с портом, через который идет маршрут к адресу назначения. Коммутационная матрица способна помочь только в том случае, если порт адреса назначения в этот момент свободен, то есть не соединен с другим портом данного коммутатора.

- 4. Если же порт занят, то, как и в любом устройстве с коммутацией каналов, матрица в соединении отказывает. В этом случае кадр полностью буферизуется процессором входного порта, после чего процессор ожидает освобождения выходного порта и образования коммутационной матрицей нужного пути.
- 5. После того как нужный путь установлен, в него направляются буферизованные байты кадра, которые принимаются процессором выходного порта. Как только процессор выходного порта получает доступ к подключенному к нему сегменту Ethernet по алгоритму CSMA/CD, байты кадра сразу же начинают передаваться в сеть. Процессор входного порта постоянно хранит несколько байтов принимаемого кадра в своем буфере, что позволяет ему независимо и асинхронно принимать и передавать байты кадра



*конвейерная обработка кадра*, когда частично совмещаются во времени несколько этапов его передачи.

- 1. Прием первых байтов кадра процессором входного порта, включая прием байтов адреса назначения.
- 2. Поиск адреса назначения в адресной таблице коммутатора (в кэше процессора или в общей таблице системного модуля).
- 3. Коммутация матрицы.
- 4. Прием остальных байтов кадра процессором входного порта.
- 5. Прием байтов кадра (включая первые) процессором выходного порта через коммутационную матрицу.
- 6. Получение доступа к среде процессором выходного порта.
- 7. Передача байтов кадра процессором выходного порта в сеть.



Экономия времени при конвейерной обработке кадра: а — конвейерная обработка, б — обычная обработка с полной буферизацией

# Полудуплексный и дуплексный режим

- В **полудуплексном режиме** работы порт коммутатора по-прежнему распознает коллизии. Доменом коллизий в этом случае является участок сети, включающий передатчик коммутатора, приемник коммутатора, передатчик сетевого адаптера компьютера, приемник сетевого адаптера компьютера и две витые пары, соединяющие передатчики с приемниками. Коллизия возникает, когда передатчики порта коммутатора и сетевого адаптера одновременно или почти одновременно начинают передачу своих кадров.
- В **дуплексном режиме** одновременная передача данных передатчиком порта коммутатора и сетевого адаптера коллизией не считается. В принципе, это достаточно естественный режим работы для отдельных дуплексных каналов передачи данных, и он всегда использовался в протоколах глобальных сетей. При дуплексной связи порты Ethernet стандарта 10 Мбит/с могут передавать данные со скоростью 20 Мбит/с — по 10 Мбит/с в каждом направлении.

# Архитектура коммутаторов

- Для ускорения операций коммутации сегодня во всех коммутаторах используются заказные специализированные **БИС — ASIC**, которые оптимизированы для выполнения основных операций коммутации. Часто в одном коммутаторе имеется несколько специализированных БИС, каждая из которых выполняет функционально законченную часть операций.

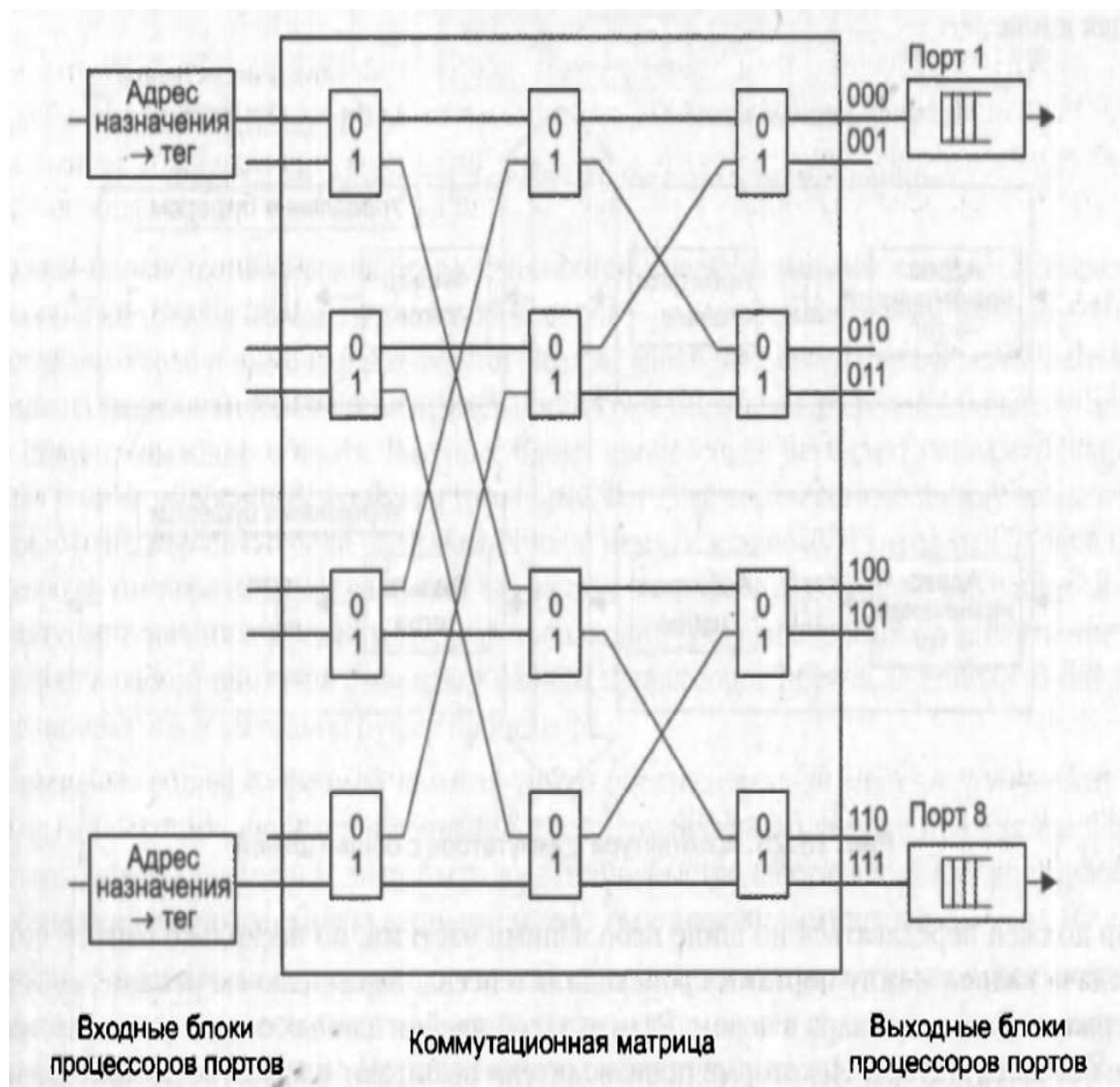
- Важную роль в построении коммутаторов играют также программируемые микросхемы **FPGA** (**Field-Programmable Gate Array** — программируемый в условиях эксплуатации массив вентилей). Эти микросхемы могут выполнять все функции, которые выполняют микросхемы **ASIC**, но в отличие от последних эти функции могут программироваться и перепрограммироваться производителями коммутаторов (и даже пользователями). Это свойство позволило резко удешевить процессоры портов коммутаторов, выполняющих сложные операции, например профилирование трафика, так как производитель **FPGA** выпускает свои микросхемы массово, а не по заказу того или иного производителя оборудования. Кроме того, применение микросхем **FPGA** позволяет производителям коммутаторов оперативно вносить изменения в логику работы порта при появлении новых стандартов или изменении действующих.

- Помимо процессорных микросхем для успешной неблокирующей работы коммутатору нужно иметь быстродействующий *узел обмена*, предназначенный для передачи кадров между процессорными микросхемами портов.
- В настоящее время в коммутаторах узел обмена строится на основе одной из трех схем:
  - коммутационная матрица;
  - общая шина;
  - разделяемая многоходовая память.
- Часто эти три схемы комбинируются в одном коммутаторе.

## Коммутационная матрица

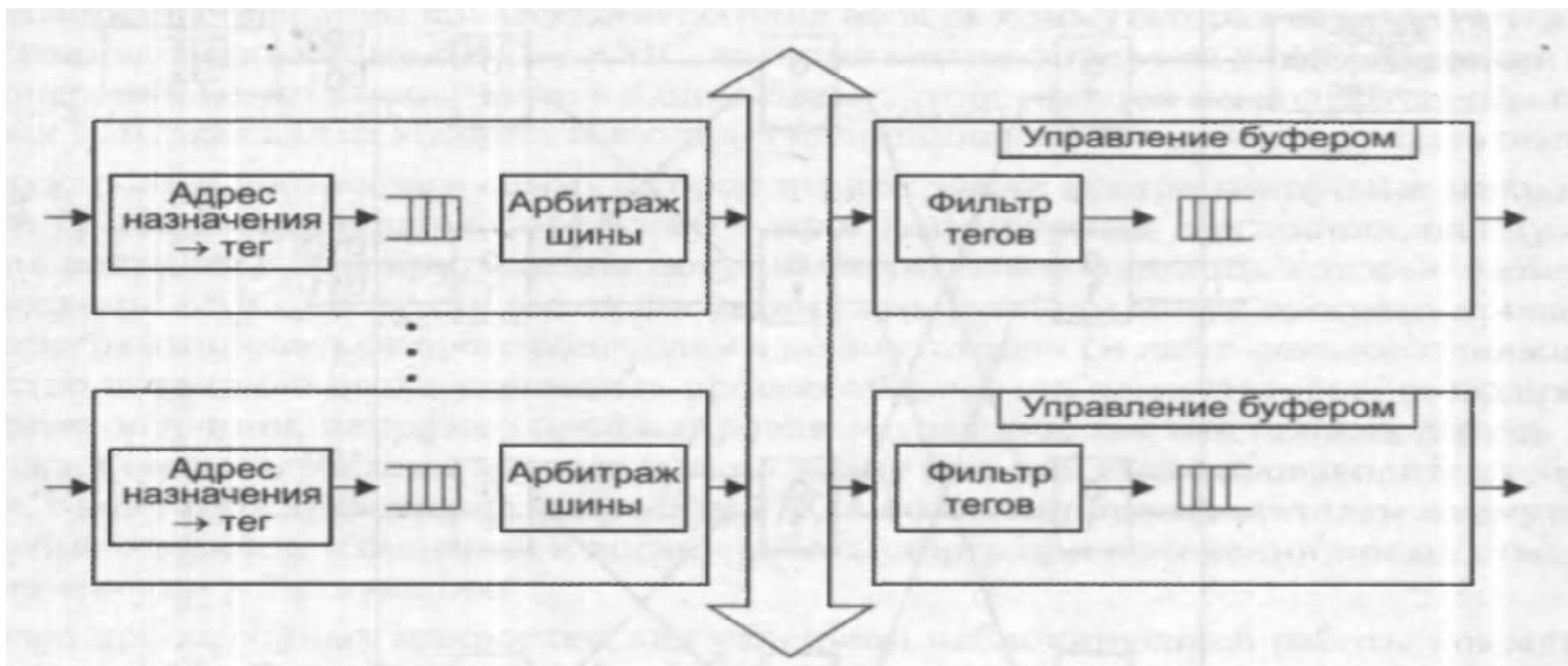
Входные блоки процессоров портов на основании просмотра адресной таблицы коммутатора определяют по адресу назначения номер выходного порта. Эту информацию они добавляют к байтам исходного кадра в виде специального ярлыка — тега

Матрица состоит из трех уровней двоичных переключателей, которые соединяют свой вход с одним из двух выходов в зависимости от значения бита тега. Переключатели первого уровня управляются первым битом тега, второго — вторым, а третьего — третьим.

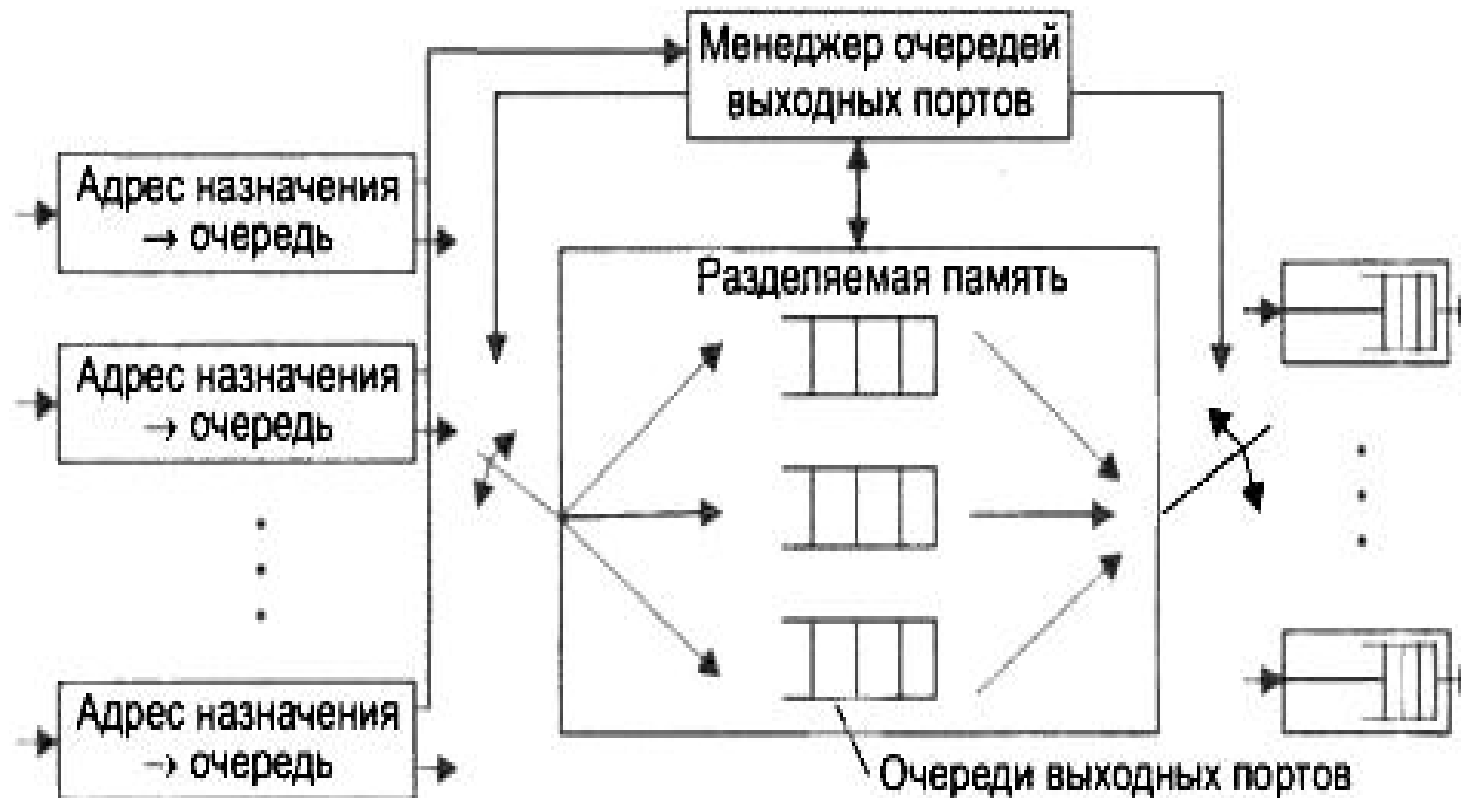




- **В коммутаторах с общей шиной** процессоры портов связывают высокоскоростной шиной, используемой в режиме разделения времени. Кадр должен передаваться по шине небольшими частями, по несколько байтов, чтобы передача кадров между портами происходила в псевдопараллельном режиме, не внося задержек в передачу кадра в целом. Размер такой ячейки данных определяется производителем коммутатора-. Некоторые производители выбирают в качестве порции данных, переносимых по шине за одну операцию, ячейку АТМ с ее полем данных в 48 байт.



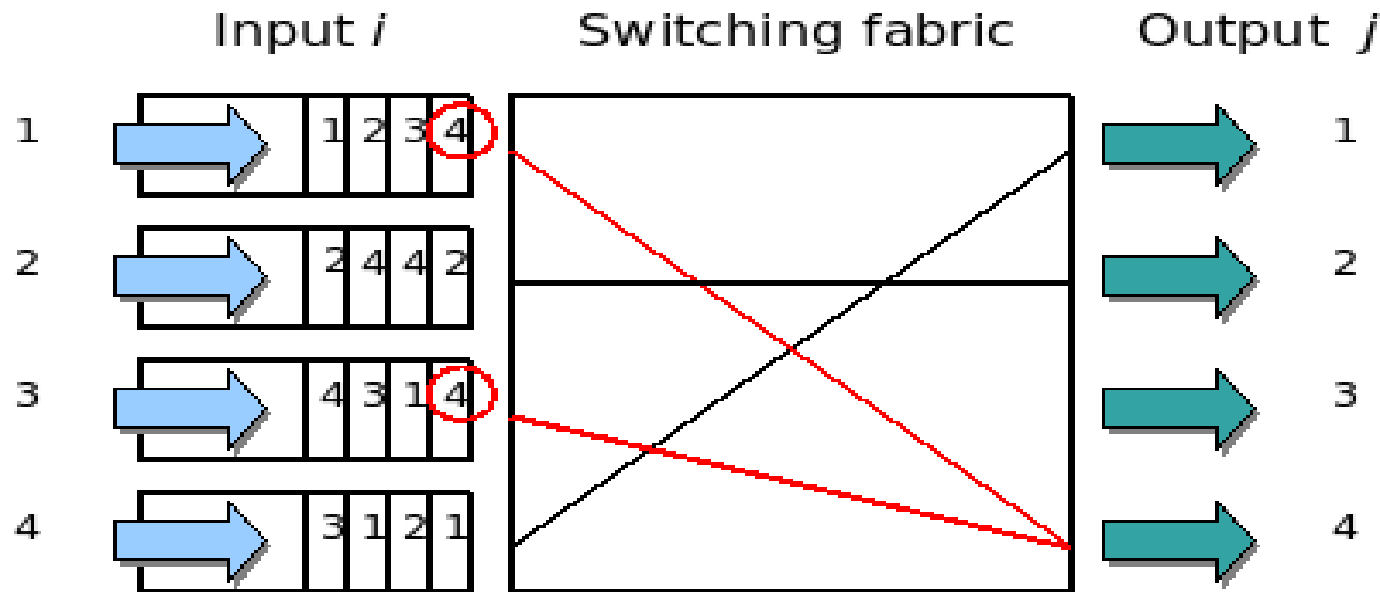
# Архитектура коммутаторов с разделяемой памятью



- Входные блоки процессоров портов соединяются с переключаемым входом разделяемой памяти, а выходные блоки этих же процессоров — с ее переключаемым выходом. Переключением входа и выхода разделяемой памяти управляет менеджер очередей выходных портов. В разделяемой памяти менеджер организует несколько очередей данных, по одной для каждого выходного порта. Входные блоки процессоров передают менеджеру портов запросы на запись данных в очередь того порта, который соответствует адресу назначения кадра. Менеджер по очереди подключает вход памяти к одному из входных блоков процессоров и тот переписывает часть данных кадра в очередь определенного выходного порта.

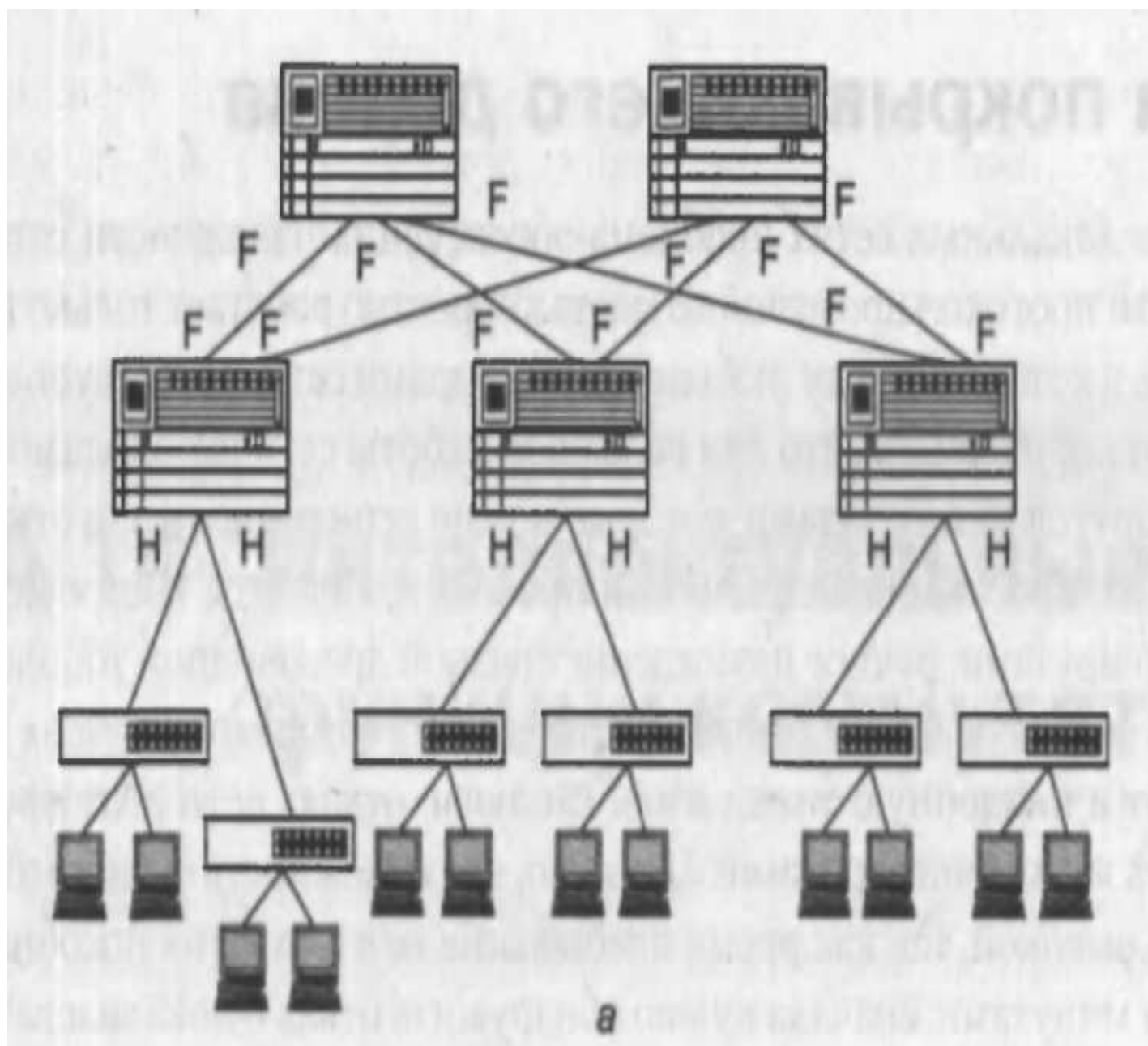
- По мере заполнения очередей менеджер производит также поочередное подключение выхода разделяемой памяти к выходным блокам процессоров портов, и данные из очереди переписываются в выходной буфер процессора.
- Применение общей буферной памяти, гибко распределяемой менеджером между отдельными портами, снижает требования к размеру буферной памяти процессора порта. Однако буферная память должна быть достаточно быстродействующей для поддержания необходимой скорости обмена данными между портами коммутатора.

# Head-of-Line blocking, HOL blocking



- Выходное направление 4 блокирует (из-за конфликта первого и четвертого входа) открытый 3 выход для кадра идущего с первого входа вслед за блокируемым.

- Для автоматического нахождения и конфигурирования активной древовидной топологии, мониторинга состояния ее связей и перехода к новой древовидной топологии при обнаружении отказа связи в коммутируемых локальных сетях используются **алгоритм покрывающего дерева (Spanning Tree Algorithm, STA)** и реализующий его **протокол покрывающего дерева (Spanning Tree Protocol, STP)**.
- **RSTP — rapid**, простой STP находит новую топологию дерева за 50 сек, быстрый за несколько секунд.



# термины

- Для нахождения покрывающего оптимального дерева используются метрики обычно в величинах обратных пропускной способности.
- **Идентификатор коммутатора** — 8 байтовый, младшая часть мак адрес, старшая 2 байта — приоритет — позволяет влиять на выбор корневого коммутатора.
- **Идентифкатор порта** — 2 байта, младший байт номер порта, старший приоритет.
- Есть также **Корневой порт**.
- **Назначенный коммутатор сегмента** — коммутатор сегмента с минимальным расстоянием до корневого коммутатора.
- **BPDU — bridge protocol data unit**, данные переносимые по сети между коммутаторами о портах, идентификаторах, включая кадр Hello пересылаемый в интервалах от 1 до 4 с.



## Алгоритм STA

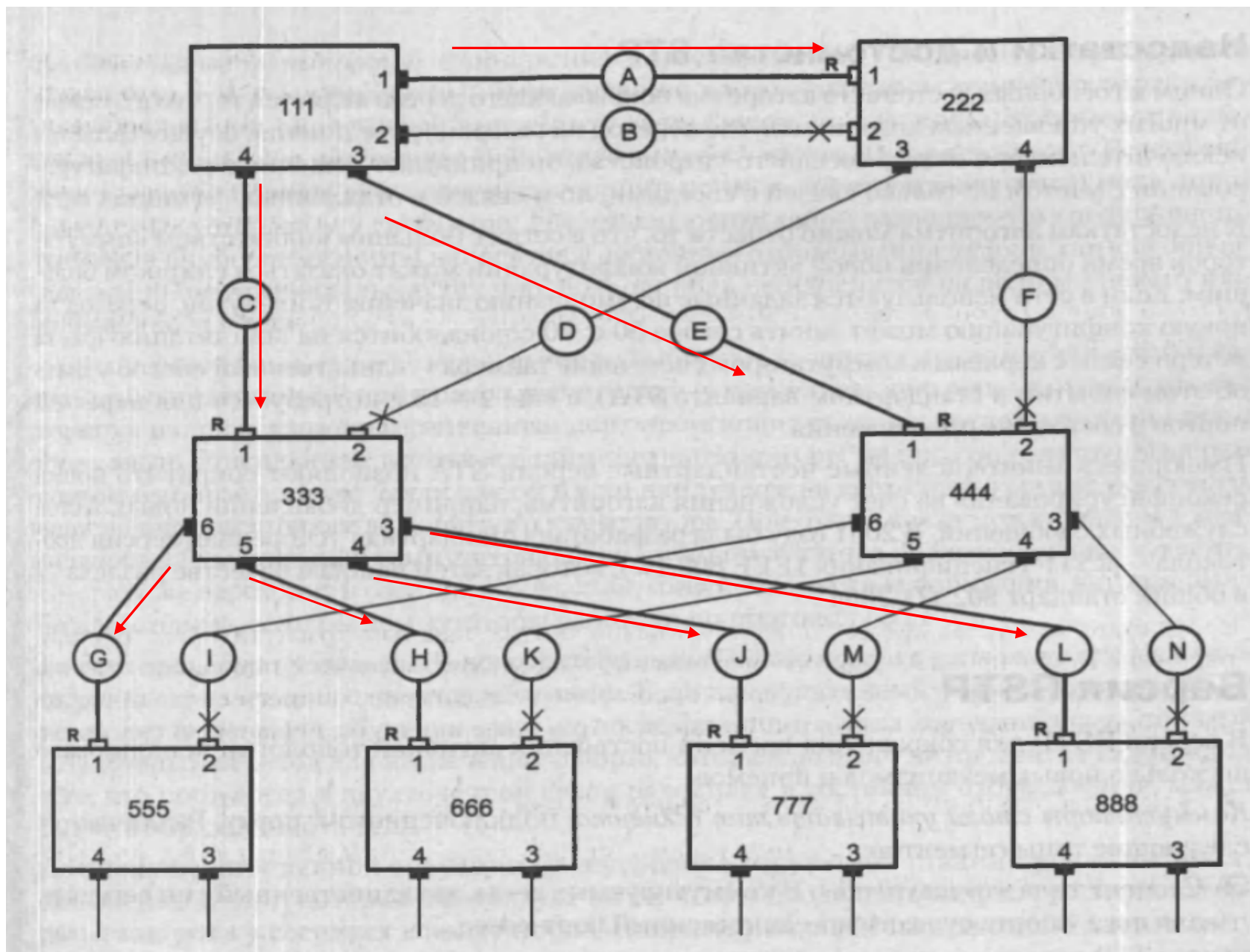
- Построение дерева в три этапа.
- Первый этап — **определение корневого коммутатора** путем рассылки сообщений Hello друг другу, в качестве корневого становится коммутатор с **минимальным номером** идентификатора, на выбор может влиять приоритет, так как изначально по умолчанию задаются старшие байты для всех коммутаторов как число 32778, можно увеличить номер идентификатора за счет добавления младших единичных бит в приоритет, тем самым уменьшив его приоритет.
- Коммутатор получив пакет hello с указанным там меньшим номером идентификатора становится просто ретранслятором пакетов hello и свои пакеты больше не генерирует, таким образом в итоге определяется корневой продолжая слать hello.

# Определение корневого порта

- Второй этап — определение корневого порта на каждом коммутаторе, порта с кратчайшим путем от корневого коммутатора (корневой коммутатор не имеет корневых портов). Когда рассылаются пакеты hello от корневого коммутатора, то каждый коммутатор получая hello накапливает через свой порт расстояние до соседнего коммутатора от которого пришло hello (метрику сегмента). Таким образом, в пакете хелло будет храниться расстояние до корневого через данный порт. Затем выбирается порт с кратчайшим расстоянием в качестве корневого порта. При равенстве расстояний, корневым становится порт с наименьшим идентификатором.

# Определение назначенного коммутатора и порта

- Третий этап — определение назначенного коммутатора и порта для данного сегмента, это коммутатор и порт с кратчайшим расстоянием от данного сегмента до корневого коммутатора.
- Все остальные порты, кроме корневых и назначенных, каждым коммутатором блокируются и не могут передавать пользовательские кадры. Математически доказано, что при таком выборе активных портов в сети исключаются петли, а оставшиеся связи образуют покрывающее дерево (если оно вообще может быть построено при существующих связях в сети).



корневые порты коммутаторов отмечены символом R, назначенные порты закрашены, а заблокированные зачеркнуты.

- После построения покрывающего дерева коммутатор начинает принимать (но не продвигать) пакеты данных и **на основе их адресов источника строить таблицу продвижения**. Это обычный режим обучения прозрачного моста, который ранее нельзя было активизировать, так как порт не был уверен в том, что он останется корневым или назначенным и будет передавать пакеты данных. Стадия обучения (learning) также **выдерживается в течение интервала 15 с**. При этом порт продолжает участвовать в работе алгоритма STA, так что поступление пакетов BPDU с лучшими параметрами переводит его в заблокированное состояние.
- И только после двукратной выдержки по таймеру порт переходит в стадию продвижения (forwarding) и начинает продвигать пользовательские кадры в соответствии с построенной таблицей (которая продолжает модифицироваться, отражая изменения в структуре сети).
- Фактически в нашем примере в продвижении пользовательских пакетов после построения активной топологии участвуют только коммутаторы 111,333 и с 555 по 888.

- В процессе нормальной работы корневой коммутатор продолжает генерировать пакеты Hello, а остальные коммутаторы получают их через свои корневые порты и ретранслируют через назначенные порты. У коммутатора могут отсутствовать соединения с назначенными портами, как у коммутаторов 222 и 444, но он все равно участвует в работе протокола STP, так как корневой порт принимает служебные пакеты BPDU.