



**COPPE/UFRJ**

**Laboratório de Redes de Alta Velocidade -- RAVEL**

---

**Arquiteturas de Comutadores ATM**

**Luís Felipe Magalhães de Moraes -- Coordenador**

**Guilherme de Melo Batista Domingues**

---

**Relatório Técnico Ravel/01-97**

**18/04/97**

## 1 ) Introdução :

A utilização de fibras óticas permitiu um aumento considerável da largura de faixa que pode ser disponibilizada para os diversos serviços de transferência de informação. Com a evolução da tecnologia VLSI, tornou-se possível implementar equipamentos de comutação que operam em altas velocidades, permitindo o aparecimento de redes que são capazes de explorar todas as potencialidades dos meios óticos de transmissão.

A padronização de um modo de transferência assíncrono denominado por ATM ( "*Asynchronous Transfer Mode*" ), voltado ao atendimento simultâneo dos diversos serviços que são demandados em um ambiente digital integrado de altas velocidades, constituiu-se no fator chave para que uma rede de serviços digitais integrada faixa larga pudesse a vir se tornar uma realidade.

Em uma rede ATM, os equipamentos de comutação recebem, em suas portas de entrada, pacotes de tamanho fixo denominados por células. Estas, são compostas por 53 octetos, com 5 octetos perfazendo um cabeçalho que transporta informação de controle. A escala do tempo é subdividida em intervalos consecutivos cuja duração corresponde ao tempo de transmissão de cada célula.

Um comutador ATM com dimensão  $N \times N$  , nem sempre terá a capacidade de transferir  $N$  células, a cada segmento, para cada saída. Tal fato decorre diretamente da utilização de um fator de *Speedup*  $L$  ( $1 \leq L \leq N$ ) que limita em  $L$  o número máximo de células que podem ser transferidas para cada saída, a cada segmento, facilitando a escalabilidade da arquitetura com relação ao número de portas presentes.

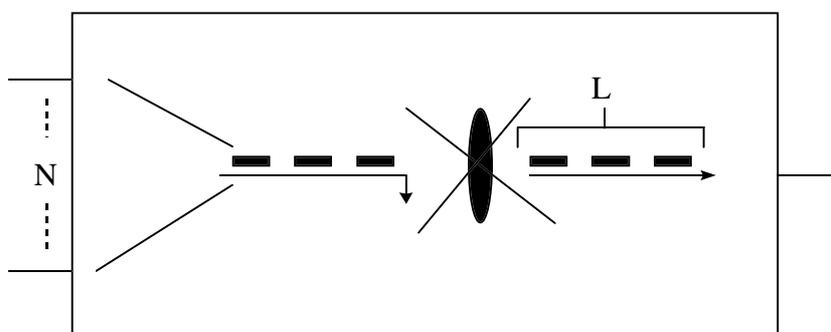


Fig.1 - Fator de *Speedup*  $L$

A obtenção analítica dos parâmetros de desempenho supõe a existência de um tráfego de entrada que seja independente. Dessa forma, em cada porta de entrada, considera-se a ocorrência, a cada segmento, da chegada de uma célula com probabilidade  $p$  e da chegada de nenhuma célula com probabilidade  $(1 - p)$ , independente do que tenha ocorrido no passado, e independente do que esteja ocorrendo nas demais portas de entrada.

Considera-se também que o tráfego de entrada esteja uniformemente distribuído pelas saídas. Conseqüentemente, em uma arquitetura com  $N$  portas de entrada e com  $N$  portas de saída, cada saída terá sempre uma probabilidade igual a  $\frac{1}{N}$  de receber uma célula que esteja chegando ao comutador.

Dentre os parâmetros de desempenho que são abordados, encontram-se a vazão máxima e a probabilidade de bloqueio das células. A vazão máxima corresponde ao maior fator de utilização por porta de saída que pode ser obtido, ao passo que, a probabilidade de bloqueio corresponde à taxa média de descarte de células por porta de entrada.

As arquiteturas ATM que operam baseadas em divisão do tempo utilizam um recurso de comunicação comum que é compartilhado por todas as portas, tipicamente uma memória ou um barramento. A escalabilidade da arquitetura quanto ao número de portas presentes estará limitada pela largura de faixa do dispositivo compartilhado, bem como pela necessidade de se utilizar uma lógica de controle centralizada para a coordenação do fluxo interno de células.

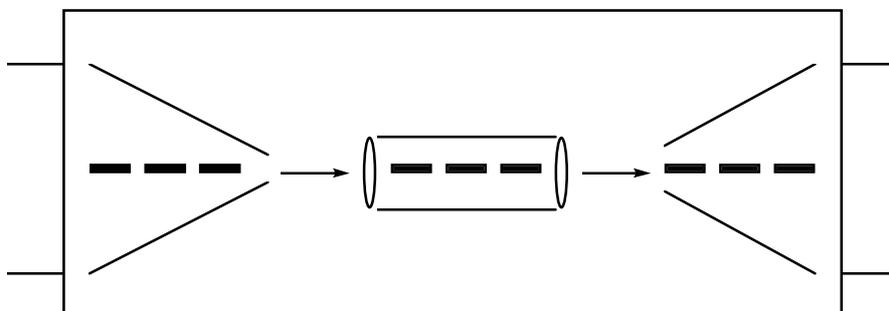


Fig. 2 - Canal de comunicação comum

Os comutadores ATM que operam baseados em divisão do espaço estabelecem várias rotas simultâneas no interior da arquitetura, podendo ou não utilizar pontos internos de interconexão para este fim. As que não utilizam elementos intermediários conseguem estabelecer  $N^2$  rotas disjuntas interligando as portas de entrada com as portas de saída, de forma a que exista um caminho físico distinto à partir de cada porta de entrada para cada porta de saída.

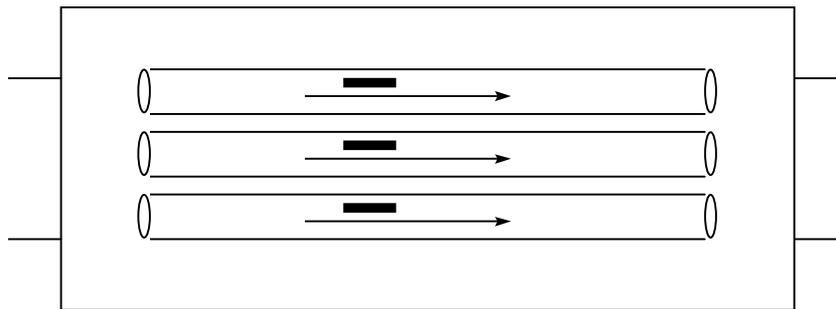


Fig. 3 - Estabelecimento de rotas simultâneas

Quando duas células que são destinadas para saídas diferentes competirem por recursos de comunicação no interior de um comutador ATM, uma delas terá a sua transferência bloqueada para as saídas, enquanto que a outra será entregue corretamente. Tais conflitos, denominados por conflitos internos, ocorrerão em arquiteturas denominadas por bloqueantes.

Devido à utilização de multiplexação assíncrona sobre os enlaces de transmissão, também ocorrerá bloqueios quando houver a tentativa do acesso simultâneo a uma mesma saída por mais de uma célula. Tais conflitos, denominados por conflitos de saída, existirão independentemente das características internas das arquiteturas dos comutadores ATM.

Por decorrência tanto de conflitos internos quanto de conflitos por saída, torna-se necessário a inserção de capacidade de armazenamento no interior do comutador, a fim de que a taxa de descarte de células seja reduzida. O modo através do qual a capacidade de armazenamento será distribuída no interior do comutador impactará sobremaneira o desempenho obtido.

## 2 ) Arquiteturas com Memória Compartilhada :

Neste tipo de arquitetura, uma memória é compartilhada por todas as portas do comutador, o que implica na utilização de uma estratégia de comutação baseada em divisão do tempo. A memória comum é particionada logicamente em  $N$  locações distintas, cada qual voltada ao armazenamento das células que estão destinadas a apenas uma das saídas. Uma largura de faixa de operação será necessária para que, a cada segmento, possa ocorrer o acesso de todas as portas à memória.

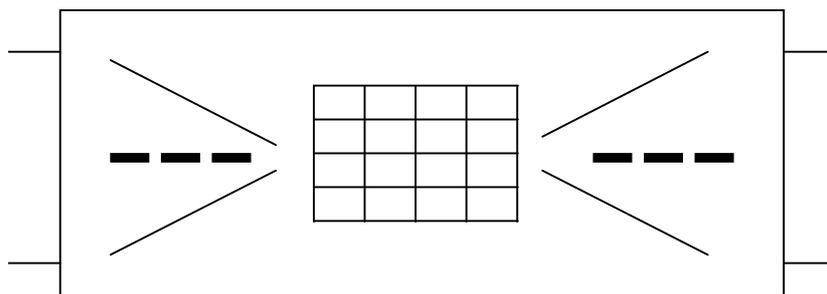


Fig. 4 - Memória compartilhada

O particionamento lógico da memória poderá ser fixo ou dinâmico. Se for fixo, cada uma das partições permanecerá sempre com a mesma capacidade, ao passo que, se for dinâmico, a capacidade de cada partição flutuará de acordo com a carga que é oferecida a cada porta de saída. Ocorrerá uma maior flexibilidade quando o particionamento for dinâmico, ao custo da utilização de algoritmos mais complexos de gerência da memória.

Uma lógica de controle centralizada deverá ser utilizada neste tipo de arquitetura, o que dificultará a sua escalabilidade com relação ao número de portas presentes. Esta lógica coordenará as portas de entrada no envio dos octetos para a memória comum, bem como responderá por enviar sinais de escrita para as portas de saída.

Como a capacidade de armazenamento é finita, algumas células poderão ser perdidas. Assim, deve-se projetar a capacidade da memória de forma a que a probabilidade de perdas e a vazão máxima se mantenham aceitáveis. Dentre os parâmetros que interferem no desempenho podemos citar : a disciplina de particionamento lógico, a dimensão  $N$  do comutador e as características do tráfego de entrada.

Um primeiro exemplo de arquitetura que utiliza uma memória compartilhada está descrito na referência [ 2 ]. Esta arquitetura, organiza as células no interior da memória de forma a que todos os primeiros octetos sejam armazenados em uma mesma região, todos os segundos octetos sejam armazenados em uma outra região distinta da primeira, etc.. .

As células são organizadas na memória de forma a que os seus octetos fiquem dispostos diagonalmente. Dessa forma, para que cada célula seja retirada integralmente da memória compartilhada, torna-se necessário apenas a localização do seu primeiro octeto. A posição dos demais octetos com relação ao primeiro já será automaticamente conhecida.

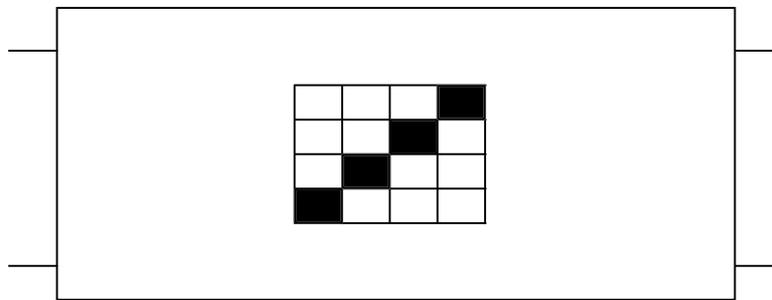


Fig. 5 - Disposição das células na arquitetura Prelude

Um segundo exemplo de arquitetura que utiliza a estratégia de se compartilhar memória pode ser encontrado na referência [ 3 ]. Esta arquitetura dedica parte da memória comum para que sejam montadas  $N$  listas encadeadas, cada qual estando associada a uma das  $N$  portas de saída existentes.

Cada lista encadeada consistirá de um conjunto de posições de memória que indicarão o local em que as células que são destinadas para uma mesma saída serão armazenadas. Associado a cada elemento da lista, existirá um ponteiro para o elemento seguinte da lista, o que permitirá a manutenção de uma sequência.

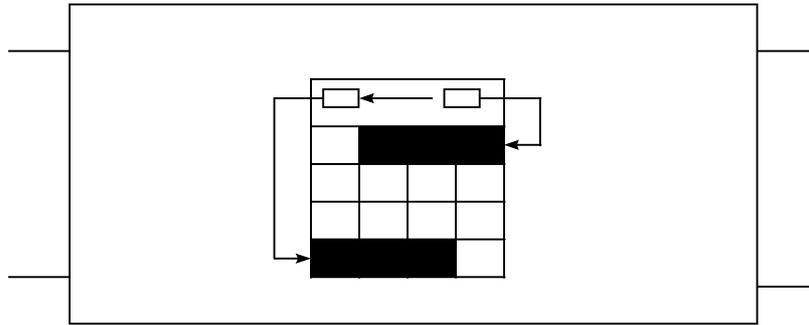


Fig. 6 - Listas encadeadas.

### 3 ) Arquiteturas com Meio Compartilhado :

Neste tipo de arquitetura, barramentos de comunicação são compartilhados por todas as portas, o que também implica em uma estratégia de comutação baseada na divisão do tempo. Cada saída possui um conjunto de interfaces que detectam, através de filtros de endereçamento, se os octetos que estão nos barramentos deverão ou não ser admitidos.

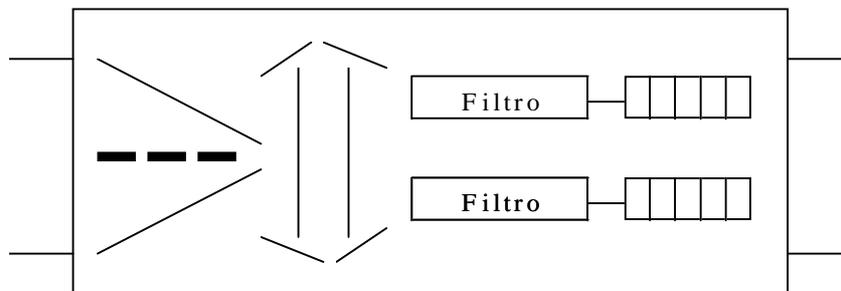


Fig. 7 - Meio compartilhado

Como mais de uma célula pode ser transferida em um mesmo segmento para uma mesma saída, os octetos que são admitidos pelas interfaces serão armazenados em filas FIFO. Essas arquiteturas diferirão das arquiteturas com memória compartilhada no que tange à disposição física dos elementos armazenadores, que serão independentes e distribuídos pelas saídas.

Uma lógica de controle centralizada também deverá ser utilizada neste tipo de arquitetura, o que dificultará a sua escalabilidade com relação ao número de portas presentes. Esta lógica coordenará as portas de entrada, para que cada uma envie, em um intervalo distinto, os seus octetos através dos barramentos. Simultaneamente, irá enviar sinais de escrita para as interfaces, permitindo que ocorra a cópia dos octetos para as filas FIFO .

Uma alternativa simples consiste no compartilhamento de apenas um barramento por todas as portas, de forma a que todos os octetos das células que estiverem presentes às entradas fluam por este canal. Haverá apenas uma interface por saída, que deverá operar a uma velocidade  $N$  vezes superior à velocidade  $V$  das portas.

Uma segunda alternativa consiste na utilização de mais de um barramento interno, de forma a que os octetos das células presentes às entradas sejam subdivididos em grupos. Neste caso, cada grupo de uma mesma célula será direcionado para uma mesma saída através de um barramento diferente. Consequentemente, existirá mais de uma interface por saída, de forma a que cada grupo possa ser armazenado em uma fila distinta.

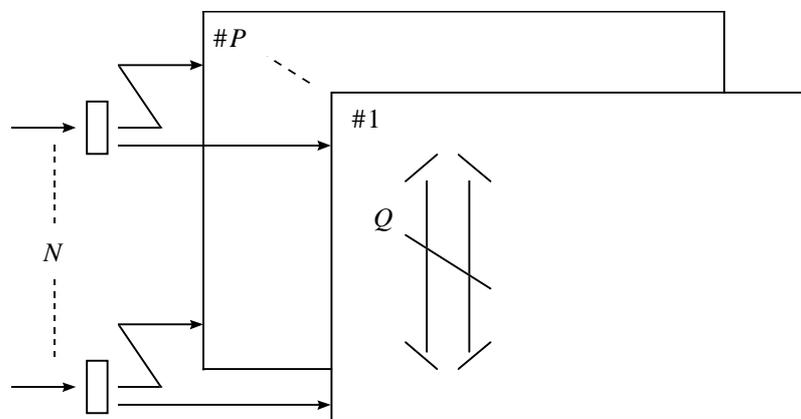


Fig. 8 - Multiplicidade de barramentos internos

No segundo caso, torna-se necessário um pré-processamento das células nas entradas, para que possa ser realizada a adição de uma referência, em cada grupo, sobre as saídas para as quais estes estão direcionados. A principal vantagem de se utilizar mais de um barramento consiste na diminuição da largura de faixa de operação dos elementos internos. Se forem utilizados  $P$  barramentos, cada interface poderá operar a uma velocidade  $Q$ , onde  $PQ = NV$ .

Um exemplo de arquitetura proposta na literatura que adota a estratégia de barramento compartilhado está descrito na referência [ 4 ]. Esta arquitetura, denominada por Athena, constitui um exemplo onde as portas estão conectadas a mais de um barramento. As filas estão distribuídas por módulos independentes e armazenam 1 /8 da carga útil de cada célula.

#### **4 ) Arquiteturas Crossbar :**

Uma arquitetura *crossbar* utiliza uma matriz composta por  $N^2$  elementos intermediários, cada qual permitindo a conexão de uma entrada com uma saída. Se uma célula na porta de entrada  $i$  ( $i = 1,2,\dots,N$ ) desejar acessar a porta de saída  $j$  ( $j = 1,2,\dots,N$ ), o comutador irá propagá-la em linha reta da porta de entrada  $i$  até o ponto de interconexão  $( i, j )$ , para, em seguida, propagá-la verticalmente até a porta de saída  $j$ . Neste tipo de arquitetura, ocorre a divisão do espaço.

O retardo de propagação de duas rotas internas pode ser diferente, já que cada uma pode atravessar um número diferente de elementos intermediários. Torna-se necessário a utilização de atrasadores no interior da arquitetura. A figura abaixo mostra um exemplo de arquitetura *crossbar* onde existem 4 entradas e 4 saídas, com as entradas dispostas horizontalmente e com as saídas dispostas verticalmente. As duas rotas estabelecidas possuem atrasos diferentes.

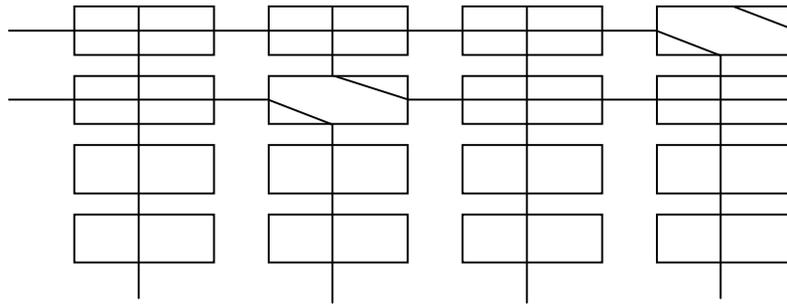


Fig. 9 - Arquitetura Crossbar

Os elementos intermediários poderão assumir dois estados mutuamente exclusivos: *cross* e *bar*. No estado *cross*, a entrada horizontal se conecta com a saída horizontal e a entrada vertical se conecta com a saída vertical, ao passo que, no estado *bar*, a entrada horizontal se conecta com a saída vertical e a entrada vertical se conecta com a saída horizontal.

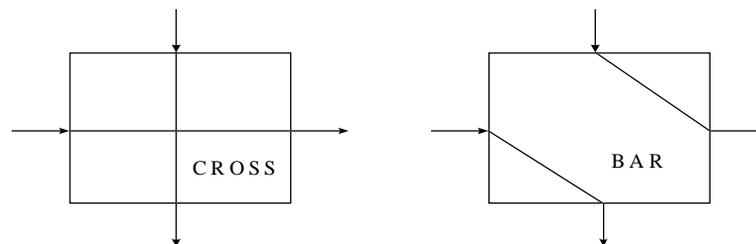


Fig. 10 - Estado possível dos elementos de interconexão

Em cada segmento temporal, todos os elementos de interconexão serão inicializados no estado *cross*. Posteriormente, se for estabelecida uma conexão entre a porta de entrada  $i$  e a porta de saída  $j$ , o ponto de interconexão ( $i, j$ ) será levado do estado *cross* para o estado *bar*.

A arquitetura do tipo *crossbar* não causa bloqueios internos, já que duas rotas para saídas distintas, não colidirão no interior do comutador. Desta forma, a existência de  $N^2$  elementos intermediários de interconexão permite que sejam estabelecidos  $N$  circuitos internos paralelos, cada qual associado a uma saída diferente.

No entanto, devido à ocorrência de conflitos por saída, torna-se necessário a utilização de capacidade de armazenamento para se reduzir as perdas. A vazão máxima de uma arquitetura *crossbar* desprovida de filas converge rapidamente, com o aumento de  $N$ , para o valor de 63,2%, quando o tráfego de entrada é independente e uniformemente distribuído. Isto provoca um descarte de 36,8% das células que chegam ao comutador [ 5 ].

Uma primeira alternativa para se reduzir a taxa de perdas consiste na inserção de uma fila em cada elemento intermediário. Uma célula presente à porta de entrada  $i$ , que seja endereçada à porta de saída  $j$ , e que não consiga acessar esta saída no segmento em que chegou ao comutador, irá ser armazenada no ponto de interconexão  $(i, j)$ .

Esta arquitetura possui uma vazão máxima próxima de 100%. Pode-se modelar tal arquitetura como uma sendo não bloqueante e como possuindo filas que são inseridas somente nas saídas. Serão utilizadas  $N$  filas independentes por saída, cada qual destinada a receber o tráfego que é proveniente de uma das portas de entrada.

Além da necessidade de muito mais capacidade de armazenamento, com relação às arquiteturas que utilizam apenas uma fila por saída, para uma mesma taxa de perdas, a existência de  $N$  filas em cada saída limita a escalabilidade do comutador com relação ao número de portas presentes, sobretudo nos casos em que as filas são extensas. Um exemplo de arquitetura proposta na literatura que utiliza este tipo de estratégia, denominada por *Bus Matrix Switch*, pode ser encontrada em [ 6 ] .

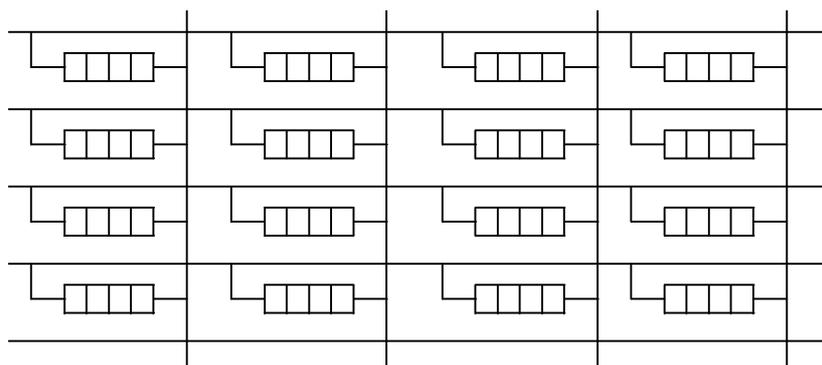


Fig. 11 - Arquitetura Bus Matrix

Uma segunda alternativa para se reduzir a taxa de perdas consiste na adição de memória somente às portas de entrada. Um mecanismo para controlar o acesso às portas de saída deverá ser implementado, a fim de que se torne possível a seleção das células que serão transferidas para as saídas dentre as que estiverem presentes à cabeça das filas de entrada.

O seu modelo consistirá de uma arquitetura não bloqueante que possuirá apenas uma fila em cada entrada. A vazão máxima que poderá ser atingida, quando o tráfego for uniforme e independente, convergirá rapidamente, com o aumento de  $N$ , para o valor de 58,6% [ 7 ].

Uma primeira alternativa para o mecanismo de controle em uma arquitetura *crossbar* com filas somente nas entradas consiste em uma lógica completamente centralizada. Por se tornar um dos fatores que limita a escalabilidade quanto ao número de portas presentes, torna-se mais interessante a adoção de uma segunda alternativa onde a lógica é distribuída pelas saídas. Neste caso, as unidades de controle se chamarão árbitros.

Cada árbitro receberá o endereço destino de todas as células que estiverem presentes à cabeça das filas de entrada. Posteriormente, cada árbitro selecionará uma célula dentre as que estiverem destinadas para a saída ao qual está acoplado. Seguirão sinais de *backpressure* para as entradas, bloqueando as células que não foram selecionadas. Um exemplo de arquitetura proposta que utiliza lógica distribuída pode ser encontrada em [ 8 ].

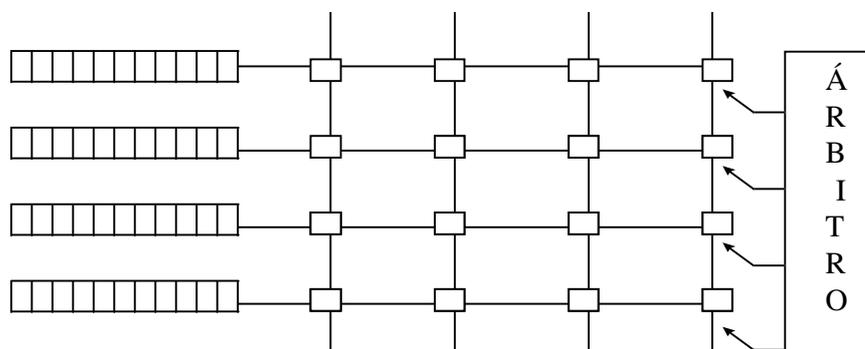


Fig.12 - Lógica distribuída

Uma terceira alternativa para se reduzir as perdas consiste na utilização das duas técnicas descritas acima, na tentativa de se aumentar o desempenho com relação às arquiteturas com filas somente nas entradas e na tentativa de se reduzir a quantidade de memória utilizada com relação às arquiteturas que usam filas apenas nos elementos intermediários.

O seu modelo consistirá de uma arquitetura não bloqueante que possui uma fila em cada entrada e  $N$  filas em cada saída, cada qual destinada a receber as células provenientes de uma entrada diferente. Um exemplo de arquitetura que utiliza esta alternativa, denominada por LIB ( *Limited Intermediate Buffer* ), na qual é inserida uma fila com apenas uma posição em cada elemento de interconexão, pode ser encontrada em [ 9 ] . Neste caso, quando  $N = 16$ , a vazão máxima passível de ser atingida possui o valor de 87,5 % , efeito direto da redução do bloqueio de células na cabeça das filas de entrada.

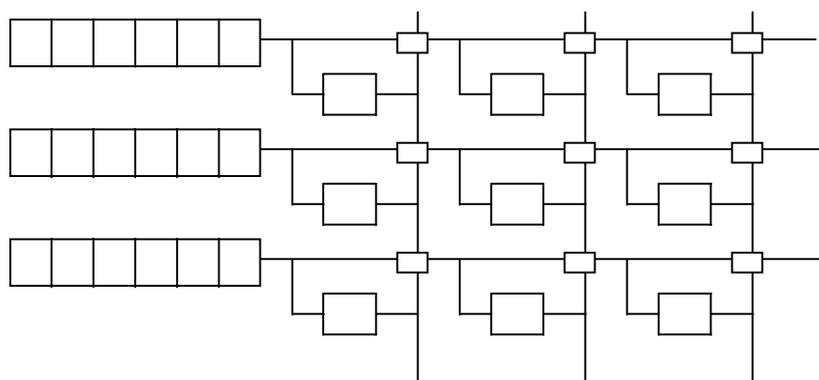


Fig.13 - Arquitetura LIB

## 5 ) Arquiteturas de Banyan :

A utilização das arquiteturas *MIN* ( *Multistage Interconnection Networks* ) [ 10 ] onde os elementos internos de interconexão estão dispostos ao longo de estágios sucessivos permite que se construa comutadores com divisão do espaço que possuem uma quantidade de pontos de interconexão inferior à que é utilizada nas redes *Crossbar*. Uma rede *MIN* com três estágios é ilustrada na figura abaixo :

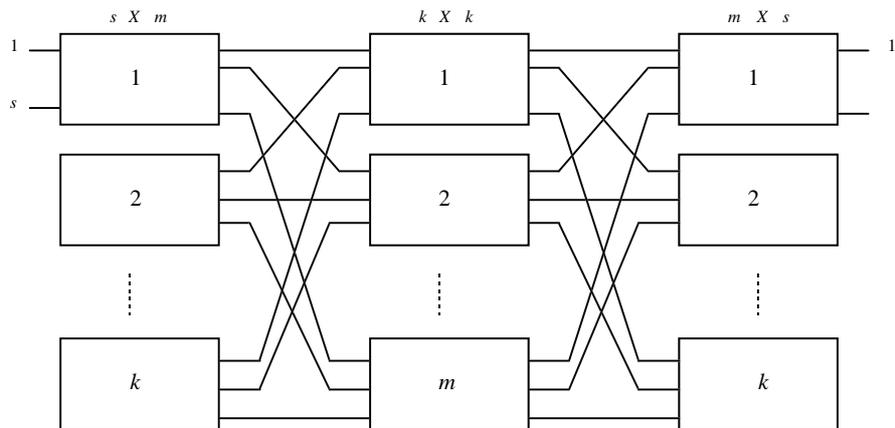
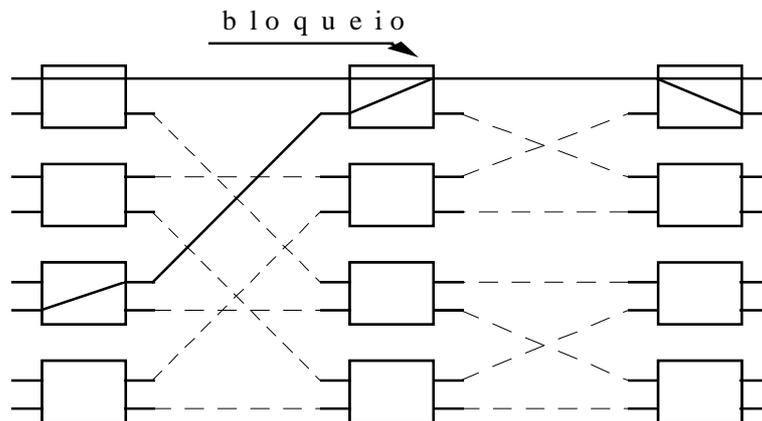


Fig. 14 - Rede MIN

Dentre as redes *MIN* se destacam as redes de *Banyan* [ 11 ], onde só é possível o estabelecimento de uma rota entre uma porta de entrada e uma porta de saída. Quando cada dígito do endereçamento de uma célula passa a ser interpretado por um estágio diferente da rede, a lógica de controle se torna distribuída pela arquitetura, de forma a que as células possam ser roteadas à medida em que forem sendo propagadas pela arquitetura.

As redes de *Banyan* descritas acima tornam-se atraentes por apresentarem o mesmo retardo de propagação para duas rotas distintas, ao contrário do que ocorre nas redes *crossbar*. A ordem de complexidade da arquitetura com relação à quantidade de elementos de interconexão será igual a  $O\left(\frac{N}{2} \log_2 N\right)$ , decorrente da existência de  $\frac{N}{2}$  estágios, cada qual sendo composto por  $\log_2 N$  nós de interconexão.

As redes de *Banyan* são caracterizadas pela ocorrência de bloqueios internos, decorrente da existência de apenas uma rota entre uma porta de entrada e uma porta de saída. Este tipo de conflito é ilustrado na figura abaixo, onde torna-se necessário configurar um mesmo elemento de duas maneiras distintas para que duas rotas com destinos diferentes possam ser estabelecidas simultaneamente.



Pode-se mostrar que uma rede de *banyan* desprovida de capacidade de armazenamento atingirá uma vazão máxima de saturação igual a 26% quando  $N = 1024$ , o que implicará na perda de 74% das células que chegam [ 5 ]. Dessa forma, a existência de bloqueios internos degradará sobremaneira o desempenho obtido.

Uma primeira alternativa para que se possa contornar o descarte elevado decorrente dos bloqueios internos consiste na utilização de filas nos pontos de interconexão. Quando ocorrer um conflito, as células que não puderem acessar os recursos internos de comunicação serão mantidas bloqueadas no interior da arquitetura.

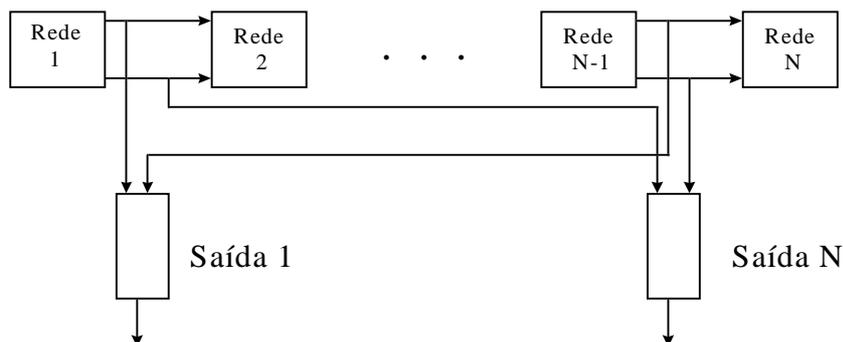
Uma segunda alternativa para que se possa contornar o descarte elevado decorrente dos bloqueios internos consiste na utilização de uma rede de distribuição precedendo a rede de roteamento. A rede de distribuição também uma rede de *Banyan*, responde pela reordenação do tráfego de entrada, de forma a que ocorra uma diminuição dos conflitos internos na rede de roteamento. Pode-se visualizar ambas as redes como uma única rede *MIN*, na qual existe mais de um circuito físico que pode ser estabelecido de uma entrada para uma saída.

Na referência [ 12 ], encontra-se uma arquitetura denominada por *ISPN*, onde se adicionam filas aos elementos de interconexão e se utiliza uma rede de distribuição ( *DN - Distribution Network* ) precedendo a rede de roteamento ( *RN - Routing Network* ). Nesta arquitetura, também se utiliza uma terceira rede que recebe as células das portas de entrada e cria várias cópias de uma mesma célula ( *CN - Copy Network* ), permitindo que se adicione características de *multicast* e de *broadcast* à arquitetura.

Uma terceira alternativa para que se possa contornar o descarte elevado decorrente dos bloqueios internos consiste na utilização de roteamento por deflexão. Neste caso, na ocorrência de um conflito interno, uma célula será direcionada corretamente para as saídas, ao passo que as demais serão roteadas por caminhos alternativos para que possam realizar novas tentativas de acesso às saídas.

Na referência [ 13 ], encontra-se uma arquitetura denominada por *Tandem Banyan*, que utiliza roteamento por deflexão através de um conjunto de redes de *banyan* dispostas em série. Cada célula é transferida das portas de entrada para a primeira rede. Se houver conflito interno, uma das células é encaminhada corretamente para as saídas enquanto que as demais serão encaminhadas para a entrada da segunda rede. São utilizadas filas nas saídas, pois existe a possibilidade de mais de uma célula ser transferida para uma mesma saída, em um mesmo segmento.

Este processo se repetirá até a última rede. Como a carga decresce à medida em que novas redes de *banyan* vão sendo utilizadas, a quantidade de conflitos internos vai também sendo reduzida. Mostra-se que, para um número de redes de *banyan* igual a 9, tem-se que a probabilidade de perdas é inferior a  $10^{-6}$  quando a carga de entrada normalizada por porta de entrada é próxima de 100 % e quando o tráfego de entrada é independente e uniformemente distribuído pelas saídas.



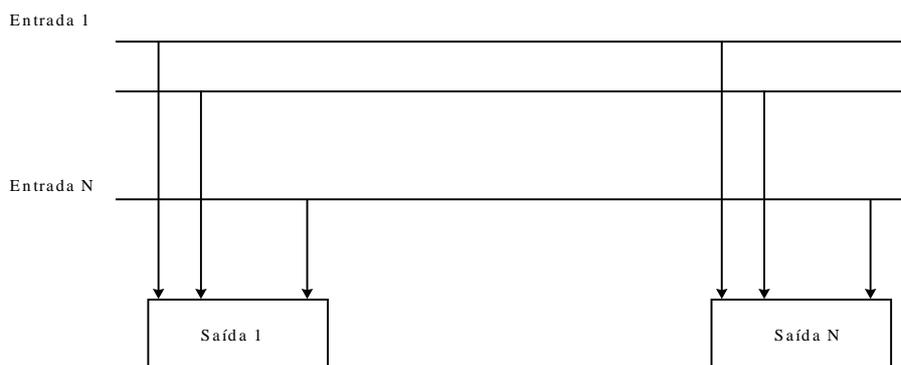
Uma quarta alternativa para que se possa contornar o descarte elevado decorrente dos bloqueios internos consiste na utilização de uma rede de ordenação precedendo a rede de roteamento. Neste caso a arquitetura se tornará não bloqueante, pois ocorrerá uma ordenação das células, com base no endereço de destino, na entrada da rede de roteamento. Os conflitos internos serão eliminados.

Na referência [ 14 ], encontra-se uma arquitetura que utiliza, além de uma rede de *Batcher*, filas nas portas de entrada. Um algoritmo que é composto por três fases é utilizado para que se possa determinar quais as células que serão mantidas bloqueadas quando ocorrerem os conflitos de acesso por uma mesma porta de saída.

Na primeira fase do algoritmo, ocorre a ordenação, por parte da rede de *Batcher*, de requisições de acesso às saídas, baseada no endereço de destino das células. Na segunda fase, é enviado para as entradas um conjunto de pacotes de confirmação que determinam quais as células, dentre as que estiverem presentes à cabeça das filas de entrada, que serão bloqueadas. Por fim, na terceira fase, procede-se a transferência das células para as saídas através da rede de roteamento.

## 6 ) Arquiteturas com $N^2$ Rotas Disjuntas :

As arquiteturas baseadas em matrizes de interconexão total provêem uma conexão física direta de cada entrada para cada saída, sem que seja necessário a utilização de elementos intermediários para realizar estas conexões. Desta forma, uma matriz com  $N^2$  linhas que permite que  $N^2$  rotas disjuntas coexistam, interligando diretamente as entradas com as saídas, de forma a que não ocorra bloqueio interno algum.



Um exemplo de arquitetura proposta que utiliza arquitetura baseada em matriz de interconexão total, denominada por *Knockout*, pode ser encontrada em [ 15 ]. Nesta proposta, as filas são inseridas apenas nas portas de saída. Cada porta de saída possui uma interface que é capaz de receber, em cada segmento temporal, todas as células que estão presentes às portas de entrada. Cada interface é composta por um filtro, por um concentrador, por uma área de armazenamento comum e por um registro de deslocamento.

Cada filtro responde por analisar o endereço de destino de cada célula que foi recebida, determinando quais as células que são destinadas a uma mesma saída. Estas, são entregues ao concentrador, que utiliza um algoritmo, denominado por algoritmo *Knockout*, a fim de que sejam selecionadas aleatoriamente um máximo de  $L$  células dentre as células que foram entregues.

O algoritmo de concentração é implementado em *hardware*, rememorando um torneio no qual um vencedor aplica um *Knockout* nos demais concorrentes. O bloco básico do concentrador consiste em um elemento de dimensão  $2 \times 2$ , e é composto por uma saída vencedora e uma saída perdedora. Quando duas células chegam simultaneamente a um elemento básico, apenas uma poderá ser encaminhada para a saída vencedora.

Na primeira rodada do primeiro torneio, sobram metade das células que chegaram ao concentrador. Na segunda rodada, os vencedores da primeira rodada serão reduzidos à metade. Este procedimento se repetirá até que, ao final de um número de rodadas consecutivas, exista apenas um vencedor. Os perdedores competirão entre si novamente em um segundo torneio, de forma a que se defina um novo vencedor. O concentrador é construído de forma a que sejam realizados, um máximo de  $L$  torneios consecutivos.

As células selecionadas como vencedores pelo concentrador serão entregues a uma mesma área de armazenagem. O comutador deverá possuir uma velocidade interna de operação tal que seja possível transferir, em cada interface, um máximo de  $L$  células, em um mesmo segmento temporal, para a área comum de armazenagem. Como um processador diferente responde por controlar a operação em cada interface, diz-se que a arquitetura possui uma velocidade de operação  $L$  vezes superior à velocidade das portas de entrada.

## 7 ) Bibliografia :

[ 1 ] - J. Garcia-Haro, A. Jajszczyk, "ATM Shared-Memory Switching Architectures", IEEE Network, July / August, pp. 18 - 26, 1994.

[ 2 ] - P. Coudrese, M.Servel, "Prelude : an Asynchronous Time-Division Switched Network", ICC '87 Conf. Rec., paper 22.2, Seattle, WA, June 1987.

[ 3 ] - H. Kuwahara et. al., "A Shared Buffer Memory Switch for an ATM Exchange", ICC' 89 Conf. Rec., Boston, MA, June 1989, pp. 118 - 122.

[ 4 ] - M. de Prycker, "Asynchronous Transfer Mode", Ellis Horwood, 1992.

[ 5 ] - F.A.Tobagi, "Fast Packet Switch Architectures For Broadband Integrated Services Digital Networks", Proc. IEEE, Vol. 78, N° 1, Janeiro 1990.

[ 6 ] - S. Nojima, et al., "Integrated services packet networking using bus matrix switch", IEEE Journal on Selected Areas in Communications, Vol. 5, N° 8, pp. 1284 - 1356, Dezembro 1987.

[ 7 ] - M.J.Karol, M.G.Hluchyj, "Input Versus Output Queueing on a Space-Division Packet Switch", IEEE Transactions on Communications, Vol COM-35, No. 12, December 1987.

[ 8 ] - M. de Prycker, M. de Sommer, "Performance of a service independent switching network with distributed control", IEEE JSAC, Vol. SAC-5, pp. 1293 - 1301, Oct. 1987.

[ 9 ] - A.K.Gupta, L.O.Barbosa, N.D.Georganas, "A 16 x 16 limited intermediate buffer switch module for ATM networks" , Proc GLOBECOM '91, Phoenix, pp.939 - 943, Dez 1991.

[ 10 ] - C. Clos, "A study of non-blocking switching network", Bell Syst. Tech. J., Vol 32, pp. 406-424, Março 1953.

[ 11 ] - L.R.Goke, G.J.Lipovski, "Banyan networks for partitioning multiprocessor systems", Proc. 1st. Annu. Int. Symp. Comput. Architecture, pp. 21-28, Dezembro 1973.

- [ 12 ] - J.S.Turner, "Design of a broadcast packet switching network", Proc. INFOCOM '86, pp. 667-675, Apr. 1986.
- [ 13 ] - F.Tobagi, T.Kwok, F.Chiussi, "Architecture, Performance and Implementation of the Tandem Banyan Fast Packet Switch", IEEE J. Select. Areas Commun., Vol.9, N° 8, pp. 1173 - 1193, Outubro 1991.
- [ 14 ] - J.Y.Hui, "A broadband packet switch for multi-rate services, IEEE J. Select. Areas Commun., Vol 5, pp. 1264-1273, Outubro 1987.
- [ 15 ] - Y.S.Yeh, M.G.Hluchyj, S.A.Acampora, "The Knockout switch: A simple, modular architecture for high-performance packet switching", IEEE J. Select. Areas Commun., Vol 5, pp.1274-1283, Outubro 1987.
- [ 16 ] - G. E. Daddis, H. C. Torng, "A Taxonomy of Broadband Integrated Switching Architectures", IEEE Communications Magazine, Maio 1989.
- [ 17 ] - P. Newman, "ATM Technology for Corporate Networks", IEEE Communications Magazine, Abril 1992.
- [ 18 ] - H. Ahmadi, W.E.Denzel, "A Survey of Modern High-Performance Switching techniques", Journal on Selected Areas in Communications, Vol.7, n° 7, Setembro 1989.
- [ 19 ] - A.R.Jacob, "A Survey of fast packet switches", Computer Communication Review, Vol.20, N° 1, pp 54-64, Janeiro 1990.
- [ 20 ] - E. W. Zegura, "Architectures for ATM Switching Systems", IEEE Communications Magazine, Fevereiro 1993.
- [ 21 ] - R. Y. Awdeh, H.T. Mouftah, "Survey of ATM switch architectures", Computer Networks and ISDN Systems, Vol 27, pp 1567 - 1613, 1995.