

**UNIVERSIDADE DE SÃO PAULO**  
**INSTITUTO DE FÍSICA DE SÃO CARLOS**  
**DEPARTAMENTO DE FÍSICA E INFORMÁTICA**

PROPOSTA E SIMULAÇÃO DE UMA ESTRUTURA DE ROTEAMENTO PARA  
CHAVES ATM

**ANGELO CESAR COLOMBINI**

Tese apresentada ao Instituto de Física de São Carlos, Universidade de São Paulo, para  
obtenção do título de Doutor em Ciências “Física Aplicada – Opção: Física Computacional”

Orientador: Prof. Dr. Carlos Antonio Ruggiero

*de*

São Carlos – São Paulo  
1999

**USP/IFSC/SBI**  
  
**8-2-001296**

**IFSC-USP** SERVIÇO DE BIBLIOTECA  
INFORMAÇÃO

Colombini, Angelo Cesar

“Proposta e Simulação de uma estrutura de roteamento para chaves ATM” / Angelo Cesar Colombini – São Carlos, 1999

Tese (Doutorado) – Instituto de Física de São Carlos da Universidade de São Paulo, 1999

Área: Física Aplicada-opção Física Computacional

Prof. Dr. Carlos Antonio Ruggiero

1. Chaves ATM. 2. BISDN. 3. Roteamento.

I. Título



MEMBROS DA COMISSÃO JULGADORA DA TESE DE DOUTORADO DE ANGELO CESAR COLOMBINI APRESENTADA AO INSTITUTO DE FÍSICA DE SÃO CARLOS, DA UNIVERSIDADE DE SÃO PAULO, EM 15 DE OUTUBRO DE 1999.

COMISSÃO JULGADORA:

Prof. Dr. Carlos Antonio Ruggiero/IFSC/USP

Prof. Dr. Alvaro Garcia Neto/IFSC-USP

Prof. Dr. Luis Carlos Trevelin/UFSCar

Prof. Dr. Sergio Takeo Kofuji/EP-USP

Dr. Fredy João Valente/S&V Consultoria e Tecnologia

## DEDICATÓRIA

*À minha dedicada esposa, Maria Emília, por seu grande incentivo e motivação. Os adjetivos se tornam poucos quando tento expressar a grandeza dessa mulher. Aos meus tesouros Renan, Caio e Bruna, que foram os anjos inspiradores.*

*E a todos meus familiares que contribuíram para a realização deste trabalho. De um modo especial a minha irmã Rita pelo suporte na redação.*

## AGRADECIMENTOS

- Ao Prof. Dr. Carlos Antônio Ruggiero, pela orientação precisa, incentivo e principalmente por estar sempre aberto a discussões. Saber ouvir é uma virtude de poucos.
- Ao Prof. Dr. Jan Slaetes, pelo incentivo, discussões e comentários sempre precisos, que acabaram por nortear muitos tópicos deste trabalho.
- Ao Prof. Dr. Gonzalo Travieso, pelo grande suporte na implementação do simulador, e por sua incrível disposição em ajudar.
- Ao Prof. José da Silveira Maia, pelo apoio e incentivo fundamentais para que os prazos fossem cumpridos.
- Ao Instituto de Física de São Carlos pelo suporte físico e instrumental.
- A Universidade de Franca – UNIFRAN pelo suporte físico e financeiro.
- Ao CNPq pelo suporte financeiro, sem o qual este trabalho certamente não teria sido desenvolvido.
- A todo grupo de redes do Instituto de Física, pela troca de informação e incentivo.
- A todos os amigos que contribuíram de uma forma direta ou indireta na realização deste trabalho. Seria difícil mencionar o nome de todos e eu correria o risco de cometer alguma injustiça se não mencionasse o nome de algum.

*De um modo especial agradeço a  
Deus, meu apoio, meu guia e minha  
força....*

## SUMÁRIO

Lista de figuras .....	x
Lista de tabelas .....	xvi
Resumo .....	xvii
Abstract .....	xviii
Introdução .....	xix

### CAPÍTULO 1: Evolução Histórica Padronização e Tecnologias

#### Alternativas

1 Histórico .....	1
1.2 O mundo das telecomunicações .....	3
1.3 Métodos de transmissão de dados .....	5
1.3.1 Transmissão de dados assíncrona e síncrona .....	6
1.3.2 Modo de transferência assíncrono versus síncrono . .....	7
1.4 Multiplex e chaves .....	8
1.4.1 Multiplexação por divisão de espaço .....	8
1.4.2 Multiplexação por divisão de frequência (FDM) .....	9
1.4.3 Multiplexação por divisão do tempo (TDM) .....	9
1.4.4 Multiplexação por endereço .....	10
1.4.5 Funções das chaves ponto-a-ponto .....	10
1.4.6 Funções das chaves ponto-para-multiponto .....	11
1.4.7 Exemplos de chaves .....	12
1.4.7.1 Chaves por divisão de espaço .....	12
1.4.7.2 Chaves por divisão do tempo .....	13
1.4.7.3 Chaveamento por endereços .....	14
1.4.7.4 Chaveamento de frequência/comprimento de onda .....	16
1.5 ATM: O que é e por quê? .....	18
1.6 Padronização .....	19
1.6.1 Atividades do ITU-T .....	19

1.6.2 Atividades do ATM Forum .....	22
1.6.3 Atividades do ANSI .....	23
1.6.4 Atividades do IETF .....	24
1.7 O protocolo B-ISDN .....	24
1.7.1 Conceitos gerais .....	24
1.7.2 Componentes das camadas .....	25
1.8 Tecnologias alternativas .....	29
1.8.1 Na LAN .....	29
1.8.1.1 Fiber Distributed Data Interface FDDI .....	30
1.8.1.2 Ethernet e Token ring chaveados .....	31
1.8.1.3 Fast Ethernet .....	32
1.8.1.4 Fiber Chanel .....	33
1.8.1.5 Outras .....	33
1.8.2 Na WAN .....	33
1.8.2.1 ISDN .....	34
1.8.2.2 Frame Relay .....	35
1.8.2.3 Distributed Queue Dual Bus (DQDB) .....	35

## CAPÍTULO 2: ATM Conceitos fundamentais

2.1 A rede ATM .....	37
2.2 A célula ATM .....	38
2.2.1 Segmentação de células ATM .....	39
2.2.2 Um exemplo de como as células podem ser organizadas .....	41
2.3 Caminho virtual e canal virtual .....	42
2.3.1 Considerações a respeito do VPI .....	44
2.3.2 Considerações a respeito do VCI .....	46
2.4 Endereçamento e roteamento .....	48
2.4.1 Atributos desejáveis pela camada ATM .....	49
2.4.2 Requisitos para roteamento .....	49
2.5. Formatos dos endereços ATM .....	50
2.5.1 IP sobre ATM .....	52

2.5.2 Sinalização .....	55
2.5.2.1 Sinalização AAL .....	56
2.5.2.2 Conexão ponto-a-ponto .....	58
2.5.2.3 Conexão ponto-para-multiponto .....	60
2.6 Qualidade de serviço QoS .....	62
2.6.1 Classes QoS .....	64
2.6.3 QoS não especificadas .....	66
2.6.4 Aplicação das diferentes QoS .....	67
2.7 Controle de tráfego em redes ATM .....	68
2.7.1 Introdução .....	68
2.7.2 Controle de tráfego .....	68
2.7.2.1 Modelo de referência .....	68
2.7.2.2 Descritor de tráfego .....	71
2.7.3 Parâmetros de controle usuário/rede .....	72
2.7.3.1 Leaky Buckety UPC/NPC .....	73
2.7.3.2 Sliding and Jumping Window UPC .....	74
2.7.3.3 Outros métodos de controle de tráfego .....	76
<b>CAPÍTULO 3: A chave ATM</b>	
3.1 Introdução .....	77
3.2 Plano de controle .....	77
3.3 Plano de gerenciamento .....	82
3.3.1 Interim Local Management (ILMI) .....	84
3.4 Característica de uma chave ATM .....	84
3.5 Arquitetura de uma chave ATM .....	85
3.6 Módulo de entrada (IM) .....	89
3.6.1 Funções SONET .....	92
3.6.2 Delineação de células e controle de erros no header .....	93
3.6.3 Bloco UPC/NPC .....	95
3.6.4 Processamento de células .....	96
3.6.5 O IM-CAC .....	97

3.6.6 IM-SM .....	99
3.7 Módulo de saída (OM) .....	99
3.7.1 Análise do processamento de células .....	100
3.7.1.1 Banco de dados VP/VC .....	101
3.7.1.2 Inserção de células de gerenciamento .....	102
3.7.1.3 Inserção de células de sinalização .....	102
3.7.1.4 Processamento e manutenção .....	102
3.7.2 Convergência da transmissão .....	103
3.7.3 Funções SONET .....	104
3.7.4 OM-CAC e OM-SM .....	105
3.8 Estrutura de célula da chave (CSF) .....	105
3.8.1 Roteamento & Buffer .....	106
3.8.1.1 Totalmente interconectada .....	107
3.8.1.2 Divisão de espaço .....	108
3.8.2 Concentração e expansão .....	112
3.8.3 Cópia e multicast .....	112
3.8.4 Gerenciamento de buffer .....	113
3.9 Controle e Admissão da Conexão (CAC) .....	114
3.9.1 Alocação de recursos .....	115
3.10 Sistema de Gerenciamento (SM) .....	116
3.10.1 Informações de gerenciamento .....	118
<b>CAPÍTULO 4: Proposta para Simulação</b>	
4.1 Revisão dos trabalhos .....	120
4.1.1 A arquitetura da chave Starlite .....	120
4.1.2 A arquitetura de chave St.Louis .....	122
4.1.3 O elemento de roteamento Knockout .....	124
4.1.4 Técnica de agrupamento de canais de saída na estrutura Tanden Banyan Switching Fabric (TBSF) .....	127
4.1.4.1 A arquitetura TBSF .....	128
4.1.4.2 Algoritmo bit mask .....	129

4.1.5 Controle adaptativo de tráfego .....	130
4.1.6 Arquitetura modular e recursiva para implementar multicast em larga escala na MOBAS .....	133
4.1.7 A arquitetura Clos Knockout .....	139
4.1.8 O compartilhamento de buffer empregado em chaves ATM com o objetivo de facilitar a realização do multicast .....	138
4.1.8.1 Multicast em chaves ATM com buffer compartilhado .....	138
4.1.8.2 Abordagem baseada em busca na fila .....	139
4.1.8.3 Abordagem baseada no endereçamento de memória CAM ...	140
4.1.8.4 Abordagem baseada em fila .....	141
4.1.9 Multicast em chaves de alto desempenho .....	142
4.2 A arquitetura proposta .....	144
CAPÍTULO 5: Simulador e resultados da simulação	
5.1 O simulador .....	151
5.1.1 A célula ATM .....	152
5.1.2 O hardware .....	157
5.1.3 A movimentação das células na estrutura de roteamento .....	162
5.2 A simulação .....	163
5.2.1 Simulação de células unicast .....	163
5.2.2 Simulação de células multicast .....	167
CAPÍTULO 6: Conclusão e proposta de trabalhos futuros	
6.1 Conclusão .....	172
6.2 Proposta de trabalhos futuros .....	173
Referências Bibliográficas .....	175
Apêndice A .....	181
Glossário .....	186

## LISTA DE FIGURAS

### CAPÍTULO 1: Evolução Histórica e Tecnologias Alternativas

Figura 1.1 -	Transmissão de caracteres no modo assíncrono	6
Figura 1.2 -	Formato de um frame síncrono	7
Figura 1.3 -	Ilustração do modo de transferência STM e ATM	8
Figura 1.4 -	Definição das funções de chaveamento ponto-a-ponto	11
Figura 1.5 -	Definição das funções de chaveamento ponto-para-multiponto	12
Figura 1.6 -	Rede de conexão cruzada de duas entradas e duas saídas	13
Figura 1.7 -	Operação de uma chave por divisão de tempo	14
Figura 1.8 -	Operação de uma chave por endereço	15
Figura 1.9 -	Ilustração de uma rede ótica WDM	16
Figura 1.10 -	Ilustração de uma rede WDM de múltiplos saltos	17
Figura 1.11 -	Célula ATM	25
Figura 1.12 -	O B-ISDN	26
Figura 1.13 -	Requisitos de largura de banda por aplicação	30

### CAPÍTULO 2: ATM Conceitos

Figura 2.1 -	Interfaces ATM privadas e públicas	39
Figura 2.2 -	Formato de uma célula ATM	39
Figura 2.3 -	Canais virtuais dando suporte a múltiplas aplicações	40
Figura 2.4 -	Exemplo do modo de transferência assíncrono	41
Figura 2.5 -	Exemplo de como as células são transportadas no ATM	43
Figura 2.6 -	Exemplo típico de translação dos campos VPI/VCI, em uma ligação fim-a-fim	44
Figura 2.7 -	Exemplo de uma típica conexão em redes ATM	45
Figura 2.8 -	Células roteadas por seus VPs	46

Figura 2.9 -	Exemplo de roteamento de células empregando os campos VPI e VCI	46
Figura 2.10 -	Exemplo de roteamento por canal virtual	47
Figura 2.11 -	Canais virtuais criados sobre o caminho destino	47
Figura 2.12 -	Canal virtual criado sobre o caminho origem	48
Figura 2.13 -	Os três formatos de endereços ATM	51
Figura 2.14 -	Representação típica IP clássico sobre ATM	53
Figura 2.15 -	Pacote de consulta ARP	54
Figura 2.16 -	Pacote de resposta ARP	54
Figura 2.17 -	Arquitetura do plano de controle	56
Figura 2.18 -	Sinalização AAL (SAAL)	57
Figura 2.19 -	Setup de uma chamada ponto-a-ponto	59
Figura 2.20 -	Liberação de uma chamada ponto-a-ponto	60
Figura 2.21 -	Setup de uma chamada ponto-para-multiponto	61
Figura 2.22 -	Ilustração das fontes de atraso	63
Figura 2.23 -	Modelo de um terminal equivalente	69
Figura 2.24 -	Modelo de referência QoS fim-a-fim	70
Figura 2.25 -	Ilustração dos principais parâmetros de tráfego e, redes ATM	72
Figura 2.26 -	Exemplo de UPC empregando Leaky Bucket	74
Figura 2.27 -	Exemplo de UPC utilizando janelas	75
Figura 2.28 -	Exemplo patológico das diferenças dos UPCs	76

### CAPÍTULO 3: A chave ATM

Figura 3.1 -	Suite de protocolos de sinalização	78
Figura 3.2 -	Formato de uma mensagem Q.2931	79
Figura 3.3 -	Formato de uma mensagem BISUP	80
Figura 3.4 -	Formato de uma célula OAM	83
Figura 3.5 -	Característica de uma chave ATM sob o ponto de vista do usuário	85

Figura 3.6 -	Representação em diagrama de blocos dos cinco módulos funcionais de uma chave ATM padrão	86
Figura 3.7 -	Diagrama de blocos dos componentes funcionais do módulo de entrada de uma chave ATM	90
Figura 3.8 -	Formato de um frame SONET	91
Figura 3.9 -	Diagrama esquemático do container SONET/SDH	92
Figura 3.10 -	Células ATM	93
Figura 3.11 -	Diagrama de estados do bloco delineador de células	94
Figura 3.12 -	Distribuição das funções do Módulo de Entrada	96
Figura 3.13 -	Diagrama esquemático do bloco de processamento de células	97
Figura 3.14 -	Diagrama de blocos do Módulo de Saída de uma chave ATM	100
Figura 3.15 -	Diagrama esquemático do bloco funcional de processamento de células	101
Figura 3.16 -	Diagrama esquemático do bloco funcional de convergência da transmissão	103
Figura 3.17 -	Diagrama esquemático do bloco funcional SONET	104
Figura 3.18 -	Diagrama funcional das funções da Estrutura de Célula da Chave	106
Figura 3.19 -	Diagrama esquemático da abordagem totalmente interconectada	108
Figura 3.20 -	Diagrama esquemático de uma rede Banyan	109
Figura 3.21 -	Exemplo de bloqueio interno	110
Figura 3.22 -	Posicionamento de buffers	111
Figura 3.23 -	Diagrama esquemático do bloco funcional de Controle e Acesso a Conexão	114
Figura 3.24 -	Diagrama funcional do Módulo de Gerenciamento	117
Figura 3.25 -	Paradigma SNMP	118

## CAPÍTULO 4: Proposta para Simulação

Figura 4.1 -	Diagrama de blocos da chave Starlite	121
Figura 4.2 -	Multicast na chave Starlite	122
Figura 4.3 -	Estrutura de roteamento da chave St. Louis	123
Figura 4.4 -	Diagrama de blocos da estrutura de roteamento da chave St. Louis	124
Figura 4.5 -	Estrutura organizacional de uma chave baseada no princípio Knockout	125
Figura 4.6 -	Interface do barramento Knockout	126
Figura 4.7 -	Módulos Multicast adicionados a chave Knockout	127
Figura 4.8 -	Arquitetura da TBSF	128
Figura 4.9 -	Estrutura básica da TBSF	129
Figura 4.10 -	Exemplo do algoritmo bit mask	130
Figura 4.11 -	Exemplo de uma rede de cópia 8x8 não bloqueada	131
Figura 4.12 -	Rede de cópia 8x8 não bloqueada com controle adaptativo	132
Figura 4.13 -	Estrutura lógica da CRAN	133
Figura 4.14 -	Arquitetura de chave ATM com buffer de saída MOBAS	134
Figura 4.15 -	Exemplo de cópia de células multicast na MOBAS	135
Figura 4.16 -	Arquitetura de uma chave multicast Clos-Knockout	136
Figura 4.17 -	Estrutura do modo de classificação	136
Figura 4.18 -	Módulo do segundo estágio da chave, elemento de comutação Knockout sem buffer	137
Figura 4.19 -	Estrutura modular do terceiro estágio da chave Clos-Knockout	138
Figura 4.20 -	Duplicação de endereço multicast em chaves ATM com buffer compartilhado	139
Figura 4.21 -	Pesquisa em filas de chaves ATM com buffer compartilhado	140

Figura 4.22 -	Multicast em chaves ATM com buffer compartilhado no CAM	141
Figura 4.23 -	Multicast em chaves ATM com buffer compartilhado	142
Figura 4.24 -	Arquitetura básica de uma chave multicast	142
Figura 4.25 -	Arquitetura proposta por Urushidani, S. et al. 1998	143
Figura 4.26 -	Estrutura interna do elemento de comutação	144
Figura 4.27 -	Diagrama de blocos da arquitetura proposta neste trabalho	148
Figura 4.28 -	Representação em diagramas de blocos do módulo de roteamento	149

#### CAPÍTULO 5: Simulador e resultados da simulação

Figura 5.1 -	Diagrama de blocos da relação existente entre as diversas classes que compõem o simulador	152
Figura 5.2 -	O status de uma célula ATM (referência para o simulador)	153
Figura 5.3 -	O cabeçalho de uma célula ATM (referência para o simulador)	154
Figura 5.4 -	Representação de uma célula ATM utilizada para testes de simulação	155
Figura 5.5 -	Representação esquemática do vetor GTP utilizado como referência para geração dos tipos de células pelo simulador	156
Figura 5.6 -	Representação esquemática do módulo hardware visto por fora como uma caixa preta	157
Figura 5.7 -	Representação das portas de entrada e saída em um objeto do tipo MODULO_HARD	158
Figura 5.8 -	Tabela de roteamento interna a cada objeto do tipo MODULO_HARD_DEF_ENTRA_SAI, referência para o roteamento das células entre as portas de entrada e saída de um mesmo objeto	159

Figura 5.9 -	Tabela de roteamento interno destacando as regiões com prioridade no atendimento a suas solicitações	160
Figura 5.10 -	Representação das regiões da estrutura de roteamento proposta	162
Figura 5.11 -	Simulação unicast livre de controle dos dados gerados	164
Figura 5.12 -	Representação da porcentagem de perdas de células em relação às células úteis geradas	165
Figura 5.13 -	Simulação unicast com a imposição de que todas as portas de entradas sejam solicitadas por células unicast	166
Figura 5.14 -	Representação da porcentagem de perdas de células em relação às células úteis que foram geradas	167
Figura 5.15 -	Simulação multicast livre de restrições quanto a entrada de dados	168
Figura 5.16 -	Porcentagem de perdas de células unicast e multicast em relação a quantidade total de células que deveriam atravessar a estrutura de roteamento	169
Figura 5.17 -	Simulações multicast com a restrição de que toda entrada de células úteis deve conter um e apenas um pedido multicast	170
Figura 5.18 -	Representação da porcentagem de perdas de células unicast e multicast em função do total de células que deveriam percorrer a estrutura	171
CAPÍTULO 6: Conclusão e proposta de trabalhos futuros		
Figura 6.1 -	Proposta de modificação na estrutura de roteamento	174

## LISTA DE TABELAS

### CAPÍTULO 1: Evolução Histórica Padronização e Tecnologias Alternativas

Tabela 1.1 -	As classes AAL	27
Tabela 1.2 -	A funções do B-ISDN	28

### CAPÍTULO 2: ATM Conceitos Fundamentais

Tabela 2.1 -	Mensagens de sinalização entre host e servidor ATMARP	54
Tabela 2.2 -	Classes QoS definidas pelo ATM Forum e especificadas pela UNI 3.1	64

### CAPÍTULO 3: A chave ATM

Tabela 3.1 -	Mensagens de sinalização de acesso para chamada (controle de conexão)	81
Tabela 3.2 -	Mensagens BISUP para chamadas (controle da conexão)	82
Tabela 3.3 -	Código das células OAM	83

### CAPÍTULO 5: Simulador e resultados da simulação

Tabela 5.1 -	Simulação unicast	164
Tabela 5.2 -	Simulação unicast com todas as portas sendo solicitadas a todo instante	166
Tabela 5.3 -	Simulação multicast	168
Tabela 5.4 -	Simulação multicast limitando a uma solicitação multicast por ciclo	170

## RESUMO

Este trabalho aborda o problema decorrente da necessidade de realizar tráfego *multicast* em redes ATM. Para isto, foi proposta uma arquitetura para a estrutura de célula da chave, mais especificamente a estrutura de roteamento de células no interior da chave ATM. Como o ATM se propõe a atender a todas as exigências do B-ISDN, a necessidade essencial da chave ATM, é que o transporte de células entre suas portas de entrada e saída se faça o mais rápido possível. Para se conseguir isso, a arquitetura proposta não utiliza *buffer* nos seus elementos de roteamento. O trabalho não implementa no todo a chave ATM, deixando em aberto o problema da perda de seqüência de células no interior da estrutura de roteamento, que seria uma responsabilidade do módulo de saída da chave.

A arquitetura proposta foi modelada e um simulador foi construído para a realização dos testes. Para as simulações foram levados em conta os tráfegos de células *unicast* e *multicast*. Como as células *unicast* e as cópias das células *multicast* disputavam por um mesmo *link* de saída, foi proposta uma estratégia para separar estas células. Isto resultou no projeto de posicionar a rede de cópia de células no segundo estágio da estrutura de roteamento, deixando o primeiro estágio para o roteamento das células *unicast*.

O resultado disso foi um segundo estágio agora destinado para as cópias de células *multicast* absolutamente descongestionado, reduzindo drasticamente a possibilidade de disputa por um *link* de saída. Além disso, a estrutura incorpora os conceitos de TBSF (*Tandem Banyan Switch Fabric*), e duplicidade de rota, que tornam a estrutura extremamente modular e flexível. Para atender à preocupação em separar o tráfego, foi introduzido entre um estágio e outro da rede Banyan, uma saída de fuga, permitindo que células que tenham atingido a porta de saída desejada possam deixar a estrutura. São apresentados os resultados da simulação, objetivando a validação do trabalho proposto.

## ABSTRACT

This thesis addresses the problem of accomplishing multicast traffic in ATM networks. An architecture is proposed for the cell structure of the switch, more specifically the routing of cells inside an ATM switch. Because ATM tries to satisfy all the demands of the B-ISDN model, an ATM switch must transport cells between its input and output ports as fast as possible. To achieve that, the proposed architecture does not use buffers in its commutation elements. This work does not implement a complete ATM switch, leaving the problem of the loss of sequence of cells inside the routing structure open. This problem must be solved by the output module of the switch.

The proposed architecture was modeled and a simulator was built for testing purposes. For the simulations, both the unicast and the multicast traffic were taken into account. As the unicast cells and the copies of the multicast cells disputed for the same output link, a strategy was proposed to separate these cells. This resulted in the positioning of the cell copy network in the second stage of the routing structure, leaving the first stage for the routing of unicast cells. The result was a very free second stage used for copying multicast cells. This new stage reduced the contention for an exit link. The structure incorporates the concepts of TBSF, and route duplicity, which make it extremely modular and flexible. To separate the traffic, it was introduced between the two stages of the Banyan Network an escape exit, allowing cells that have reached the correct output port to leave the structure.

## INTRODUÇÃO

Na atual “Era da Informação”, acredita-se que a necessidade de comunicação da sociedade é muito maior que aquela que é provida pelos atuais meios de comunicação.

O crescimento das telecomunicações, porém, não deverá se constituir simplesmente de um aumento da qualidade dos recursos já existentes, mas envolverá também uma mudança radical nos próprios meios que conduzem a informação. De fato, as atuais redes de comunicação apresentam um defeito bastante grave: elas são muito especializadas, ou seja, existe uma rede específica para comunicação de voz (a telefonia), para comunicação de caracteres (telex), para dados de computador (Internet, serviços X.25, etc.), para televisão (cabo, ondas de rádio ou via satélite), etc. A especialização é desvantajosa por várias razões, como: ineficiência na utilização de recursos (uma rede telefônica ociosa não pode ser utilizada para a transmissão de imagens) e falta de flexibilidade (por exemplo, uma transmissão de TV em alta definição exige uma mudança quase que completa na rede atual de televisão). Assim, é necessário achar-se um meio de se transmitir a informação em todas as formas acima (voz, dados e imagem) numa rede única.

Nesse contexto, surgiu o B-ISDN, ou *Broadband Integrated Services Digital Network*. Um dos protocolos fundamentais do B-ISDN é o ATM (*Asynchronous Transfer Mode*) que se prevê, será largamente utilizado em todo o mundo nos próximos anos segundo Prycker (1994).

Uma característica que diferencia bastante a rede ATM das redes atuais é que, no ATM, não existe um barramento comum (como no caso da Ethernet) ou um anel comum (como nos casos do FDDI e do Token Ring). Esse barramento (ou anel) comum, limita a taxa de transmissão total possível ao valor máximo que o meio consegue transmitir. Além disso, a velocidade máxima é atingida na troca de dados entre duas estações apenas; nesse caso, se outras duas estações quiserem trocar dados entre si, a largura de banda total da rede deverá ser dividida entre os dois pares, e, no melhor caso, cada um dos pares será capaz de utilizar apenas a metade da capacidade total da rede. Se outros pares tentarem se comunicar, a situação piora

ainda mais com cada um dos pares tendo acesso a apenas uma fração da capacidade total do meio. Assim, a velocidade de transmissão entre duas estações quaisquer depende da quantidade de comunicação que a rede está transmitindo naquele determinado momento. Essa é claramente uma situação indesejável no caso de transmissão de imagem e voz, pois então, ter-se ia que a qualidade da transmissão dependeria da utilização instantânea da rede (num momento a imagem poderia ser boa, enquanto que, no momento seguinte ela degradaria).

O ATM resolve esses problemas determinando que, entre duas estações quaisquer, deve ser garantida uma determinada largura de banda, independentemente da carga de comunicação instantânea da rede. A ligação é então, feita na topologia em estrela, de forma que todas as estações se concentrem a um nó central, que faz o chaveamento dos pacotes ATM. Esse nó central recebe o nome de chave. A chave ATM tem características diferentes tanto das chaves tradicionais de telefonia quanto das chaves utilizadas em computadores paralelos. No contexto da chave ATM, um dos pontos que tem despertado muita atenção dos pesquisadores nos últimos anos é a realização de *multicast*. Neste contexto, as arquiteturas das chaves podem ser divididas em dois grandes grupos: o primeiro chamado de árvore *multicast* e o segundo grupo chamado de *broadcast*.

As chaves baseadas no princípio árvore de *multicast*, consistem basicamente de dois estágios, sendo o primeiro uma rede de cópia, onde as células ATM são replicadas de tal forma a atender às solicitações *multicast*, e o segundo, uma rede de roteamento, com o propósito de rotear as células *unicast* e *multicast* até as portas de saída desejadas. Nas chaves tipo *broadcast*, cada célula que entra na estrutura de roteamento é enviada em *broadcast* para todas as portas de saída, e filtros são empregados para retirarem da estrutura de roteamento apenas as células com destino à porta de saída desejada, sendo descartadas as cópias feitas em excesso.

Neste trabalho, propomos uma arquitetura de chave do tipo árvore de *multicast*. O trabalho está organizado da seguinte forma: O capítulo 1 trata da evolução histórica, da padronização e das tecnologias de redes alternativas, evidentemente, o leitor inteirado no assunto, pode ignorar o capítulo e partir para o seguinte, mesmo porque sua última atualização ocorreu no ano de 1996, não sendo alterado desde então. Na seqüência, o capítulo 2, estuda o ATM e seus conceitos

fundamentais, como requisitos básicos para a evolução teórica e prática do trabalho. O capítulo 3 estuda a chave ATM, sua constituição e forma de integração entre seus diversos módulos. Em particular o módulo de estrutura de célula da chave (CSF), responsável pelo roteamento de células no interior da chave, é o módulo que constitui a base para a proposta deste trabalho, devido à sua importância e por ser um gargalo a ser superado na tentativa de transpor os obstáculos impostos pelo ATM, de se conseguir velocidades altíssimas a taxas de perdas de células incrivelmente baixas.

No capítulo 4, foi feito um levantamento bibliográfico, onde procurou-se extrair da literatura existente, a importância, os problemas intrínsecos e as metodologias adotadas para a solução dos problemas. Neste capítulo faz-se também a apresentação da arquitetura proposta e objeto de discussão dos capítulos seguintes. Cabe ressaltar que, além das bibliografias diretamente citadas (que influenciaram no trabalho), muitas outras foram consultadas, e auxiliaram a compor o quadro de estudos ora apresentado. O capítulo 5, apresenta o simulador e os resultados das simulações computacionais. Por fim, no capítulo 6, são apresentados as conclusões e discussões sobre as propostas de continuidade do trabalho.

# Capítulo 1

*Este capítulo apresenta um breve histórico a respeito da evolução das redes de computadores, revisa os conceitos de transmissão assíncrono, síncrono, multiplexadores e chaves. Apresenta o ATM e sua padronização. Descreve o protocolo B-ISDN e finaliza com uma exposição das tecnologias alternativas.*

## **Evolução Histórica Padronização e Tecnologias Alternativas**

### **1.1 Histórico**

Durante três décadas, temos visto a indústria de computadores evoluir a passos largos. Inicialmente, todo o processamento era executado por computadores de grande porte, utilizando o processamento em lote. O conceito de conectividade já existia naquela época, sendo que, por volta de 1957, com a criação da *Advanced Reserch Projects Agency* (ARPA), pelo presidente americano Dwight D. Eisenhower (1953 – 1961), com o objetivo de desenvolver alta tecnologia para as forças armadas, tem-se o início da revolução da conectividade.

Em 1962, a ARPA dá início à criação de um método com o propósito de assegurar as comunicações governamentais mesmo em caso de um ataque nuclear. A princípio a comunicação seria feita por comutação de pacotes, utilizando uma rede composta por diversos caminhos redundantes, ligando os diversos nós da rede. Sendo assim, seria necessário destruir praticamente toda rede para impedi-la de funcionar.

Por volta de 1968, Larry Roberts, Ivan Sutherland e Bob Taylor, membros da ARPA, convocam quatro universidades (Stanford, Berkeley, UCLA e Universidade de Utah) para implementar a rede de pacotes, a qual receberia o nome de ARPANET. Em 1969, a ARPANET já era uma realidade, operando com um equipamento chamado *Interface Message Processor* (IMP), o IMP, construído pela *Bolt, Berabek and Newman* (BBN), utilizando um protocolo chamado *Network Control Protocol* (NCP). O interessante é que,

paralelo a isso, tem-se no mesmo ano a criação do UNIX, que seria fundamental para a transformação da ARPANET em Internet.

A ARPANET cresceu, ganhando na década de 70 dois grandes propulsores: o TCP/IP, proposto por Vinton Cerf e Bob Kahn, que possibilitou o endereçamento de quatro bilhões de máquinas em lugar dos 256 endereços possíveis no NCP, e o Ethernet de Bob Metcalfe, firmando-se como um método simples e barato para ligar computadores a pequenas distâncias. Paralelamente a este avanço, a indústria eletrônica consegue a cada dia um grau maior de integração, permitindo que as máquinas se tornem mais baratas e compactas, viabilizando ainda mais o crescimento da ARPANET.

Na década de 80, muitas redes surgiram, redes acadêmicas como a *Because Its Time Network* (BITINET), financiada pela IBM para utilizar os protocolos NJE da empresa. Esta rede ligava a *City University* de Nova York, com a Universidade de Yale. Surgiram também redes militares como a MILNET, e uma extensão da MILNET foi criada na Europa, recebendo o nome de MINET; duas redes de satélite a SATNET e WIDENET; a CSNET uma rede acessível a todos os departamentos de computação dos EUA; a TELENET aberta a toda a comunidade, e outras. É também desta época o modelo das sete camadas *Open Systems Interconnection* (OSI), conhecido como modelo OSI, resultado de uma proposta da *International Standards Organization* (ISO), como um primeiro passo na padronização internacional dos protocolos, que se multiplicavam juntamente com as novas redes.

A consolidação da rede acontece em 1984, quando a *National Science Foundation* (NSF) assume a responsabilidade pela manutenção da ARPANET. Em 1986 são criados cinco centros de supercomputação, com o objetivo de aumentar a largura de banda disponível, e a conexão destes centros recebeu o nome de NSFNET, que passou então a ser conhecido como a espinha dorsal da ARPANET. A rede cresce a passos largos, com a liberação do acesso ao grande público e apoiada pelo surgimento dos microcomputadores e protocolos como *Novell Netware* e *Banyan Vines* específicos para micros.

A grande expansão da ARPANET chamou a atenção para as limitações aos serviços por ela prestados, sobretudo serviços envolvendo voz, vídeo de alta definição e dados de alta velocidade. Isto levou o ITU-T conforme descrito por Kim; Suda (1994) a definir em 1985 o conceito *Broadband Integrated Services Digital Network* (B-ISDN), através da recomendação I.121 aprovada em 1988, com o objetivo de integrar serviços de dados voz e imagem em um mesmo meio e também servir como padrão para todos os meios de comunicação. Em 1990, treze recomendações para o B-ISDN foram propostas e o *Asynchronous Transfer Mode* (ATM) foi escolhido para implementar o B-ISDN.

Em 1990, depois de dividida em MILNET e NSFNET, a ARPANET foi formalmente encerrada. A NSFNET expandiu englobando empresas, nascendo daí a Internet compreendendo 1.500 sub-redes e 250 mil hosts. Na década de 90, estimulado pelo vice-presidente americano Al Gore (1993 - ), com a promulgação da Lei de Infra-estrutura Nacional de Informação (NII), conhecida pelo nome de superestrada da informação, a Internet explode em âmbito mundial.

Em fevereiro de 1996, o presidente Willian J. Clinton (1993 - ) sanciona a nova Lei de Telecomunicações americana, que libera por completo o setor. A Internet cresce e se diversifica, e já ultrapassa seis milhões de servidores e 50 mil sub-redes.

## 1.2 O mundo das telecomunicações

As redes de telecomunicações, segundo Prycker (1994), são caracterizadas pela especialização, o que significa que para cada serviço de telecomunicação, existe ao menos uma rede para atender suas necessidades de transporte. Neste contexto se enquadram:

- As redes de telex, onde os caracteres são transportados a uma velocidade da ordem de 300 bps;
- *Plain Old Telephone Service* (POTS) transporte realizado por *Public Switched Telephone Network* (PSTN). Esta rede oferece aos clientes tráfegos de voz bi-direcional;
- O transporte de dados se divide em duas categorias: *Packet Switched Data Network* (PSDN), baseada no protocolo X.25, e em menor número as *Circuit Switched Data Network* (CSDN), que utilizam o protocolo X.21;
- Os sinais de TV são transportados de três formas: broadcast<sup>1</sup> via ondas de rádio (*wireless*); cabo coaxial/fibra ótica *Community Antenna TV* (CATV); e os atuais sistemas de *broadcast* via satélite *Direct Broadcast System* (DBS);
- No domínio privado, os dados são transportados pelas *Local Area Networks* (LANs).

---

<sup>1</sup> **Broadcast:** transmissão de dados, voz ou vídeo para vários receptores (difusão)

Cada uma destas redes foram projetadas para atender a um serviço específico, não sendo, portanto, adequadas para o transporte de outro tipo de serviço. Por exemplo, uma rede CATV não permite o transporte de serviços telefônicos (POTS); também, PSTN não transporta sinais de TV.

Uma consequência importante da especialização destes serviços é a existência de um grande número de redes independentes, cada uma com seus próprios requisitos de projeto, construção e manutenção. Outra consequência desta particularidade, é a impossibilidade de alocar recursos, de uma rede com largura de banda disponível, para uma rede com carência de recursos para atender seus serviços. Por exemplo: o horário de pico das redes telefônicas está entre 9:00 e 17:00 horas, ao passo que, os horários de pico das redes CATV acontecem no período noturno. Como é impossível o compartilhamento de recursos destas redes, cada uma deve ser dimensionada para atender suas necessidades de tráfego para o pior caso.

O resultado desta especialização são listados a seguir:

- **Dependência do serviço:** cada rede é capaz de transportar o serviço para o qual foi projetada. É possível em alguns casos, utilizando equipamento adicional (por exemplo: modem<sup>2</sup>) realizar adaptações e atender de forma, mesmo que precária, outros serviços;
- **Inflexibilidade:** a evolução dos algoritmos de compressão de dados, voz, áudio e vídeo e o avanço na tecnologia de integração de circuitos em *Very Large Scale Integration* (VLSI), podem influenciar a taxa de bit gerada por certos serviços, o que mudaria a exigência dos recursos para atender estes serviços. Num futuro próximo, serviços com requisitos de recursos desconhecidos poderão surgir Prycker (1994): Por exemplo, ainda não está claro quais os requisitos em termos de taxa de bits para a *High Definition TV* (HDTV) Prycker (1994). Uma rede especializada terá dificuldades em se adaptar para atender às novas exigências;

---

<sup>2</sup> **Modem:** dispositivo que permite que dados sejam enviados via telefone através da conversão de sinais binários de um computador em sinais analógicos.

- **Ineficiência:** os recursos disponíveis são usados ineficientemente. Uma rede com escassez de recursos, não pode utilizar, mesmo se disponíveis, os recursos de outra rede .

O primeiro passo na tentativa de resolver este problema, foi a criação da *Narrowband ISDN* (N-ISDN), com a proposta de se tornar uma rede universal que atendesse todas as necessidades de transporte de dados e voz em um mesmo meio. Como esta rede não poderia atender às necessidades do transporte de sinais de TV, devido à sua limitada largura de banda, a necessidade de uma rede especializada para atender estes serviços continuaria existindo e o problema não seria de todo resolvido.

Neste contexto, foi proposto o conceito de *Broadband ISDN*, onde uma rede universal ATM atenderia os requisitos de todos os serviços, integrando em um único meio os serviços de dados, voz e vídeo, possibilitando uma padronização de todos os meios de comunicação, segundo Lagerstedt, Nyman (1993).

Em contraste com as desvantagens apresentadas acima, uma rede universal teria as vantagens apresentadas a seguir:

- **Flexibilidade:** os avanços no estado da arte dos algoritmos de codificação e nas tecnologias de integração (VLSI), poderão reduzir a largura de banda requisitada pelos tele-serviços existentes, conforme Prycker (1994). Uma rede capaz de transportar todos os tipos de serviços será capaz de adaptar-se às novas mudanças;
- **Eficiência:** todos os recursos disponíveis podem ser compartilhados entre todos os serviços;
- **Econômico:** a existência de uma rede única proporcionará uma redução nos custos de projeto, manutenção, construção e operação das redes.

### 1.3 Métodos de transmissão de dados

Os métodos de transmissão de dados digitais são frequentemente classificados em síncronos ou assíncronos .

### 1.3.1 Transmissão de dados assíncrona e síncrona

Em transmissão assíncrona para cada carácter a ser transmitido, utiliza-se um elemento de sinalização que indica o início do carácter (bit de *START*) e outro para indicar o término do carácter (bit de *STOP*), a figura 1.1 apresenta um exemplo de transmissão assíncrona de um caractere ASCII<sup>3</sup>.

O bit de *START* é utilizado para sincronizar o receptor para que ele possa ler cada bit no momento apropriado. O termo Assíncrono refere-se à irregularidade dos instantes de ocorrência dos caracteres, ou seja, o tempo decorrido entre dois caracteres (tempo de repouso) pode ser variado pelo equipamento transmissor sem que o equipamento receptor tome conhecimento.

Usualmente a transmissão de caracteres assíncrona opera a baixas velocidades (da ordem de 9600 bps) e utiliza equipamentos de baixo custo segundo Prycker (1994). Porém a má utilização do canal, já que os caracteres são transmitidos irregularmente no tempo, e o alto *overhead*<sup>4</sup>, ocasionam uma baixa eficiência na transmissão.

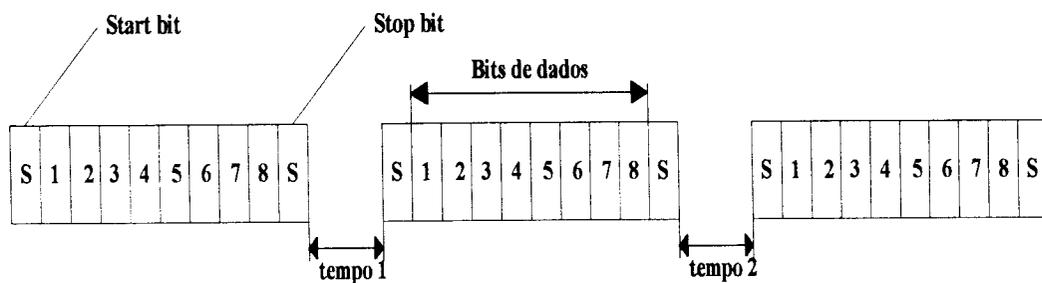


Figura 1.1: Transmissão de caracteres no modo Assíncrono

A transmissão de dados síncrona requer um sinal de *clock*<sup>5</sup> associado com os bits transmitidos, sendo a transmissão realizada em blocos em uma cadência fixa. Isto implica que tanto o emissor quanto o receptor devem estar sincronizados. A figura 1.2 a seguir mostra uma típica fila de dados síncrona. A mensagem começa

<sup>3</sup> ASCII: código que representa caracteres alfanuméricos em código binário.

<sup>4</sup> Overhead: bits de controle adicionais à informação.

com dois caracteres de sincronização (SYNC) e um caracter de início da mensagem (SOM). O caracter de controle (C) especifica o tipo de dado ou a mensagem que está sendo enviada. O campo CRC, é utilizado para realizar a verificação de erros, e o caracter de fim de mensagem (EOM), sinaliza o fim da transmissão.

SYNC	SYNC	SOM	C	Dados	CRC	EOM
------	------	-----	---	-------	-----	-----

**Figura 1.2:** *Formato de um Frame Síncrono*

### 1.3.2 Modo de transferência assíncrono versus síncrono

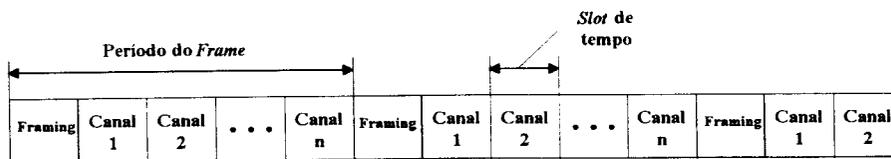
Os conceitos básicos que diferenciam o *Synchronous Transfer Mode* (STM) do *Asynchronous Transfer Mode* (ATM) podem ser visualizados na figura 1.3.

A figura 1.3(a) ilustra uma fila STM onde cada *slot* de tempo representa uma porção da largura de banda reservada para um único canal. Cada frame contém *n* *slots* de tempo dedicados; (por exemplo *n* é 192 no DS1). O campo de *overhead* é utilizado para identificar o frame STM. Como os *slots* de tempo estão reservados independentemente para cada canal, se o canal não esta transmitindo por um motivo qualquer, o *slot* de tempo permanece reservado, dedicado para o canal. Caso outro canal possua mais dados para transmitir, ele deve esperar até que seu *slot* de tempo esteja novamente disponível. A consequência da existência de *slots* de tempo vazios é uma baixa utilização dos recursos disponíveis na rede.

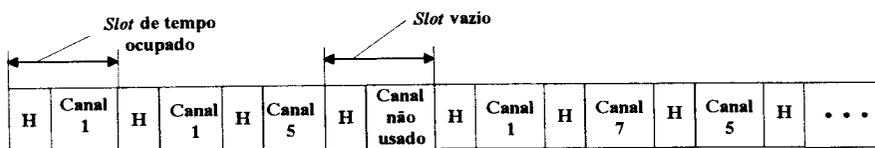
O ATM emprega uma abordagem completamente diferente onde existe um campo fixo *header*<sup>6</sup> para cada *payload*<sup>7</sup>, como mostrado na figura 1.3(b), que identifica o canal ou caminho virtual, permitindo que o *slot* de tempo esteja disponível para qualquer usuário que realmente possua dados a transmitir. Caso nenhum usuário possua células para transmitir, células vazias ou *idle* são enviadas. Esta flexibilidade do ATM resulta em um modelo de tráfego mais eficiente quando comparado com o STM.

<sup>5</sup> **Clock:** sinal utilizado para sincronização de circuitos eletrônicos.

<sup>6</sup> **Header:** cabeçalho para fornecer informação sobre destinos, rotas e referente ao resto dos dados.



a. Multiplexação por divisão de tempo síncrona (STM)



b. Multiplexação por divisão de tempo assíncrona (ATM)

Figura 1.3: Ilustração da multiplexação STM e ATM

## 1.4 Multiplex e chaves

Existe uma íntima relação entre multiplex e chaves.

Multiplex define um meio pelo qual múltiplos canais de informação podem compartilhar um meio físico comum. Em outras palavras, a função básica do multiplex é compartilhar uma saída com múltiplas entradas, sendo a função inversa realizada pelo demultiplex. As chaves atuam na informação contida em um meio de transmissão multiplexado, e redirecionam esta informação de suas portas de entrada para a porta de saída apropriada.

### 1.4.1 Multiplexação por divisão de espaço (SDM)

Multiplexação por divisão de espaço é o mesmo que separar fisicamente. Uma boa aplicação para o SDM é em aplicações onde se tem múltiplos cabos interconectando equipamentos. Foi muito utilizada nas primeiras redes telefônicas, onde um par de cabos conectava todos que desejavam se comunicar. Esta tecnologia foi rapidamente abandonada, devido à grande expansão das redes

<sup>7</sup> Payload: campo de dados.

telefônicas, tornando impraticável a conexão de todos os usuários por meio de um par de cabos. Outro exemplo de aplicação do STM foi nos antigos terminais de comunicação de dados, onde um cabo separado conectava o computador principal a seus terminais, configuração que também foi abandonada com a expansão das redes de computadores.

#### **1.4.2 Multiplexação por divisão de frequência (FDM)**

Com a evolução dos sistemas de comunicação, descobriu-se que sinais de voz (analógico), poderiam ser multiplexados em frequência, empregando uma técnica chamada modulação de sinal. Através da modulação, um sinal a uma dada frequência (banda base) pode ser trasladado para outra frequência (banda passante) e recuperado no final realizando-se a função inversa, ou seja, a demodulação. Este avanço possibilitou o transporte de múltiplos sinais de voz sobre um mesmo meio, cabo ou ondas de rádio.

Apesar de proporcionar um grande avanço, tanto em eficiência quanto em economia, o problema não foi de todo resolvido, pois a tecnologia ainda apresentava problemas como ruído, distorções e interferência entre os canais que compunham o espectro de frequência da transmissão.

#### **1.4.3 Multiplexação por divisão do tempo (TDM)**

É uma técnica totalmente digital desenvolvida nos anos 50 com o propósito de eliminar os problemas do FDM (ruídos, distorções e interferências): as informações analógicas são convertidas para digitais antes de serem transmitidas. Apesar de um custo inicial muito alto, a substituição dos cabos existentes e escavações de novos túneis seria ainda mais caro. O resultado foi que o TDM se tornou o principal método de multiplexação em todas as modernas redes de telecomunicações. Uma consequência direta é que a transmissão de voz por sistemas TDM são absolutamente independentes da distância, devendo a voz ser digitalizada na transmissão e convertida para a forma analógica na recepção.

Apesar da transmissão de dados ser mais sensível que a transmissão de voz, ela está sendo fortemente beneficiada com o emprego da tecnologia TDM em redes públicas.

#### 1.4.4 Multiplexação por endereço

A multiplexação por endereço foi desenvolvida na época em que as transmissões empregando FDM apresentavam uma qualidade muito baixa. O nome mais comum para multiplexação por endereço é *Asynchronous Time Division Multiplexing* (ATDM). Cada pacote de informação era prefixado com um endereço que devia ser interpretado em cada nó da rede. Cada nó verificava se o pacote havia sido recebido corretamente ou não, e em caso negativo ele re-enviava o pacote para o nó anterior até que ele fosse recebido corretamente. SNA, DECNET e X.25 são alguns exemplos mais antigos de multiplexação por endereço. Exemplos mais recentes são o *Frame Relay* e o ATM.

#### 1.4.5 Funções das chaves ponto-a-ponto

A figura 1.4 ilustra os quatro tipos básicos de conexões ponto-a-ponto que podem ser implementadas por chaves ou multiplex.

Chaveamento por divisão de espaço, entrega um sinal de uma *interface* física para outra *interface* física. Um exemplo típico são as chaves *crosspoint*<sup>8</sup>.

Chaveamento por divisão do tempo, muda a ordem do *slot* de tempo, dentro de uma fila de dados, organizada pelo método de multiplexação, por divisão do tempo (TDM).

Chaveamento de endereços, muda o campo de endereço do pacote de dados, que poderá posteriormente ser multiplexado no espaço, tempo ou frequência.

Chaveamento de frequência, translada um sinal de uma dada frequência de entrada para outra frequência na saída. Este método, normalmente é empregado em fibra ótica, segundo McDysan, Spohn (1994).

---

<sup>8</sup> **Croisspont**: chaves de conexão cruzada.

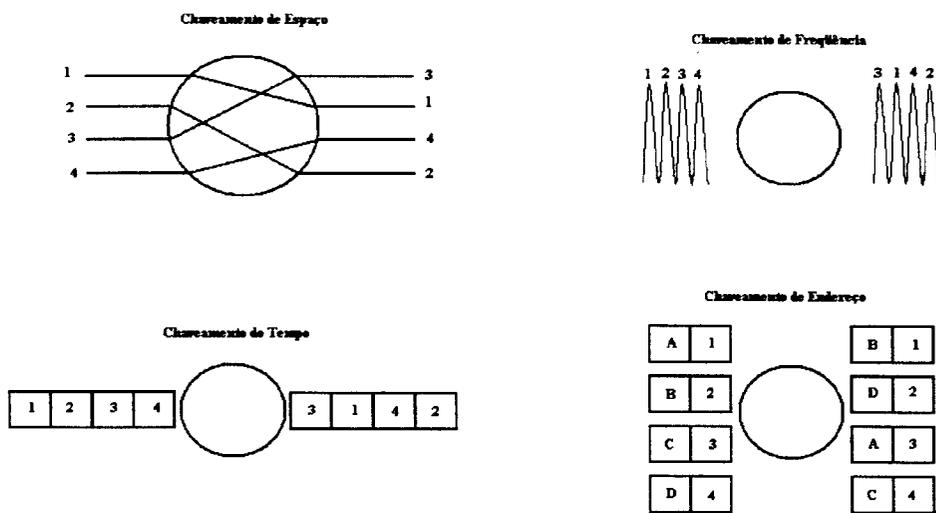
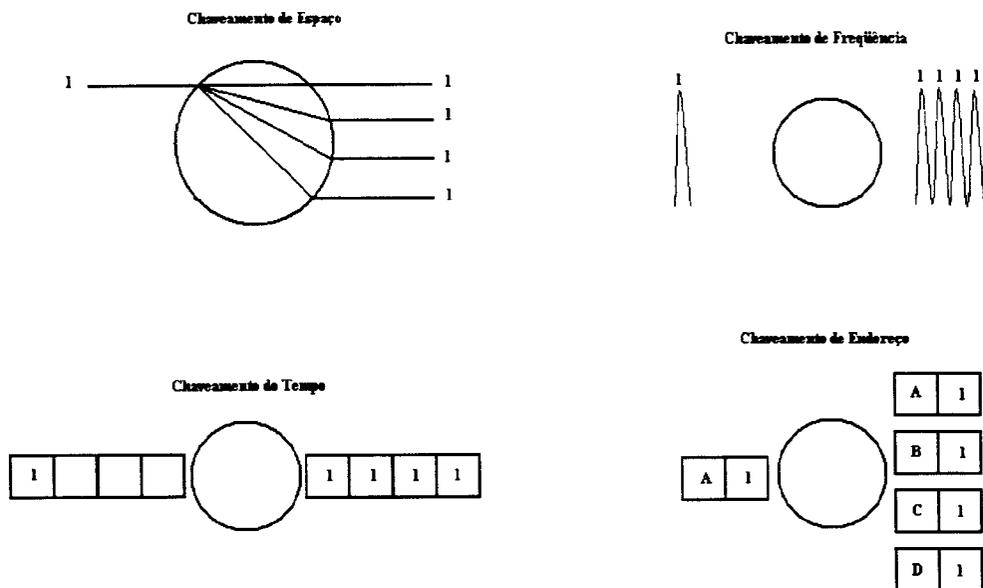


Figura 1.4: Definição das funções de chaveamento ponto-a-ponto

#### 1.4.6 Funções das chaves ponto para multi ponto

É uma extensão das funções das chaves ponto-a-ponto para atender as necessidades de *broadcast* ou *multicast*<sup>9</sup>, como mostra a figura 1.5. Uma chave *broadcast* por divisão de espaço, copia um único sinal de entrada em duas ou mais saídas. Chaves *broadcast* FDM copiam o mesmo sinal de entrada em múltiplas frequências de saída. As chaves *broadcast* TDM, preenchem múltiplos *slots* de tempo de saída, com os dados de uma única entrada. Chaves *broadcast* de endereços, preenchem múltiplos pacotes, com diferentes endereços de saída, com informação proveniente de um único pacote de entrada.

<sup>9</sup> **Multicast:** transmissão de dados, voz ou vídeo para grupos de receptores.



**Figura 1.5:** Definição das funções de chaveamento ponto-para-multiponto

### 1.4.7 Exemplos de chaves

O propósito destes exemplos é ilustrar terminologias e fornecer subsídios para uma boa compreensão dos próximos capítulos.

#### 1.4.7.1 Chaves por divisão de espaço

A figura 1.6 ilustra uma rede simples de conexão cruzada, de duas entradas e duas saídas. As linhas mais espessas ilustram um exemplo de conexão, e a linha pontilhada, os sinais de controle. Observa-se que quatro pontos são necessários para o completo estabelecimento da conexão. As estruturas clássicas de chaves por divisão de espaço têm sido construídas utilizando componentes eletromecânicos e eletrônicos para realizar as funções dos pontos de cruzamento. Novas tecnologias, envolvendo elementos óticos nos pontos de cruzamento, com controle ótico ou eletrônico, estão sendo pesquisadas e desenvolvidas, segundo McDysan, Spohn (1994), Ginsburg (1996).

Grandes chaves por divisão de espaço, são construídas a partir da conexão de múltiplos estágios de redes de conexão cruzada.

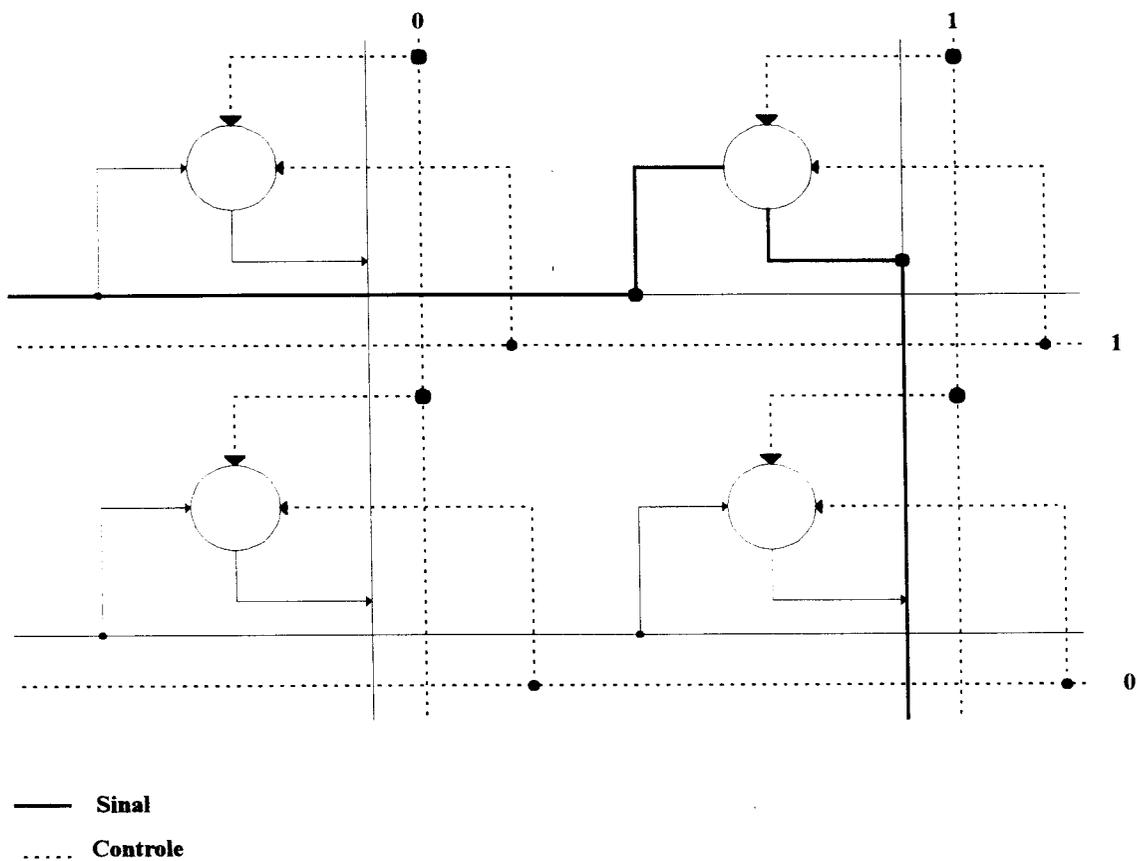


Figura 1.6: Rede de Conexão Cruzada com duas entradas e duas saídas

#### 1.4.7.2 Chaves por divisão do tempo

Os sistemas de telefonia digital empregam chaves por divisão do tempo (TDS), para a interconexão de suas redes, segundo Kaiser (1985), Ginsburg (1996).

A figura 1.7 descreve a operação de uma chave por divisão de tempo. Cada *frame* TDM possui  $M$  *slots* de tempo. O *slot* de tempo de entrada, rotulado como  $I(m)$ , é armazenado no vetor  $x(t)$  na posição  $m$ . O endereço de saída do vetor  $y(t)$ , é percorrido sequencialmente incrementando-se  $t$  de 1 até  $M$  a cada *slot* de

tempo. O endereço contido no vetor  $y(t)$  identifica o índice do *slot* de tempo de entrada  $x(t)$ , que será colocado na saída no *slot* de tempo  $t$ . No exemplo da figura 1.7,  $y(n)$  possui o valor  $m$ , que faz com que o *slot* de tempo  $m$  seja chaveado para a saída no *slot* de tempo  $n$ .

Grandes chaves por divisão de tempo podem ser construídas a partir da interconexão de chaves TDS, por meio de múltiplos estágios de redes de conexão cruzada, conforme Kaiser (1995).

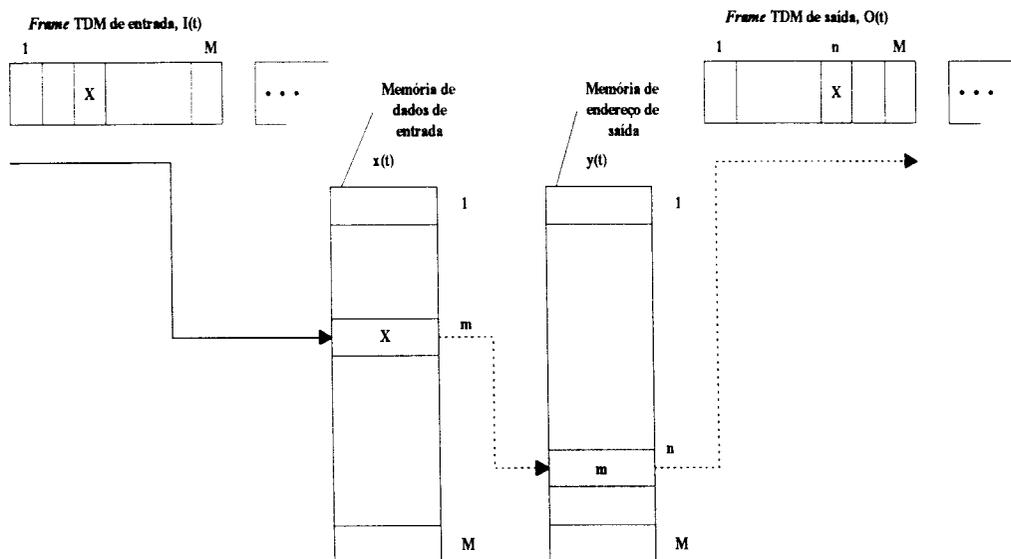


Figura 1.7: Operação de uma chave por divisão de tempo

### 1.4.7.3 Chaveamento por endereços

Atuam sobre uma fila de dados, os quais devem estar organizados em pacotes, divididos em *header* e *payload*, sendo que, no *header* estão as informações de endereço, que serão usadas pela chave em cada nó, para a realização do roteamento. O endereço determina para qual saída física o pacote deve ser enviado. Todas as possíveis topologias de conexão podem ser implementadas com esta arquitetura de chave: ponto-a-ponto, ponto-para-multiponto, multiponto-para-ponto,

e multiponto-para-multiponto. A figura 1.8 ilustra quatro chaves, cada uma com duas entradas e duas saídas.

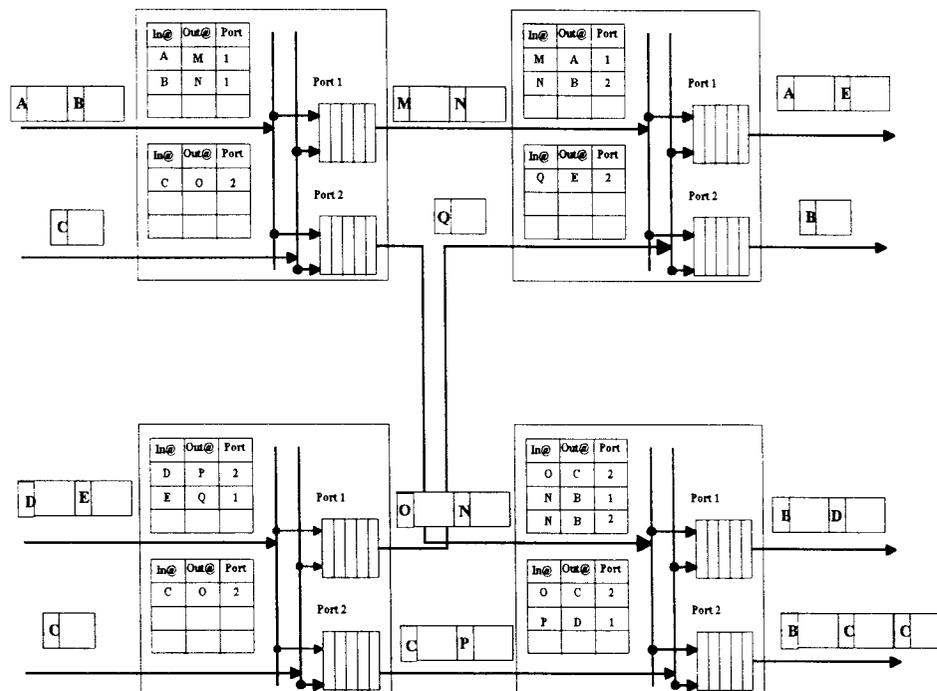


Figura 1.8: Operação de uma chave por endereço

Os pacotes (que podem ser de tamanho fixo ou variável) contendo *header* e *payload*, chegam nos módulos de entrada das chaves. Os endereços de entrada são utilizados para indexar a tabela de roteamento na coluna *In@*, identificar o endereço de saída na coluna *Out@*, e a porta física de saída para a qual o pacote deve ser enviado, na coluna rotulada por *Port*.

Do exemplo acima observa-se que os pacotes A, D e E formam uma conexão ponto-a-ponto. O pacote B forma uma conexão ponto-para-multiponto. E o pacote C forma uma conexão multiponto-para-ponto.

As chaves e multiplex de endereços, são o coração do ATM, conforme McDysan, Spohn (1994).

#### 1.4.7.4 Chaveamento de frequência/comprimento de onda

Trabalhos significativos de redes óticas têm sido apresentados Jajszczyk (1993). O conceito básico que está por trás destas redes é o compartilhamento do meio, no qual uma rede fotônica interconecta um certo número de sistemas finais como mostrado na figura 1.9.

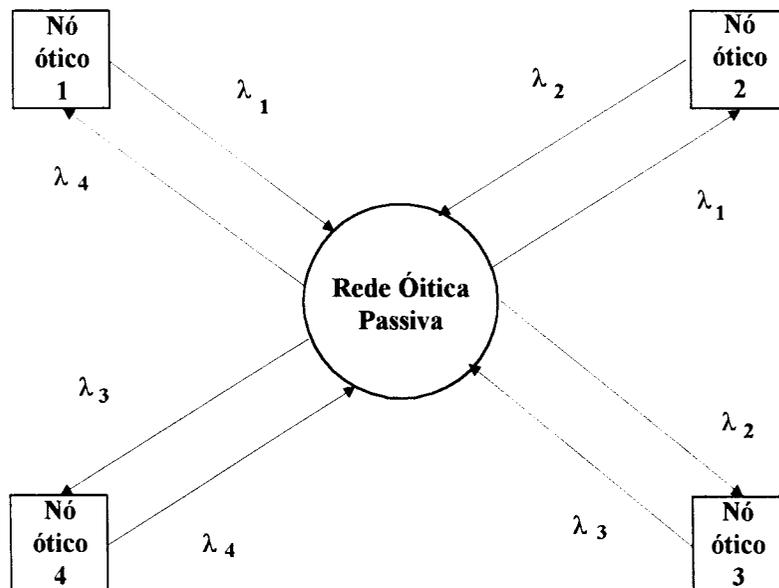
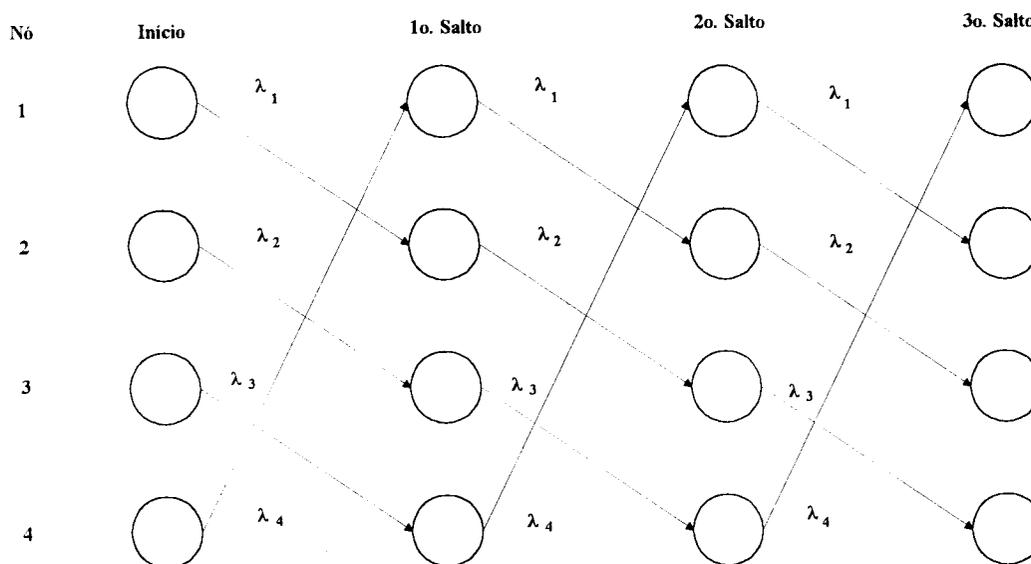


Figura 1.9: Ilustração de uma rede ótica WDM

Os nós do sistema ótico transmitem e recebem, ao menos um comprimento de onda  $\lambda$ . A figura 1.10, representa uma rede ótica com múltiplos caminhos. Cada nó possui autonomia para realizar todas as funções de roteamento. Se um nó recebe um pacote que não se destina a ele, este é retransmitido em seu comprimento de onda, até que o pacote atinja seu destino como mostrado na figura 1.10. No exemplo da figura 1.10 a estação 1 transmite para a estação 4: primeiramente ela envia sobre o comprimento de onda  $\lambda_1$ , que é recebido pelo nó 2. O nó 2 examina o cabeçalho do pacote e verifica se ele não é o destino: não o sendo,

ele é retransmitido com comprimento de onda  $\lambda_2$ . O nó 3 recebe o pacote, examina seu cabeçalho, e o retransmite com comprimento de onda  $\lambda_3$ , que é recebido pelo destino nó 4.



**Figura 1.10:** Ilustração de uma rede WDM de múltiplos saltos

Este processo torna-se muito ineficiente quando o número de nós cresce muito. Pesquisas recentes enfocam projetos com um único caminho. Nestes projetos, transmissor e receptor devem estar sintonizados. Existe a necessidade de algum meio para alocar e compartilhar largura de banda em redes óticas. Inicialmente foi proposto uma sinalização baseada em circuitos, mas constatou-se que o chaveamento de pacotes, e não o chaveamento de circuitos era o mais adequado para realizar o compartilhamento em redes óticas, segundo Green (1992). A alocação de largura de banda fixa em *slot* de tempo, também foi considerada inaceitável. A proposta de escalonamento dinâmico, parece ser a solução para atender a demanda de redes de alta velocidade, conforme Jajszczyk (1993).

## 1.5 ATM: O que é e o porquê

*Asynchronous Transfer Mode* (ATM), é uma tecnologia de rede capaz de transportar voz, vídeo e dados sobre um mesmo meio, tanto nas LANs, quanto nas WANs. Esta universalidade é o porquê do ATM, uma única plataforma e tecnologia capaz de reduzir custos e melhorar a qualidade dos serviços para os usuários finais segundo Chen, Liu (1995).

O ATM é uma tecnologia que trabalha com o conceito de células que equivalem aos pacotes no X.25 e no *Frame Relay*. E entre dois pontos quaisquer de uma rede ATM existirá ao menos uma chave, segundo Chen, Liu (1995). As chaves trabalham recebendo uma célula e redirecionando-a para a próxima chave ou ao destino final.

A velocidade entre um ponto da rede e a chave, e entre chaves, pode variar, isto é, uma mesma chave pode ter portas com velocidades diferentes, permitindo a ligação de um servidor numa porta mais rápida e estações em portas mais lentas, o que permite uma melhor estruturação da rede.

Uma característica importante do ATM é que ele é orientado a conexão, ou seja, quando uma estação se conecta a um servidor ou outra estação, a rede ATM cria uma conexão entre os dois pontos que só terminará quando assim for solicitado.

Ao conceito de comunicação baseado em conexões, alia-se o conceito de Qualidade de Serviço (QoS), segundo Chen, Liu (1995). O estabelecimento de uma conexão fim-a-fim e a definição da QoS, permitem às chaves ATM priorizarem as conexões de uma forma bem mais inteligente. O usuário pode ainda requerer níveis diversos de confiabilidade nos serviços, ou seja, ele pode exigir que as informações tenham ou não garantia de integridade.

O ATM trabalha com unidades de informações de tamanho fixo chamada célula, compostas por 53 bytes, sendo 5 bytes usados para controle (*header*) e os outros 48 bytes para a informação (*payload*). A vantagem de se trabalhar com unidades de informação de tamanho fixo é que ela possibilita a implementação de chaves ATM diretamente no *hardware*, utilizando mecanismos extremamente

simples e rápidos, possibilitando um ganho considerável de velocidade em relação às chaves baseadas em decisões por *software*.

A tecnologia é promissora. Assim como o *Ethernet* se tornou um padrão para a LAN, o ATM tem tudo para se tornar um novo padrão não apenas para LANs, mas também para WANs, sistemas de telefonia, televisão à cabo etc. conforme Ginsburg (1996).

## 1.6 Padronização

Os padrões ATM são o resultado da cooperação de empresas de telecomunicações, fabricantes de *hardware* e *software* e de diferentes corpos de normatização. A idéia é construir um corpo de normas que seja confiável e garanta a interoperabilidade entre todas as organizações. Existem duas classes de padronização e especificações envolvidas no ATM e no B-ISDN: as corporações de padronizações formais e os fóruns industriais.

O corpo de padronização formal é composto por: *International Telecommunications Union-Telecommunications Sector* (ITU-T), também conhecido como *International Telegraph and Telephone Consultative Committee* (CCITT); *American National Standards Institute* (ANSI); *European Telecommunications Standards Institute* (ETSI). Os quatro principais fóruns industriais são: ATM Forum; *Internet Engineering Task Force* (IETF); *Frame Relay Forum*; *Switched Multimegabit Data Service* (SMDS) e *Special Interest Group* (SIG). Existe uma hierarquia na qual o ATM Forum submete as propostas dos vários corpos de padronização diretamente para um de seus *Sub Working Groups* (SWGs), onde estas propostas são colocadas em debate, podendo ou não sofrer alterações, sendo posteriormente colocadas em plenário para aprovação.

### 1.6.1 Atividades do ITU-T

As definições para o ATM foram finalizadas pelo CCITT SGXVIII. Algumas opções feitas pelo CCITT foram baseadas em argumentos puramente técnicos, ao passo que outras tiveram um compromisso entre os argumentos técnicos

e os interesses específicos de cada continente. As preferências dos continentes foram influenciadas principalmente pela ênfase dada a certos parâmetros. Por exemplo: alguns continentes enfatizavam a importância dos serviços de voz e vídeo, ao passo que outros valorizam os serviços de dados a altas velocidades.

Diferentes recomendações foram combinadas, e um grande número de parâmetros para o ATM foram definidos. Existem muitos problemas ainda não resolvidos e que deverão ser complementados por novas Recomendações.

O encontro SGXVIII de 1992 resultou, em junho de 1992 nas seguintes Recomendações, segundo Ginsburg (1996):

- **Recomendação I.113**, vocabulário para o B-ISDN: é um glossário dos termos e acrônimos usados no B-ISDN e ATM.
- **Recomendação I.121** CCITT, aspectos do B-ISDN: define os princípios básicos e as características do B-ISDN.
- **Recomendação I.150** CCITT, características funcionais do ATM B-ISDN: define as características funcionais do ATM.
- **Recomendação I.211** CCITT, aspectos gerais de serviços do B-ISDN: define a distribuição e a interação das classes de serviço, exemplos de aplicações e alguns atributos como taxa de bits, QoS e sincronização.
- **Recomendação I.311** CCITT, aspectos gerais da rede B-ISDN: detalha os conceitos que existem por trás do ATM, tais como as sub-camadas física e ATM, define as noções de canal virtual (VC) chaveados e caminho virtual (VP) de conexão cruzada, utilizando um certo número de exemplos. Cobre também o controle e gerenciamento do B-ISDN.
- **Recomendação I.327** CCITT, arquitetura funcional do B-ISDN: define a arquitetura básica do B-ISDN e os serviços não orientados a conexão.
- **Recomendação I.350** CCITT, aspectos gerais da qualidade de serviço, desempenho da rede em redes digitais, incluindo o ISDN: define os termos de qualidade de serviço QoS, a desempenho da

rede e os parâmetros específicos em termos de um número genérico de categorias.

- **Recomendação I.356** CCITT, camada ATM e o desempenho na transferência de células: define como os parâmetros de desempenho da camada ATM definidos na I.350 podem ser calculados teoricamente.
- **Recomendação I.361** CCITT, especificações da camada ATM: define o formato da célula ATM.
- **Recomendação I.362** CCITT, descrição funcional da camada de adaptação ATM (AAL): define os princípios básicos da subcamada, a classificação de serviços em termos da taxa de bits constante ou variável e se o serviço é orientado ou não orientado a conexão.
- **Recomendação I.363** CCITT, especificação da camada de adaptação ATM (AAL): define as especificações das AALs 1, 2, 3/4 e 5.
- **Recomendação I.365.1** CCITT, subcamada de convergência de serviços específicos de transporte *Frame Relay* (FR-SSCS): define as especificações para *interworking*<sup>10</sup> entre ATM e *Frame Relay*.
- **Recomendação I.364** CCITT, suporte a serviços de dados não orientados a conexão no B-ISDN: define suporte para serviços não orientados a conexão.
- **Recomendação I.371** CCITT, controle de tráfego e controle de congestionamento no B-ISDN: define as terminologias para os parâmetros de tráfego, controle e admissão da conexão e prioridades.
- **Recomendação I.413** CCITT, *interface* usuário rede: define a configuração e a terminologia utilizada no B-ISDN.
- **Recomendação I.432** CCITT, *interface* usuário rede: especificações da camada física: define os detalhes de como as células ATM são mapeadas no *frame* SONET ou SDH.

- **Recomendação I.555** CCITT, *Frame Relay*: define como o *Frame Relay* interage com um certo número de outros serviços, incluindo o B-ISDN.
- **Recomendação I.580** CCITT, *interworking* entre B-ISDN e 64 kb/s ISDN: define em linhas gerais como o N-ISDN pode ser *interworked* com o B-ISDN.
- **Recomendação I.610** CCITT, funções e princípios do OAM B-ISDN: cobre os princípios gerais e as funções necessárias para Operação, Administração e Manutenção (OAM).
- **Recomendação G.804**, mapeamento de células ATM em Plesiochronous Digital Hierarchy (PDH): define como as células ATM são mapeadas em várias estruturas TDM, tais como E1, DS1, E3 e DS3.

### 1.6.2 Atividades do ATM Forum

O ATM Forum foi formado em outubro de 1991, por quatro companhias: *Northern Telecom*, *Sprint*, *SUN Microsystems*, e *Digital Equipment Corporation* (DEC). Em janeiro de 1992, a sociedade foi aberta à indústria, sendo composta hoje por três categorias: principal, auditoria e usuário. Os membros principais participam dos encontros técnicos e de marketing. Os membros de auditoria recebem cópias dos documentos técnicos e de marketing, mas não participam dos encontros. Apenas os usuários participam do encontro *End User Roundtable* (ENR) McDysan; Spohn (1994).

Os trabalhos do ATM Forum são voltados para: operações, sinalização, interface Rede-Rede, controle de congestionamento, gerenciamento de tráfego, meios físicos adicionais, novas aplicações e camadas de adaptação. Algumas especificações do ATM Forum são descritas a seguir, conforme Ginsburg (1996):

- *Interface* ATM Usuário-Rede (UNI) especificação versão 2.0, define a capacidade de PVC para a UNI ATM, adiciona a camada física tecnologia baseada em FDDI e *Fiber Channel* para LAN,

---

<sup>10</sup> **Interworking**: define como diferentes redes podem interagir.

define uma *Interim Local Management Interface (ILMI)*, baseada no SNMP, e adota um subconjunto de funções OAM ATM.

- *Interface ATM Usuário-Rede (UNI)* especificação versão 3.0, substitui a versão 2.0, corrigindo erros e adicionando novas funções. A principal função acrescentada foi a definição de um protocolo de sinalização, definido como um subconjunto do padrão ITU-T, especificando as funções de controle de tráfego.
- *Interface ATM de troca de dados (DXI)* especificação versão 1.0; define uma *interface* baseada em *frame*.
- *Interface ATM Broadband-InterCarrier (B-ICI)* especificação versão 1.0, define características de serviço para conexão PVC entre as redes de telecomunicações
- *Interface ATM Rede-Rede (NNI)* especificação versão 1.0, define a interligação entre chaves.

### 1.6.3 Atividades do ANSI

O ANSI comitê T1 adaptou os padrões CCITT/ITU-T para atender as especificações da América do Norte. Os padrões e as datas de aprovação são listados a seguir, segundo Ginsburg (1996):

- **T1.624-1993**, BISDN UNI: taxas e especificações de formato - define o mapeamento das células ATM nos *payloads* DS3 e SONET e também como é realizado o gerenciamento de faltas.
- **T1.627-1993**, BISDN ATM funções e especificações: define a camada ATM, segundo a I.361, acrescentando explicações ao protocolo, interpretações do gerenciamento de tráfego, e alguns exemplos descrevendo os requisitos funcionais para uma implementação.
- **T1.629-1993**, BISDN ATM Camada de Adaptação 3/4, funcionalidade, parte comum e especificações: define as funcionalidades da AAL3/4 de acordo com a I.363.

- **T1.630-1993**, BISDN Camada de Adaptação para serviços a taxa de bits constante, funções e especificações: define a AAL1 com base na I.363 e define também as funções de emulação de circuitos DS1.
- **T1.635**, BISDN ATM Camada de Adaptação Tipo 5: define a AAL5 baseada na I.363.

#### 1.6.4 Atividades do IETF

Alguns RFCs, relativos ao ATM, serão listados a seguir, sendo que muitos outros estão em andamento Ginsburg (1996).

- **RFC 1483** *Multiprotocol Encapsulation Over ATM*: define como os protocolos das camadas superiores, como o IP, serão encapsulados e roteados sobre uma rede ATM.
- **RFC 1577** *Classical IP Over ATM*: define como o protocolo da internet (IP) pode utilizar as capacidades do ATM.

### 1.7. O protocolo B-ISDN

Em razão das limitações do modelo OSI para transmissão de voz, o ITU-T padronizou o conceito de (ISDN), com o objetivo de garantir serviços de telefonia. Com a expansão dos serviços digitais, como vídeo de alta definição, este conceito foi estendido, mantendo a mesma hierarquia lógica do modelo de camadas OSI, para as redes ATM B-ISDN, através da Recommendation I.321 CCITT (1991), com o propósito de integrar em um único meio, serviços de voz, dados e imagens.

#### 1.7.1 Conceitos Gerais

- Utiliza o conceito de chaves;
- Trabalha com células de tamanho fixo;
- É orientado a conexão;

- Toda conexão é feita por meio de caminho e canal virtual;
- Modo de transmissão é assíncrono.

A Figura 1.11 ilustra uma célula ATM para as interfaces (UNI e NNI).

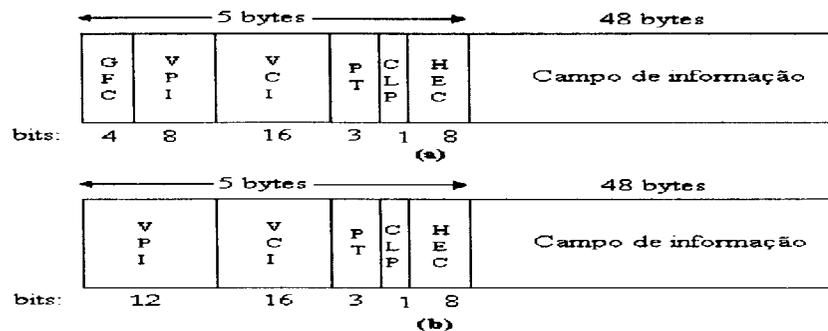


Figura 1.11: Células ATM (a) UNI (b) NNI

### 1.7.2 Componentes das Camadas

O ITU-T especificou um modelo de referência para o B-ISDN, o qual está ilustrado na Figura 1.12. O B-ISDN é representado por meio de camadas, que se baseiam no modelo OSI. Apesar do conceito de camadas estar presente também no B-ISDN, ele diverge em alguns pontos do modelo OSI, sobretudo com respeito ao número de camadas, e pelo fato de introduzir o conceito de planos associados às camadas, segundo Kim et al. (1994).

A camada física é dividida em duas sub-camadas: a camada inferior é a sub-camada de meio físico (PM), e a superior, é a sub-camada de convergência de transmissão (TC). Conforme McDysan; Spohn (1994), três corpos de norma definem a camada física em suporte para ATM: ANSI, CCITT/ITU-T e ATM Forum. As funções da sub-camada PM estão relacionadas com o meio de transmissão utilizado e incluem itens tais como bit de sincronismo e codificação da linha. A sub-camada TC contém funções relacionadas com o sistema de transmissão (ex.: Hierarquia Digital Síncrona - SDH, Rede Ótica Síncrona SONET, etc.) e a adaptação ao meio de transmissão para o transporte de células ATM pela camada ATM. A camada ATM se restringe à operação e manipulação das células ATM, e desde que a sub-camada TC

trate da adaptação entre a camada ATM e o meio, as funções da camada ATM são independentes do meio físico. As funções da camada ATM incluem geração e extração do *header* da célula, translação dos valores de VCI/VPI no *header* da célula, transferência das células das ligações de entrada para as ligações de saída, num nó de chaveamento dentro da rede, multiplexação e demultiplexação de células, e outras. Na terminologia OSI, a célula ATM corresponde a ATM-PDU, e o *header* da célula corresponde a ATM-PCI. A célula ATM é detalhada em McDysan; Spohn (1994) e Boudec (1992).

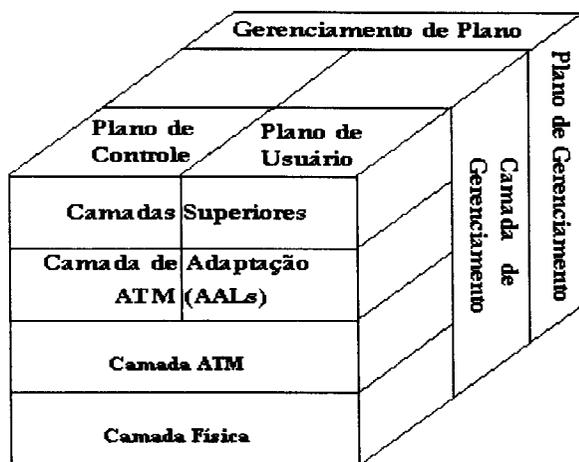


Figura 1.12: O B-ISDN.

O serviço que a camada ATM provê para as camadas superiores é a transferência de células ATM de tamanho fixo entre sistemas finais. Para adaptar este método de transferência para qualquer serviço requisitado, a camada de adaptação ATM (AAL) foi introduzida no topo da camada ATM. A AAL é dividida em duas sub-camadas McDysan; Spohn (1994), Sub-camada de Convergência (CS) e Sub-camada de Segmentação e Remontagem (SAR). A sub-camada (CS) é então subdividida em Serviço Específico (SS) e Parte Comum (CP). A sub-camada SSCS pode ser nula, não precisando ser implementada. A sub-camada CPCS sempre deve ser implementada junto com a sub-camada SAR. Estas camadas passam primitivas com referência a seus respectivos PDUs, resultando na passagem das primitivas SAR-PDU (que é o *payload* da célula ATM) para a camada ATM através do SAP ATM. O propósito da camada AAL é adaptar o serviço de transferência de

informação para a camada ATM para qualquer serviço, tal que a AAL é no entanto, dependente do tipo de serviço. O ITU-T especificou quatro diferentes classes AAL de acordo com três critérios diferentes, como descrito na tabela 1.1.

**Tabela 1. 1:** *As classes AAL*

<b>Classes AAL</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>Temporização</b>	<b>Requerida</b>		<b>Não Requerida</b>	
<b>Taxa de Bit</b>	<b>Constante</b>	<b>Variável</b>		
<b>Modo de Conexão</b>	<b>Orientado a Conexão</b>			<b>Não Orientado a Conexão</b>

As classes AAL são especificadas em Ginsburg (1996), Kim, Suda (1994), McDysan, Spohn (1994) e Boudec (1992) como segue:

- **Classe A:** Taxa de bit constante (CBR), serviço com temporização fim-a-fim e orientado a conexão;
- **Classe B:** Taxa de bit variável (VBR), serviço com temporização fim-a-fim e orientado a conexão;
- **Classe C:** Taxa de bit variável (VBR), serviço sem temporização fim-a-fim e orientado a conexão;
- **Classe D:** Taxa de bit variável (VBR), serviço sem temporização e não orientado a conexão.

Acima da camada AAL, o B-ISDN é verticalmente dividido em um plano de controle e um plano do usuário, onde o plano de controle está preocupado com o controle e gerenciamento da conexão dentro do plano do usuário. As entidades relacionadas com a sinalização no B-ISDN serão, portanto, inseridas no plano de controle. As funções do plano do usuário serão restritas apenas à transferência de informação do usuário.

O B-ISDN inclui um terceiro plano perpendicular ao plano de controle e ao plano de usuário, chamado gerenciamento de plano, o qual está dividido em camadas de gerenciamento e plano de gerenciamento. A camada de gerenciamento está horizontalmente dividida, sendo a responsável pelas funções de gerenciamento

dentro de cada camada, assim como o controle dos recursos dentro da camada e o tratamento do fluxo de informações de Operação, Administração e Manutenção (OAM) relacionadas com a camada. O plano de gerenciamento contém funções relacionadas com o sistema como um todo. A função de cada camada pode ser vista na tabela 1.2.

A AAL1 especifica como circuitos tipo TDM podem ser emulados sobre uma rede ATM, conforme ANSI T1.630. A AAL2 especifica o transporte ATM de circuitos de conexão orientada de áudio e vídeo de alta taxa de bits. O CCITT recomenda o uso da AAL 3/4 para transferência de dados sensíveis a perdas, mas não sensíveis a atraso, podendo também ser empregada para comunicação de dados orientado a conexão ou não. Em função do excessivo *overhead* da AAL 3/4, o CCITT não a recomenda para serviço de dados a altas velocidades. Por isso, o ATM Forum especificou a AAL5, com o objetivo de oferecer um serviço com um *overhead* reduzido e uma melhor detecção de erros.

**Tabela 1. 2:** As funções do B-ISDN

Camadas Superiores	Plano de Controle	Controle de Conexão no plano do usuário Sinalização
	Plano do Usuário	Operações sobre as informações do usuário
AAL	CS	Provisão de serviços da AAL para camadas superiores
	SAR	Segmentação e remontagem da CS-PDUs
ATM		Controle Genérico do Fluxo Geração e extração do cabeçalho da célula Chaveamento de células Multiplexação e demultiplexação de células
Camada Física	TC	Taxa de desacoplamento de células Geração/verificação do campo HEC Delineação de células Adaptação e transmissão do frame Transmissão geração/verificação do <i>frame</i>
	PM	Sincronização de bit Meio físico

As AAL tipos 3/4 e 5 possuem dois modos de operação: confiável e não confiável. Em operações confiáveis, as AAL-SDUs serão liberadas para o

destinatário sem possibilidades de erros. Isto será tratado na sub-camada SSCS pela retransmissão de erros CPCS-PDUs e impondo um controle de fluxo. Na operação não confiável, a AAL não retransmite erros AAL-SDUs; assim, qualquer retransmissão das AAL-SDUs será tratada pelas camadas superiores. A função da sub-camada SAR da AAL tipo 3/4, juntamente com a segmentação e remontagem do SAR-SDUs inclui a multiplexação acima de 1024 conexões CPCS sobre a mesma conexão AAL. A AAL tipo 5 não possui esta habilidade, mas utiliza os SAR-PDUs inteiros para transferir dados do usuário, conseguindo uma alta eficiência. Desde que não exista qualquer campo SAR-PCI no SAR-PDU para a AAL tipo 5, parte do *header* das células ATM (ATM-PCI) é usado para indicar se o campo *payload* da célula ATM contém a última parte do CPCS-PDU. Na AAL tipo 3/4, a delimitação dos CPCS-PDUs utiliza um campo dentro do SAR-PCI Staalhagen (1996).

## **1.8 Tecnologias alternativas**

O ATM concorre com muitas outras tecnologias, tanto nas LANs quanto nas WANs. A escolha sobre um padrão de rede a adotar recai sobre alguns fatores importantes como: custo, padrões estáveis, facilidade de gerenciamento, alcance, se a arquitetura é compartilhada ou não e a escalabilidade. Os itens a seguir descrevem algumas das opções de redes separadas em dois grupos, as tecnologias que são voltadas para LANs e aquelas voltadas para WANs

### **1.8.1 Na LAN**

Alternativas para o ATM em redes locais incluem: FDDI, 100 Mbps Ethernet, Ethernet e Token Ring chaveados e Fiber Channel. A figura 1.13 a seguir ilustra os requisitos de largura de banda para algumas das aplicações mais importantes que podem estar disponíveis na rede.

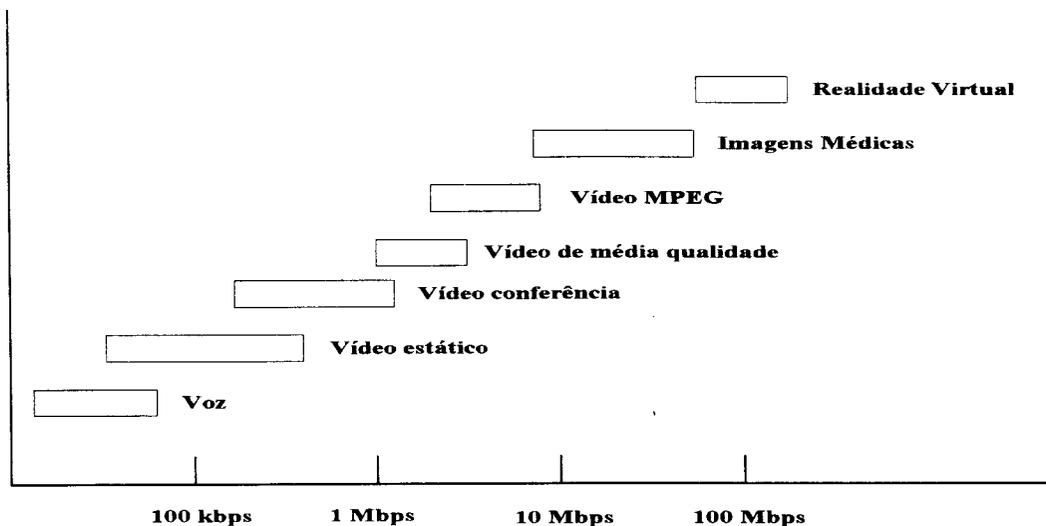


Figura 1.13: Requisitos de largura de banda por aplicação

### 1.8.1.1 Fiber Distributed Data Interface FDDI

Padronizado pelo ANSI em 1993, é considerada a tecnologia mais viável para *backbone*, devido à maturidade do *software* e do *hardware*, por possuir uma ampla base instalada, e capacidade nativa para *multicast/broadcast* Ginsburg (1996).

FDDI é uma tecnologia de rede de alta velocidade, que utiliza topologia em anel e cabeamento de fibra ótica com uma taxa de transmissão de dados da ordem de 100 Mbps. Utiliza a codificação 4B/5B (4 bits para cada 5 *baud*), o que lhe permite com um *clock* de 125 MHz, conseguir os 100 Mbps desejados, ao passo que se utilizasse codificação Manchester este *clock* deveria ser de 200 Mhz. Emprega topologia em anel com redundância (anel de *backup* para os casos de falha). O anel de *backup* permite que a rede continue funcionando mesmo em alguns casos de falhas de computador ou mesmo de conexão. Falhas subseqüentes podem ser corrigidas automaticamente pela reconfiguração do anel, mas podem provocar a criação de sub-redes desconectas. A extensão máxima do sistema em fibra é de 100 km, com as estações conectadas em ambos os anéis e separadas por uma distância máxima de 2 km e um máximo de 500 estações podem compartilhar o anel. O *frame* FDDI contém um comprimento máximo de 4500 bytes, o tamanho dos campos do

*frame* FDDI é medido em símbolos de 4 bits cada um, com um *header* contendo o endereço físico da estação origem e destino (o endereço FDDI pode conter de 4 a 12 símbolos) Prycker (1994).

O anel FDDI é percorrido por uma ficha (*token*) e quando uma estação quer transmitir, ela deve capturar a ficha e fazer sua transmissão. Cada estação ao longo do anel pega o pacote e verifica se é para ela; não sendo o pacote é retransmitido até chegar a seu destino. Após transmitir um pacote, a estação transmissora deve liberar a ficha, concedendo assim o direito de transmitir a outras estações.

O fato do FDDI trabalhar com compartilhamento do meio dificulta a implementação de aplicações multimídia, pois, além da exigência de uma boa largura de banda, exige-se também uma latência fixa, requisitos impossíveis de serem garantidos pelo FDDI. Em 1994, foi proposto uma extensão ao FDDI, que recebeu o nome de FDDI II, como uma tentativa de resolver os problemas do tráfego de multimídia no FDDI. Para isso, emprega o conceito de chaveamento de circuitos, o que garante uma latência baixa e constante, e a largura de banda de 100 Mbps é dividida em 16 circuitos de 6.144 Mbps, os quais podem ser sub-divididos em 96 sub-canais de 64 kbps, segundo Prycker (1994).

Apesar das inovações, o FDDI II, não teve muita aceitação e já se encontra em fase de padronização o FDDI III, com capacidade para operar a 1 Gbps, segundo Ginsburg (1996).

### **1.8.1.2 Ethernet e Token Ring chaveados**

As tecnologias Ethernet e Token Ring são consideradas inadequadas para redes de alta velocidade devido ao compartilhamento do meio.

Com as LANs chaveadas, este paradigma mudou completamente, podendo os usuários serem atendidos pela largura de banda completa oferecida pela tecnologia: 10 Mbps para o Ethernet e 16 Mbps para o Token Ring. Numa configuração full duplex, estas cifras podem ser dobradas.

De acordo com a figura 1.13, a disponibilidade de 10 ou 16 Mbps de largura de banda permite atender aos requisitos de muitos dos principais serviços

que a rede pode disponibilizar. Acrescenta-se a isso a possibilidade de se otimizar a largura de banda das LANs chaveadas através da utilização de protocolos como o PIM. Como em qualquer tecnologia orientada a pacotes, o grande desafio que esta por trás das LANs chaveadas é conseguir minimizar a latência e o controle do *jitter*, através do desenvolvimento de protocolos de reserva de recursos como o RSVP, segundo Ginsburg (1996).

### 1.8.1.3 Fast Ethernet

Uma extensão do padrão Ethernet 10 Base-T, o Fast Ethernet transporta dados a 100 Mbps. Com as regras definidas pelo IEEE 802.3u, o Fast Ethernet abandona a antiga tecnologia Ethernet, mas retém o protocolo CSMA/CD do Ethernet de 10 Mbps. Três tipos de Fast Ethernet estão disponíveis: 100 Base-TX que roda sobre UTP categoria 5; 100 Base-T4 com quatro pares trançados categoria 3; e 100 Base-FX, que opera sobre cabos de fibra ótica multi modo. Como não opera com cabo coaxial, a topologia em barramento tradicional nas redes Ethernet torna-se inviável. O formato do pacote permanece idêntico ao do antigo Ethernet, o que garante a utilização das antigas aplicações sem modificações. Como muitas das características do Ethernet são mantidas, muitos dos problemas também são mantidos, como: dificuldade na transmissão multimídia, eficiência baixa em carga alta, etc. A grande vantagem desta tecnologia está na facilidade de aumentar a velocidade de transmissão de 10 Mbps para 100 Mbps, com poucas modificações na topologia, no cabeamento e nos aplicativos que se utilizam da rede.

Assim como na Ethernet, muitos destes problemas podem ser resolvidos através da utilização de chaves. Utilizando-se apenas uma interface por porta, consegue-se eliminar completamente o problema da colisão, reduzir consideravelmente a latência e o *jitter* e utilizando-se configuração full duplex, consegue-se dobrar a largura de banda, segundo Ginsburg (1996).

#### 1.8.1.4 Fiber Channel

Fiber Channel é uma arquitetura de rede de alta velocidade. Em contraste com o ATM, Fiber Channel, é uma arquitetura voltada apenas para dados, o que não exclui a possibilidade do transporte de voz e vídeo especialmente combinando a reserva de recursos e mecanismo de priorização das camadas superiores. Os padrões elaborados pelo comitê ANSI's X3T11, definem taxas de dados de 133, 266 e 531 Mbps, e 1.06, 2.134 e 4.25 Gbps.

O comprimento máximo do pacote é de 2148 bytes, sendo que o *overhead* corresponde a 36 bytes.

Classes de serviços suportadas: **classe 1** adequada para serviços orientados a conexão que requeiram um alto *throughput* e uma longa duração; **classe 2**, que provê compartilhamento da largura de banda sendo adequada para protocolos de rede como o TCP ou SPX e **classe 3** para serviços não confiáveis, sendo ideal para vídeo conferência. Prevê suporte para *multicast*, transporte de redes como Ethernet e AAL5, segundo Ginsburg(1996).

#### 1.8.1.5 Outras

Outras tecnologias se enquadram nesta categoria como: *Integrated Services LAN* (ISLAN); *Gigabit Ethernet*, *100 VG AnyLAN*. Para leitura complementar consulte a bibliografia.

#### 1.8.2 Na WAN

Nas LANs, existem muitas tecnologias e serviços alternativos para o ATM. Nas WANs destacam-se: ISDN; *Frame Relay* e *Distributed Queue Dual Bus* (DQDB).

Fatores decisivos na escolha de uma tecnologia para WAN são: requisitos de largura de banda, maturidade dos padrões, escalabilidade e o mais importante, a disponibilidade de serviços e se eles podem ser tarifados adequadamente.

### 1.8.2.1 ISDN

*Narrowband* ISDN (N-ISDN) ou 64 kbps ISDN, precursor do ATM, desenvolvido para telefonia digital, tendo seus padrões publicados em 1984. A intenção original do ISDN foi prover a conversão dos antigos meios de telecomunicações para uma arquitetura totalmente digital, provendo serviços de voz, dados e vídeos em um mesmo meio.

Dois padrões foram definidos para interface física no ISDN: *Basic Rate Interface* (BRI), definido no CCITT ISDN I.430 e *Primary Rate Interface* (PRI), definida no CCITT ISDN I.431. A arquitetura ISDN foi a primeira a introduzir o conceito de planos associado a camadas, sendo precursor direto da arquitetura do B-ISDN/ATM. A configuração de serviços são definidas como segue:

- *Basic Rate Interface* (BRI): provê dois canais de 64 kbps chamado de canal (B), utilizado para dados, e um canal de 16 kbps chamado canal (D), utilizado para controle e gerenciamento. Normalmente refere-se a esta interface como 2B+D;
- *Primary Rate Interface* (PRI): provê 23 canais de 64 kbps canais (B) e um canal de 64 kbps chamado canal (D) para sinalização, sendo referido como 23B+D.

A ISDN PRI provê um único canal DS1 de 1.544 Mbps ou um canal de 2.048 Mbps E-1 numa conexão *full duplex*, utilizando multiplexação por divisão do tempo (TDM), segundo McDysan, Spohn (1994).

Em muitas instalações, um ou mais canais B, podem estar dedicados para voz, fax ou vídeo conferência. A tecnologia se mostra adequada para *sites* que requeiram uma largura de banda inferior a 2 Mbps. É importante frisar que as empresas de telecomunicações prevêm o *interworking* entre o ATM e o N-ISDN, podendo também ser utilizado para transmitir X.25 e Frame Relay, podendo ser tarifado em qualquer continente, segundo The Motorola Codex Basics Book Series (1992) e Ginsburg (1996).

### 1.8.2.2 Frame Relay

Assim como o N-ISDN, o Frame Relay não compete com o ATM em transmissão de taxas de dados mais altas (acima de 2 Mbps). Derivado do ISDN, os padrões que definem a operação do Frame Relay são: CCITT Recomendação Q.921, Q.922 e Q.933, segundo *The Motorola Codex Basics Book Series* (1992), McDysan e Spohn (1994).

O Frame Relay é uma evolução do X.25, e apresenta as seguintes características: acrescenta uma maior confiabilidade aos meios físicos, não garante a integridade dos dados, consegue uma maior velocidade no protocolo, adotando uma filosofia fim-a-fim e não armazena-e-envia como o X.25, o que reduz o atraso na rede em até 5 vezes e elimina a necessidade da utilização de buffers intermediários.

Quando a demanda por uma largura de banda não exceder a 2 Mbps, o Frame Relay pode ser uma boa opção pelas facilidades de tarifação dos serviços e a sua eficiência na utilização da largura de banda do canal, o que possibilita a redução de custos dos serviços prestados. Outro ponto a favor é a facilidade de *interworking* com o ATM, segundo Ginsburg (1996).

### 1.8.2.3 Distributed Queue Dual Bus (DQDB)

O DQDB foi padronizado pelo IEEE 802.6 para uso em altas velocidades nas redes de área metropolitana (MANs), tendo se firmado como uma tecnologia adequada para suporte aos serviços *Switched Multimegabit Data Service* (SMDS), segundo, Casale et al. (1994), Prycker (1994) e Ginsburg (1996). Uma das grandes vantagens do DQDB é sua habilidade em operar sobre várias plataformas desde E1/DS1 até E4 e SDH, o que permite uma variação na largura de banda de 34 a 155 Mbps.

O DQDB trabalha com unidades de tamanho fixo conhecidas como *slots* (equivalentes as células do ATM), embora a tecnologia seja não orientada a conexão, o *header* contém um VCI utilizado para identificar segmentos de dados do mesmo frame. Todo *header* é composto por quatro bytes, contendo campos de

sincronização e manutenção, o que diverge do ATM onde as células possuem um *header* de 5 bytes, segundo Casale et al. (1994).

## Capítulo 2

*Apresenta a rede ATM, as interfaces definidas para redes públicas e privadas, a célula ATM, caminhos e canais virtuais, os requisitos de roteamento e a formação de endereços ATM. Trata também das classes QoS, e das especificações do contrato de tráfego, além de apresentar alguns algoritmos de policiamento.*

### ATM Conceitos

#### 2.1 A rede ATM

Uma rede ATM é constituída por uma série de elementos de redes interligados entre si, com a particularidade de que, entre dois pontos finais quaisquer, a conexão se dará por intermédio de pelo menos uma chave ATM, sendo a chave o coração de uma rede ATM. Chen, Liu (1994)

Uma rede ATM pode ser pública ou privada. A rede pública, se caracteriza por fazer uso de circuitos compartilhados, e sua administração é de competência das empresas de telecomunicações. As redes privadas são caracterizadas pela utilização de circuitos dedicados, mesmo quando a posse e manutenção da linha forem de competência de uma operadora pública de telecomunicações.

Um terminal acessa a rede, por meio de uma *User Network Interface* (UNI). São três as UNIs disponíveis: *Private UNI*, para conexão em redes privadas; *Public UNI*, para conexão em redes públicas e *Residential UNI*, que foi definida e padronizada para interconectar sistemas finais residenciais às redes públicas. A interface *Network Network Interface* (NNI) é a interface que define a comunicação entre chaves. São três as NNIs disponíveis para interconexão entre chaves: *Private*

*NNI* (PNNI), que interconecta chaves em redes privadas; *Public NNI*, que interconectam chaves públicas, na realização de serviços simples e *Broadband Inter Carrier Interface* (B-ICI), utilizada na interconexão de troncos, em redes ATM públicas. A figura 2.1, exibe uma arquitetura típica de uma rede ATM, segundo Ginsburg (1996).

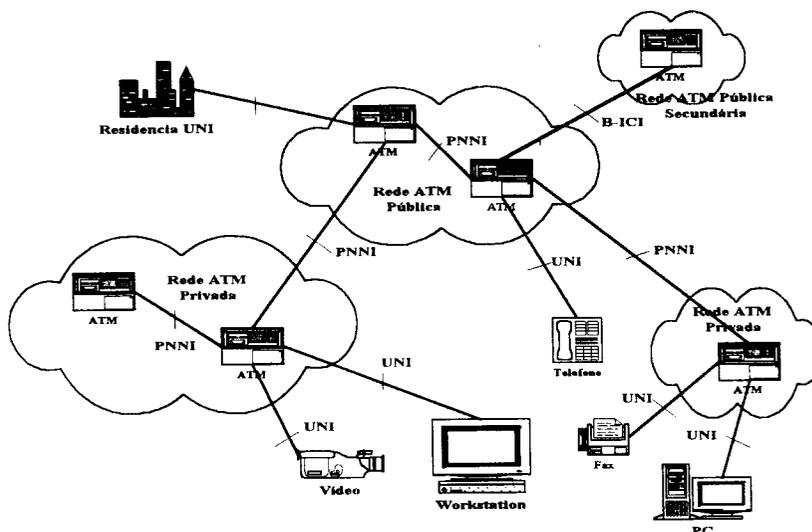


Figura 2. 1: Interfaces ATM privadas e públicas

## 2.2 A Célula ATM

Os padrões ATM, definem uma célula de tamanho fixo, com um comprimento de 53 bytes, sendo 5 bytes para *header* e 48 bytes para *payload*, conforme a figura 2.2. Os bits na célula, são transmitidos da esquerda para a direita, em uma estrutura contínua. As células são mapeadas em um meio físico de transmissão, podendo ser: nos EUA: DS1, DS3 ou SONET; na Europa: E1, E3 e E4, ou algum outro padrão *Synchronous Transfer Mode* (STM) do ITU-T. Dentre as funções do *header*, estão: identificação do destino, o tipo de célula e a prioridade da célula. O *Virtual Path Identifier* (VPI), e o *Virtual Channel Identifier* (VCI), detêm apenas significado local, e identificam o destino. O campo *Generic Flow Control*

(GFC), permite a um multiplexador controlar a disputa por recursos em um tronco compartilhado, utilizando técnicas de controle seletivo do tráfego. O *Payload Type* (PT), indica qual o tipo de informação contida no *payload*, a qual pode ser: dados do usuário, sinalização, ou informação de manutenção. O bit (CLP), indica a prioridade da célula. Células com baixa prioridade, são descartadas antes das células de prioridade maior, na ocorrência de um congestionamento.

O *Header Error Control* (HEC), utiliza CRC<sup>11</sup>, para detectar erros no *header* da célula, provendo correção de erros para um único bit, e detecção de erros para múltiplos bits. O campo *payload* é transmitido sem detecção e correção de erros. A rede ATM, possui nas camadas mais altas, mecanismos para a verificação e correção de erros no *payload*. O tamanho fixo das células simplifica a implementação de chaves e multiplexadores ATM, e habilita a implementação de altas velocidades Chen; Liu (1995).

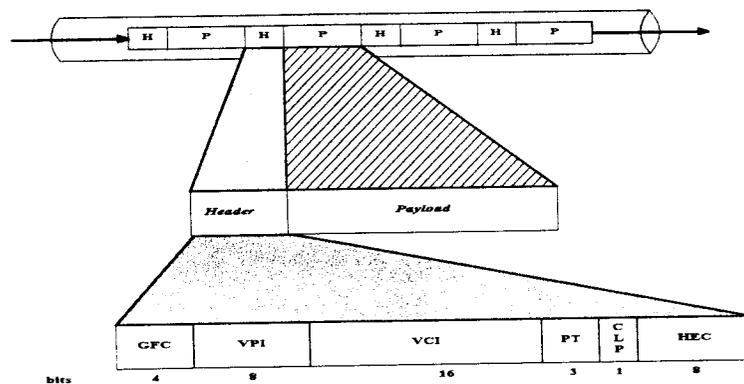


Figura 2. 2: Formato de uma célula ATM

### 2.2.1 Segmentação de células ATM

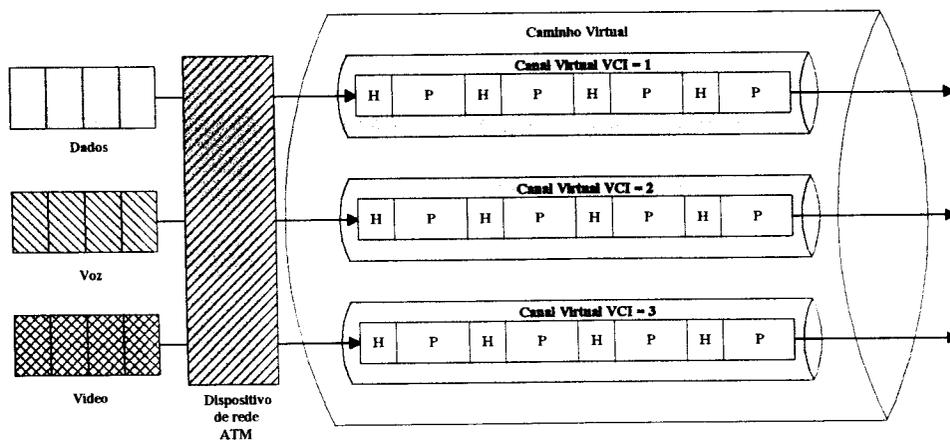
Em um sistema ATM, toda informação do usuário, seja ela voz, vídeo ou dados, deve ser particionada em células de tamanho fixo, multiplexadas em uma

<sup>11</sup> **CRC**: Cyclic Redundancy Check, algoritmo de verificação de erros por redundância cíclica.

fila de bits e transmitidas através de um meio físico. Em uma rede ATM, é possível a um usuário fazer uma chamada multimídia, transmitir dados e manter um diálogo, simultaneamente e em tempo real.

Vídeo e voz são aplicações muito sensíveis ao tempo; a informação não pode atrasar mais que uma fração de segundo, e o atraso não pode sofrer variações significativas. Os dados podem ser enviados orientados a conexão ou não orientados a conexão. De qualquer forma, os dados, embora não tão sensíveis ao atraso quanto as aplicações de voz e vídeo, são muito sensíveis às perdas. Portanto, o ATM deve discernir entre o tráfego de voz, vídeo e dados, dando ao tráfego de voz e vídeo, prioridade e garantia de um atraso mínimo, e simultaneamente assegurando uma perda muito baixa no tráfego de dados.

O exposto acima pode ser implementado, mediante o estabelecimento de um caminho virtual VP entre dois usuários finais e sobre este, três canais virtuais VC. A figura 2.3 mostra como estes três tipos de tráfegos podem ser combinados sobre um único caminho virtual VP.



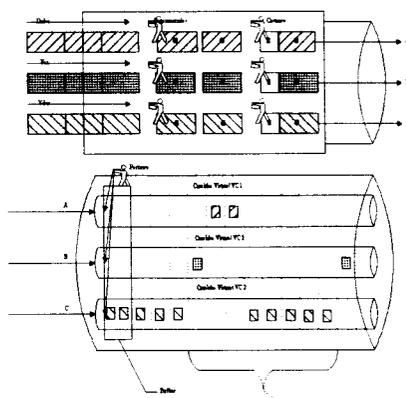
**Figura 2. 3:** *Canais virtuais dando suporte a múltiplas aplicações*

## 2.2.2 Um exemplo de como as células podem ser organizadas

Analisando um pouco mais de perto o item anterior, é possível visualizar como o tráfego e as células podem ser organizados em uma rede ATM.

O emissor, um PC ou uma *Workstation* que possua uma interface ATM, particiona os dados (por um “segmentador”), em pacotes de 48 bytes (o *payload* da célula). No próximo passo, um “carteiro” acrescenta o endereço (o *header* da célula) ao *payload*, preenchendo os campos VPI e VCI e todos os campos restantes até compor os 5 bytes do *header*. Neste momento, as células ATM (53 bytes) proveniente das fontes de voz, dados e vídeo estão prontas, e passam a disputar um *slot* de tempo na interface conectada ao PC ou à *Workstation*. As fontes de voz, vídeo e dados são associadas a um VCC<sup>12</sup>, sendo que os dados são transmitidos no VCI = 1, a voz no VCI = 2 e o vídeo no VCI = 3 (por exemplo).

A figura 2.4, mostra como se comporta um terminal ATM. Num próximo passo, um “porteiro”, se encarrega de particionar o tempo de transmissão de dados em intervalos de 8 células (por exemplo) (em torno de 80μs à taxa de DS3), escalonando duas células de texto uma de voz e cinco de vídeo. Todas as fontes (voz, vídeo e dados) competem por uma largura de banda a cada intervalo de 8 células, com o “porteiro” garantido sempre a mesma proporção. Células podem ser retidas no buffer pelo “porteiro”, quando todos os *slots* de tempo do intervalo já estiverem preenchidos.



**Figura 2. 4:** Exemplo do Modo de Transferência Assíncrono

<sup>12</sup> VCC: Conexão de Canal Virtual.

### 2.3 Caminho Virtual e Canal Virtual

Os campos VPI e VCI, no *header* das células, possuem significado apenas local, e identificam o destino da célula ATM conforme Kim et al. (1994). O número de bits alocados no *header* da célula ATM, limitam uma interface física UNI a suportar no máximo  $2^8 = 256$  caminhos virtuais e a interface física NNI, a suportar no máximo  $2^{12} = 4.096$  caminhos virtuais. Cada caminho virtual suporta, no máximo,  $2^{16} = 65.536$  canais virtuais, em ambas as interfaces.

Um canal virtual (ou *Virtual Channel VC*), é um termo genérico para uma conexão unidirecional, para o transporte de células ATM. Uma conexão de caminho virtual (VCC), é uma concatenação de consecutivos *links*<sup>13</sup> VC. Uma porção contínua de VCC é chamada de um segmento VC. Os atributos de um VCC são negociados entre o usuário e a rede, assim como a Qualidade de Serviço QoS e o *throughput*<sup>14</sup>. Estes atributos permanecem constantes durante a conexão, a não ser que sejam renegociados.

O caminho de transmissão (o caminho virtual e o canal virtual), pode por analogia, ser comparado a um sistema rodoviário, onde as células representam os veículos, o meio de transmissão as rodas, o caminho virtual, um conjunto de direções possíveis, e o canal virtual, a rota (rodovia) definida pelo caminho virtual. A figura 2.5 ilustra esta analogia considerando três cidades dos EUA: Dallas, Fort Worth e Houston, McDysan, Spohn (1994)

---

<sup>13</sup> **Link:** caminho de comunicação ou canal entre dois componentes ou dispositivos.

<sup>14</sup> **Throughput:** medida de como a informação útil total é processada em um período de tempo definido.

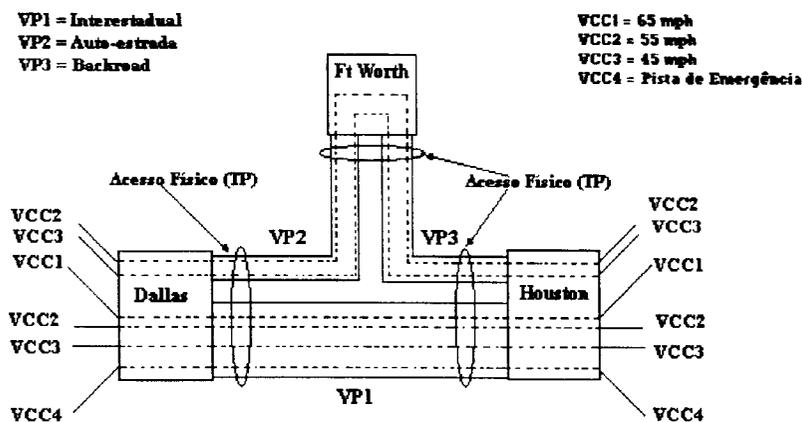


Figura 2. 5: Exemplo de como as células são transportadas no ATM

A presença de rodovias interestaduais, auto estradas e *backroad*, entre 2 cidades, permite a escolha de diferentes rotas para se chegar ao mesmo destino.

Assim, um carro (célula), pode viajar de Dallas para Houston, através da auto estrada para Fort Worth, e em seguida pegar a *backroad* para Houston, ou poderia ir direto para Houston utilizando a interestadual.

Se a escolha for pela interestadual (VP1), existe ainda a possibilidade de optar por um dos três VCCs: VCC1 – 65 mph, VCC2 55 mph e VCC3 45 mph. A escolha de diferentes VCCs, implica em atrasos diferentes, no percurso origem destino. O exemplo possibilita também que os veículos (células), dentro das várias possibilidades de escolha de rota, possam perder a seqüência. Para evitar a perda de seqüência, deve-se impor alguma disciplina, para que todos sigam exatamente pela mesma rota, obedecendo uma certa ordem e respeitando a velocidade limite para que a analogia possa ser aplicada.

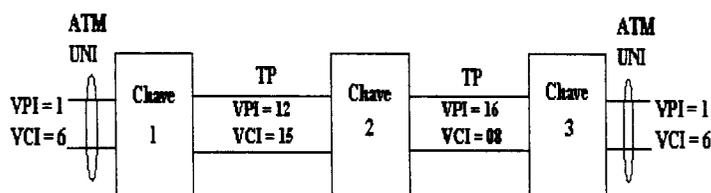
As rotas possuem também diferentes qualidades. Quando se utiliza um mapa da Associação Americana de Automobilismo (AAA), pode-se selecionar a rota com base em muitos critérios: mais rápida; melhor cenário; menor custo, e permite saber qual o tempo que será gasto no percurso. Estes mesmos princípios se aplicam ao ATM.

Na camada ATM, os usuários estabelecem uma escolha entre uma conexão de caminho virtual VPC, ou uma conexão de canal virtual VCC. Uma conexão VPC, é chaveada através do seu identificador de caminho virtual (VPI). O

usuário de um VPC pode associar a ele vários VCCs, que serão transportados transparentemente pela rede desde que sigam a mesma rota.

As conexões VCCs são chaveadas através da combinação dos campos VPI e VCI. Os valores de VPI e VCI são únicos sobre um caminho de transmissão ou *Transmission Path* (TP) específico, e possuem apenas significado local. Assim cada TP entre dois dispositivos de rede (como chaves ATM), utilizam VPIs e VCIs independentes, como mostra a figura 2.6. Cada chave mapeia uma entrada VPI e VCI para uma saída VPI e VCI.

A chave 1 associa o VCI 06 à saída VCI 15, e a entrada VPI 01 à saída VPI 12. Na chave 2, o VPI 12 é associado ao VPI 16 e o VCI 15 ao VCI 08. Por fim na chave 3, o VPI 16 é associado ao VPI 01 e o VCI 08 ao VCI 06. Ao longo de uma mesma rede ATM, é perfeitamente possível encontrar valores de VPIs e VCIs repetidos em *links* diferentes.



**Figura 2. 6:** Exemplo típico de translação dos campos VPI/VCI, em uma ligação fim-a-fim

### 2.3.1 Considerações a respeito do VPI

A figura 2.7, mostra uma conexão ATM entre as *Workstation P* e *Q*. Uma conexão ATM inicia com um protocolo de sinalização negociando a criação do caminho e canal virtual, necessários para o estabelecimento da conexão entre os dois pontos finais. Mas como criar as condições necessárias para estabelecer o caminho e canal virtual em uma rede ATM, é o que será discutido nos próximos tópicos. Todas

as considerações nestes tópicos se referem a *FORERUNNER ATM Switch Configuration Manual* (1996).

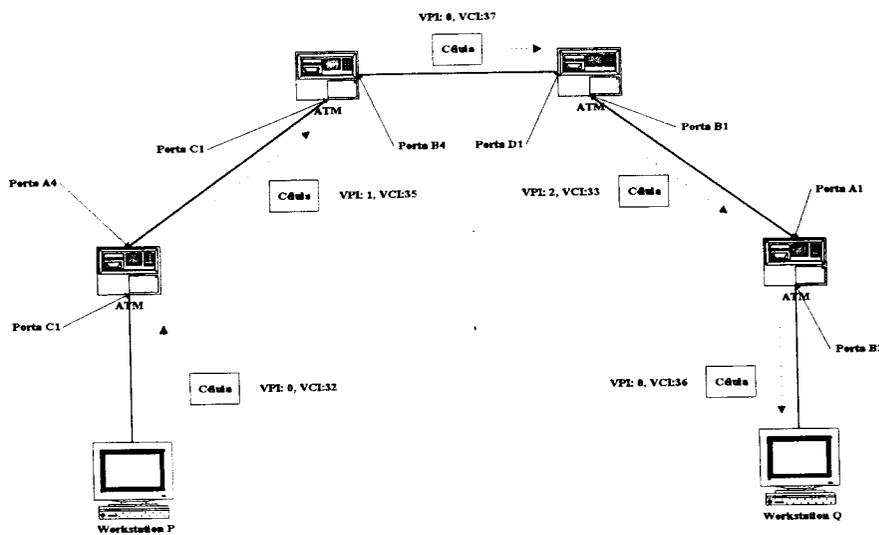


Figura 2. 7: Exemplo de uma típica conexão em redes ATM

Um único caminho virtual, pode ser usado para rotear muitos canais virtuais em uma rede ATM. O valor de VCI, não é alterado quando a célula é roteada através da rede, via caminho virtual.

Quando uma célula é recebida em uma porta de entrada de uma chave, seu VPI é examinado para determinar a porta de saída e seu novo VPI. O campo VCI do *header* ATM, permanece inalterado, tal que todos os canais dentro de um caminho, serão chaveados corretamente, sem alterar o valor do VCI das células deste canal.

Quatro parâmetros são necessários para definir um caminho em uma chave ATM: porta de entrada (*iport*), VPI de entrada (*ivpi*), porta de saída (*oport*) e VPI de saída (*ovpi*). Podem ser representados como segue:

**<iport> <ivpi> <oport> <ovpi>**

O valor de VCI permanece inalterado quando as células são chaveadas através do caminho virtual. Por exemplo, um caminho definido como A4|10 → B4|20, irá mapear as células recebidas na porta A4 com VPI: 10 para a porta B4 com VPI: 20, e o mesmo VCI. Outra consideração importante é que a informação irá

percorrer o canal em uma única direção, sendo o canal unidirecional. Observe a figura 2.8.

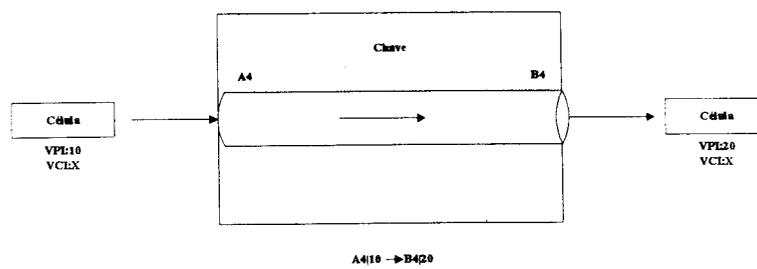


Figura 2. 8: Células roteadas por seus VP. .

### 2.3.2 Considerações a respeito do VCI

Uma combinação de canal e caminho virtual especifica uma conexão. Em uma chave, cada canal virtual chaveia as células com um específico VPI e VCI, recebidos em uma porta específica, para outra porta com novos valores de VPI e VCI. Ao contrário dos caminhos virtuais, os canais virtuais descrevem uma única conexão entre dois pontos finais, como por exemplo duas chaves.

Seis parâmetros são necessários para definir um canal virtual: porta de entrada (*iport*), VPI de entrada (*ivpi*), VCI de entrada (*ivci*), porta de saída (*oport*), VPI de saída (*ovpi*) e VCI de saída (*ovci*). São representados pela notação:

<iport> <ivpi> <ivci> <oport> <ovpi> <ovci>

A figura 2.9 ilustra um exemplo de canal virtual

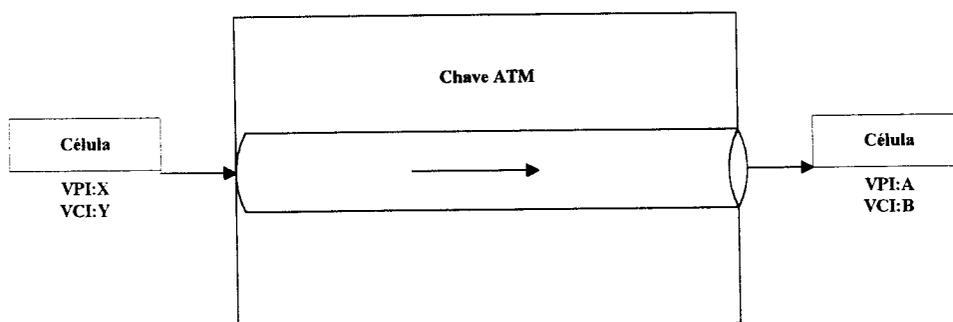
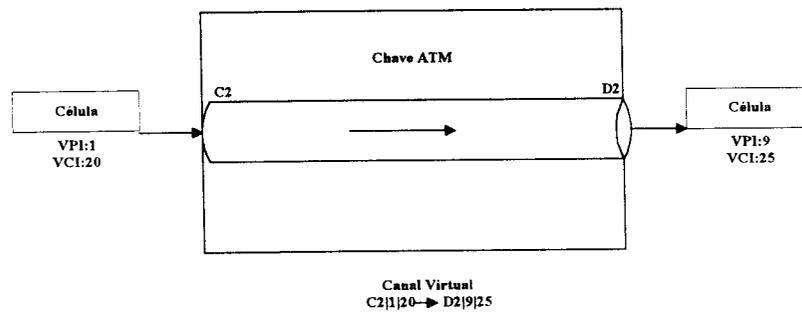


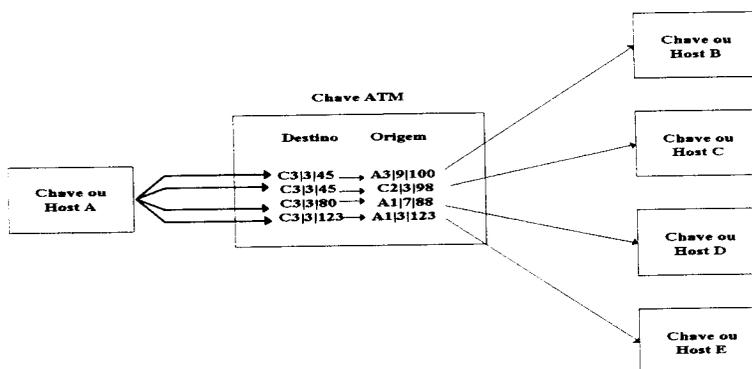
Figura 2. 9: Exemplo de roteamento de células empregando os campos VCI e VP

Ambos os valores de VPI e VCI podem ser alterados quando uma célula é chaveada por canal virtual. A figura 2.10, ilustra um exemplo onde o canal virtual  $C2|1|20 \rightarrow D2|9|25$ , irá chavear as células recebidas na porta C2 com VPI:1 e VCI: 20, para a porta de saída D2 com VPI:9 e VCI:25. Assim como o caminho virtual, para estabelecer uma conexão bidirecional entre duas portas de uma chave, é necessário negociar o estabelecimento de dois canais virtuais (um para transmitir outro para receber).



**Figura 2. 10:** Exemplo de roteamento por canal virtual.

Antes que um canal virtual possa ser criado, um caminho correspondente a origem e destino deve existir. Por exemplo, antes que o canal mostrado na figura 2.11 possa ser criado, o caminho  $C3|3$  deve existir.



**Figura 2. 11:** Canais virtuais criados sobre o caminho destino  $C3|3$ .

Similarmente, antes que o canal virtual mostrado na figura 2.12 possa ser criado, é necessário que exista o caminho origem C2|2.

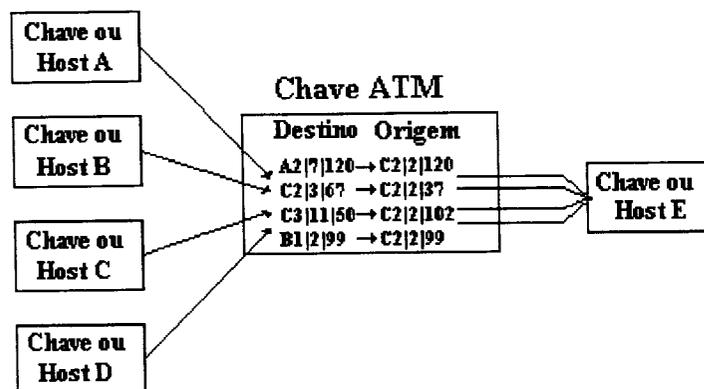


Figura 2. 12: Canal virtual criado sobre o caminho origem C2|2

## 2.4 Endereçamento e roteamento

Em uma rede chaveada, endereçamento e roteamento são operações críticas. Como em qualquer outro protocolo, um endereço é um identificador que serve para designar univocamente, um elemento constituinte da rede. Como a tecnologia ATM é orientada a conexão, endereços são utilizados apenas na fase de *setup*. Uma vez estabelecida a conexão, os identificadores da conexão passam a ser utilizados, não sendo mais necessário que o endereço destino continue a trafegar pela rede.

O protocolo de sinalização automaticamente associa os valores de VPI/VCI, para o endereço ATM e a interface física UNI ATM, de acordo com o tipo de conexão, que pode ser: ponto-a-ponto ou ponto-para-multiponto. Uma porta física UNI ATM deve ter pelo menos um endereço ATM. Uma porta ATM UNI, pode ter mais de um endereço ATM, segundo Ginsburg (1996).

Como uma VCC ou VPC é definida em uma única direção (*simplex*<sup>15</sup>), uma conexão ponto-a-ponto é constituída por um par de VCCs ou PVCs *simplex*, ou seja, um canal ou caminho para transmissão, outro para recepção. Os parâmetros para o VCC ou VPC na transmissão e na recepção podem ser completamente diferentes, assim como, os valores de VPI ou VCI associados em cada ponto da conexão, podem diferir dependendo do sentido do fluxo de informação, segundo Ginsburg (1996).

#### 2.4.1 Atributos desejáveis pela camada ATM

Um conjunto de atributos desejáveis devem ser seguidos, quando forem designar um esquema de endereçamento para a camada ATM. Cada endereço deve ser único e possuir ao menos os seguintes atributos conforme McDysan, Spohn (1994):

- Simplicidade;
- Uso eficiente do espaço de endereçamento;
- Facilidade de administrar mudanças de endereços;
- Extensibilidade do sistema de endereçamento;
- Atribuição automática.

#### 2.4.2 Requisitos para roteamento

Células de um mesmo VPC ou VCC devem seguir pela mesma rota, definida como uma seqüência ordenada da porta física da chave que as células deverão atravessar da origem para o destino. Uma rota é estabelecida segundo McDysan, Spohn (1994) em resposta aos seguintes eventos:

- Um PVC foi recentemente estabelecido;
- Um PVC falhou e está sendo restabelecido automaticamente.

Em uma rede ATM, uma rota é liberada de acordo com McDysan, Spohn (1994) em resposta a pelo menos um dos seguintes eventos:

---

<sup>15</sup> **Simplex:** transmissão de dados em apenas uma direção.

- Um PVC foi desconectado;
- Uma falha foi detectada em um PVC restaurado;
- Uma chamada foi desconectada em resposta a uma falha.

Alguns atributos são desejáveis para o roteamento, segundo McDysan, Spohn (1994), e incluem ao menos o seguinte:

- Simplicidade;
- Determinação da rota de menor custo;
- Facilidade de administrar mudanças na rede em termos de novos *links* e nós;
- Extensibilidade para grandes redes.

## 2.5 Formatos dos Endereços ATM

A UNI 3.1 define os formatos para os endereços ATM em redes públicas e privadas.

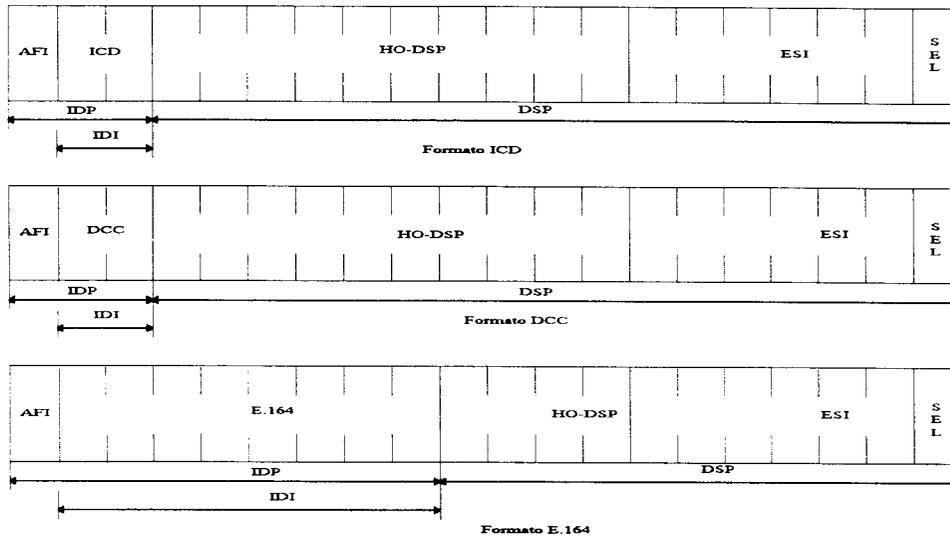
Análogo às redes de telefonia, os endereços para as redes públicas são oferecidos pela autoridade que fornece o serviço de rede, ao passo que, para as redes privadas, um endereço ATM possui 20 *octetos*, e seu formato é baseado no *Network Service Access Point* (NSAP), OSI. Muito embora um endereço ATM não seja um NSAP, ele usa a mesma estrutura, codificação binária, semântica e sintaxe de seu modelo. Em uma UNI pública, um endereço ATM, segundo Chao et al. (1994), pode ter os seguintes formatos:

- E.164, se for adotado o formato E.164 do ITU-T;
- Qualquer dos formatos NSAP (três), definidos para as redes privadas, se a rede pública adotar um deles;
- Ambos, se a rede pública adota o formato E.164 do ITU-T e interage com uma rede privada.

Apesar de existirem três formatos de endereço para redes privadas, a especificação UNI 3.1 estabelece que um terminal qualquer da rede é capaz de

realizar uma conexão com um outro usuário, independentemente dos formatos de endereço adotados pelos dois pontos.

A figura 2.13 apresenta os três formatos de endereços:



**Figura 2. 13:** Os três formatos de endereços ATM

Os campos dos três formatos são definidos a seguir:

O campo *Initial Domain Part* (IDP), está subdividido em dois outros campos. O campo *Authority and Format Identifier* (AFI) possui um byte e especifica qual o formato adotado (DCC, ICD ou E.164), e qual a autoridade administrativa que define o restante do endereço. O segundo campo, *Initial Domain Identifier* (IDI), assume diferentes significados dependendo do campo AFI:

- Caso AFI indique o formato *International Code Designator* (ICD), o campo IDI possui 2 bytes e o significado de um ICD. Este código é definido pelo Instituto Britânico de Normatização e designa uma organização internacional;

- Caso AFI indique o formato DCC, o campo IDI possui 2 bytes e especifica o país ao qual o endereço está associado. Os códigos são os definidos na norma ISO 3166;
- Caso AFI indique o formato E.164, o campo IDI possui 8 bytes e especifica um número de até 15 algarismos de uma RDSI. Estes números incluem números telefônicos e são definidos pela provedora de serviços RDSI.
- O campo IDP é dividido em dois outros campos: *Domain Specific Part* (DSP) de maior ordem (HO-DSP), e o de menor ordem, que é subdividido, em dois outros campos chamados de *End System Identifier* (ESI), e *Selector* (SEL);
- O campo ESI é um identificador que deve ser globalmente único (como um identificador MAC IEEE) ou, pelo menos, único para uma dada combinação IDP+HO-DSP, sendo o tamanho para este campo de 6 *octetos*;
- O campo SEL, possui apenas significado local. Assim, o campo SEL não é utilizado para roteamento ATM e seu tamanho é de 1 *octeto*.

### 2.5.1 IP sobre ATM

O RFC 1577 do IETF especifica a operação em modo nativo do protocolo IP Clássico sobre ATM, no ambiente de uma sub-rede lógica IP (*Logical IP SubNetwork* - LIS). São abordados dois aspectos essenciais desta operação: o encapsulamento dos pacotes e a resolução de endereços. No RFC 1577, as implementações do IP sobre ATM devem suportar o método de encapsulamento conhecido como IEEE 802.2 *Logical Link Control/SubNetwork Access Protocol* (LLC/SNAP).

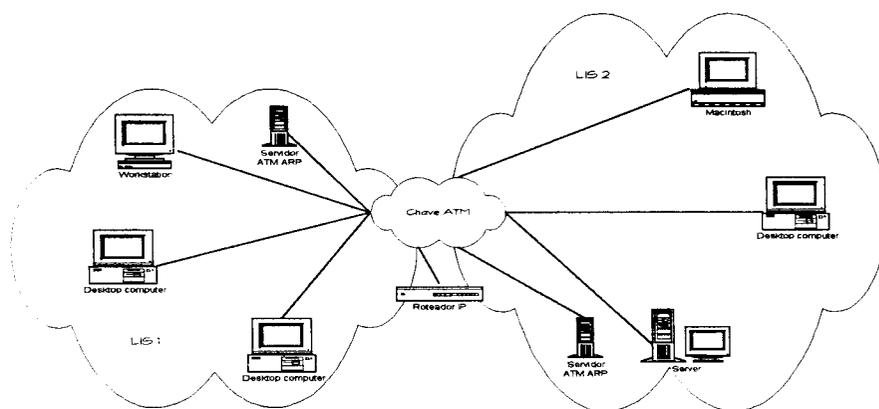
O RFC 1483, define métodos para o encapsulamento de pacotes de protocolos diversos nos pacotes definidos pelo protocolo AAL5, e foi aproveitado pelo RFC 1577.

A resolução de endereços pelo RFC 1577 define dois protocolos chamados: *ATM Address Resolution Protocol (ATMARP)* e *Inverse ATM Address Resolution Protocol (InATMARP)*, baseados, respectivamente, nos protocolos ARP e RARP da família de protocolos TCP/IP. A recomendação estabelece que todas as estações IP participantes devam implementar esses dois protocolos, segundo Chao et al. (1994).

Nas redes que suportam apenas PVCs, exige-se somente que todas as estações IP utilizem o protocolo InATMARP para compor suas tabelas internas de mapeamento de endereços.

Para as redes que suportam *Switched Virtual Connection (SVCs)*, a recomendação exige que cada LIS (sub-rede IP) contenha um servidor ATMARP, responsável por atender às solicitações do protocolo ATMARP, emitidas pelos seus membros, os clientes ATMARP.

Cada LIS deve possuir um único servidor ATMARP, embora um mesmo servidor possa atender inúmeras sub-redes LIS. A figura 2.14 ilustra como duas sub-redes LIS independentes, partilham a mesma rede ATM, e interagem através de um roteador IP.



**Figura 2. 14:** Representação típica de IP clássico sobre ATM

O modelo impõe algumas limitações, como a necessidade da utilização do roteador IP para a interconexão de múltiplas sub-redes LIS.

O servidor ATMARP, necessita manter tabelas que possam transladar endereços IP para endereços ATM. A tabela 2.1, lista um conjunto mínimo de mensagens, necessárias para a implementação de um protocolo simples de *consulta/resposta*, conforme Chao et al. (1994), necessário para a interação entre um *host* e o servidor de ATMARP.

**Tabela 2. 1:** Mensagens de sinalização entre *host* e servidor ATMARP

<b>Tipo</b>	<b>Campo 1</b>	<b>Campo 2</b>	<b>Campo 3</b>	<b>Campo 4</b>	<b>Comentário</b>
<b>Consulta ARP</b>	<b>Endereço IP fonte</b>	<b>Endereço ATM fonte</b>	<b>Endereço IP destino</b>		<i>Do host IP para o servidor, para obter o endereço ATM destino</i>
<b>Resposta ARP</b>	<b>Endereço IP fonte</b>	<b>Endereço ATM fonte</b>	<b>Endereço IP destino</b>	<b>Endereço ATM destino</b>	<i>Do servidor para o host IP com endereço ATM destino</i>
<b>Adiciona host</b>	<b>Endereço IP</b>	<b>Endereço ATM</b>			<i>Do host IP para o servidor, para atualizar o endereço ATM</i>
<b>Confirmação</b>	<b>Endereço IP</b>	<b>Endereço ATM</b>	<b>VCI Reservado</b>		<i>Do servidor para o host IP com VCI reservado</i>

Quando um *host* necessita enviar um pacote para um *host* com endereço IP, ele obtém o endereço ATM correspondente da tabela de endereços, e passa o pacote IP e o endereço ATM para a entidade de processamento, que realiza as funções de gerenciamento da conexão. Se a tabela de endereço não possuir uma entrada para o *host* IP destino, uma *consulta* ARP (figura 2.15) é enviada usando a seguinte mensagem:

<b>VCI usado</b>	<b>Payload</b>				
	<b>Tipo</b>	<b>Campo 1</b>	<b>Campo 2</b>	<b>Campo 3</b>	<b>Campo 4</b>
<b>VCI-U</b>	<b>Consulta ARP</b>	<b>IP<sub>s</sub></b>	<b>ATM<sub>s</sub></b>	<b>IP<sub>d</sub></b>	<b>?</b>

**Figura 2. 15:** Pacote de consulta ARP

O VCI-U é o identificador de canal virtual para o VC, globalmente conhecido pelo servidor ATMARP. Recebendo esta mensagem, o servidor ATMARP

retornará o endereço ATM destino para a fonte, usando a seguinte mensagem de resposta ARP FIGURA 2.16:

VCI usado	Payload				
	Tipo	Campo 1	Campo 2	Campo 3	Campo 4
VCI-ATM-S	Resposta ARP	IP <sub>s</sub>	ATM <sub>s</sub>	IP <sub>d</sub>	ATM <sub>d</sub>

Figura 2. 16: Pacote de resposta ARP

Os pacotes ATMARP e InATMARP, assim como ocorrem com os pacotes IP, são encapsulados em pacotes AAL5, utilizando o método LLC/SNAP.

São muitas as possibilidades de interconexão de ATM com redes heterogêneas. Porém aprofundar neste assunto, não é o objetivo deste trabalho.

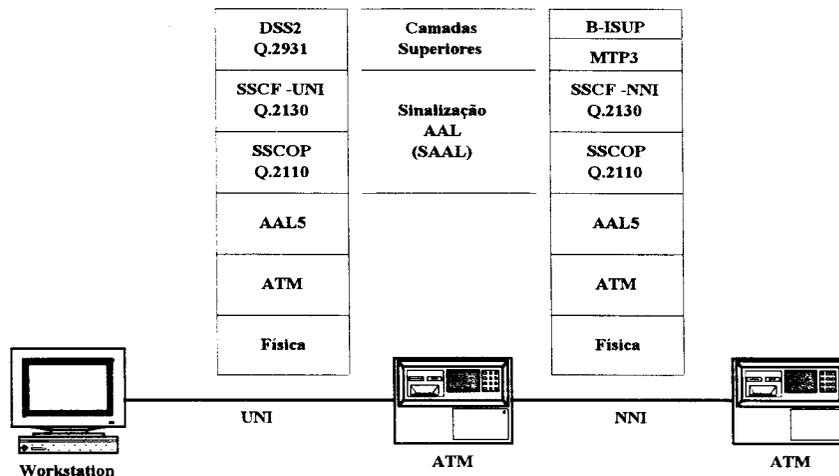
### 2.5.2. Sinalização

A negociação entre o usuário e a rede, com respeito à alocação de recursos (VCI/VPI, *throughput*, e QoS), é realizada por um canal virtual separado chamado de canal de sinalização, sendo a sinalização uma designação genérica para as funções que controlam dinamicamente as conexões ATM, devem ser implementadas por um protocolo de sinalização.

O chaveamento de VPC(s)/VCC(s), é controlado através da troca de mensagens de sinalização, o que acontece na fase de estabelecimento da conexão, ou fase de *setup*, que antecede a fase de transferência de informação, de acordo com Prycker (1994).

As funções de sinalização, são de competência do Plano de Controle do modelo de referência do B-ISDN. A arquitetura do protocolo de sinalização do Plano de Controle é mostrada na figura 2.17. O protocolo de sinalização Q.2931 foi derivado dos protocolos de sinalização UNI Q.931, específico para NISDN, e o Q.933 do *Frame Relay*. O nome formal do protocolo de sinalização UNI ATM é *Digital Subscriber Signaling System 2 (DSS2)*. A Recomendação Q.2130 do ITU-T, especifica a *Service Specific Coordination Function (SSCF)* para UNI. A Recomendação Q.2110 do ITU-T, especifica *Service Specific Connection Oriented*

*Protocol* (SSCOP). O protocolo B-ISUP, opera sobre o *Message Transfer Protocol 3* (MTP3), segundo Ginsburg (1996).



**Figura 2. 17:** *Arquitetura do plano de controle*

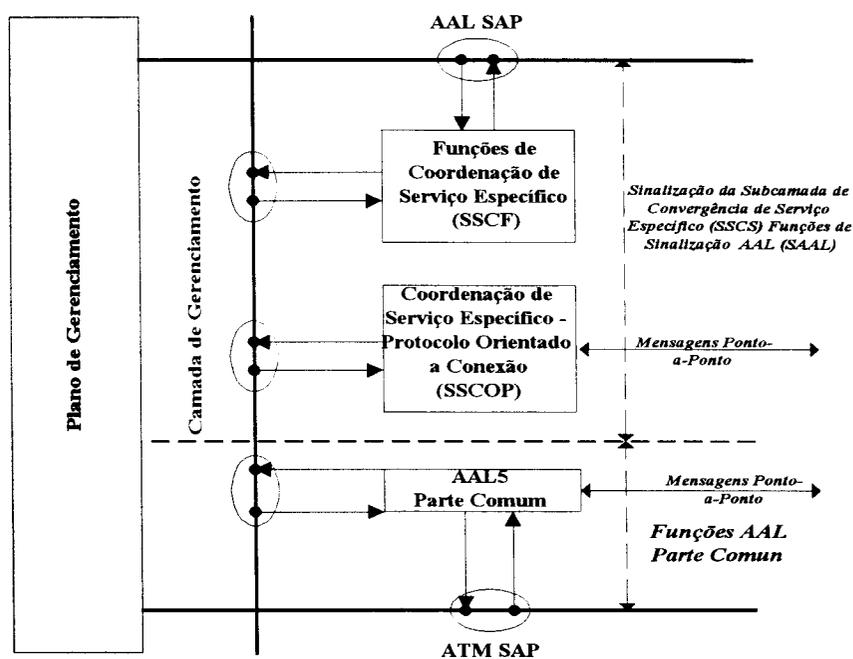
O protocolo de sinalização especifica a seqüência de mensagens que devem ser trocadas, as regras para verificação da consistência dos parâmetros, e as ações que devem ser tomadas a fim de estabelecer e liberar conexões da camada ATM. Uma porção importante desta especificação é envolvida com a manipulação do conjunto de erros, mensagens inválidas, parâmetros inconsistentes e um número de outras situações improváveis.

### 2.5.2.1. Sinalização AAL

O CAC recebe e envia mensagens de sinalização Q.2931 e BISUP, encapsuladas em células ATM, caracterizadas por possuírem os campos VCI = 5 e VPI = 0 em seu *header*. É a entidade da chave ATM encarregada de interpretar as mensagens de sinalização das camadas superiores e da camada ATM, segundo McDysan, Spohn (1994).

A figura 2.18 mostra o modelo do protocolo para a sinalização da AAL (SAAL). A Parte Comum AAL (CP-AAL) é a AAL5. Esta porção da SAAL, é composta por dois protocolos:

- *Service Specific Coordination Function (SSCF)*;
- *Service Specific Connection Oriented Protocol (SSCOP)*.



**Figura 2. 18:** Sinalização AAL (SAAL)

Os protocolos de sinalização SSCF, SSCOP e CP-AAL são gerenciados como camadas separadas pelas funções de Gerenciamento de Camadas. O Gerenciamento de Camadas do B-ISDN é responsável pelo ajuste dos parâmetros nos protocolos das camadas individuais, e pela monitoração de seu estado e desempenho.

O SSCF apresenta as seguintes características:

- Independência das camadas básicas;
- Modo de transferência de dados assegurado;
- Transferência de dados sem confirmação;

- Transparência na transmissão da informação;
- Estabelecimento de conexões para o modo de transferência assegurado.

O SSCOP apresenta as seguintes características:

- Integridade de seqüência;
- Correção de erros;
- Informação de erros para a Camada de Gerenciamento;
- Transferência de dados do usuário no modo assegurado e não assegurado;
- Detecção de erros;
- Informação de *status* entre entidades;
- Capacidade de estabelecer, desconectar e sincronizar uma conexão SSCOP.

O modo não assegurado é um protocolo de datagrama não confirmado, similar ao UDP. Uma conexão deve ser estabelecida antes que qualquer dado seja enviado. A capacidade de detecção de erro da AAL5, determina se o *frame* foi recebido com sucesso. O SSCOP requer que o transmissor sonde periodicamente o receptor para detectar se existe um *gap* na seqüência de *frames* recebidos com sucesso. O receptor deve responder à sondagem e, se algumas sondagens não forem respondidas ou perdidas, o transmissor encerra a conexão. Quando o receptor identifica que um ou mais *frames* estão fora de seqüência, o transmissor reenvia os *frames* perdidos.

### 2.5.2.2 Conexão ponto-a-ponto

As figuras 2.19 ilustra um exemplo de uma chamada de *setup* em uma rede ponto-a-ponto. O exemplo emprega uma parte que solicita uma conexão, com endereço ATM **A**, uma rede ATM e uma parte chamada, com endereço ATM **B**. A parte que chama **A** inicia o processo, usando uma mensagem de *setup*, indicando **B** como a parte chamada. A rede roteia a mensagem chamada para a interface física na

qual **B** esta conectada, fornecendo indicações sobre qual VPI/VCI poderiam ser usados caso a chamada seja aceita.

A parte chamada, aceita a mensagem e retorna uma mensagem *connect*, a qual é propagada de volta o mais rápido possível. Uma mensagem *connect acknowledge*, é enviada pela rede para a parte chamada, e da entidade que chamou para a rede, como estágio final para assegurar que a conexão realmente está ativa, segundo McDysan, Spohn (1994).

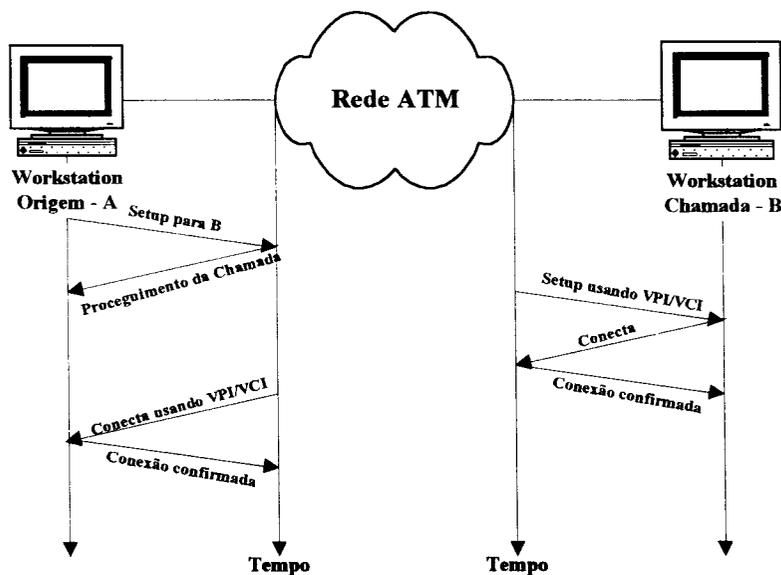


Figura 2. 19: Setup de uma chamada ponto-a-ponto

A figura 2.20 ilustra um exemplo de *release* (liberação), de uma chamada ponto-a-ponto. Qualquer uma das partes pode iniciar o processo de *release*. Neste exemplo, o processo de liberação começa com a entidade que efetuou a chamada, enviando uma mensagem de *release*. A rede propaga a mensagem de *release* até a parte **B**, a qual confirma o pedido de *release*, retornando uma mensagem de *release complete*, a qual é propagada pela rede até **A**, segundo McDysan, Spohn (1994).

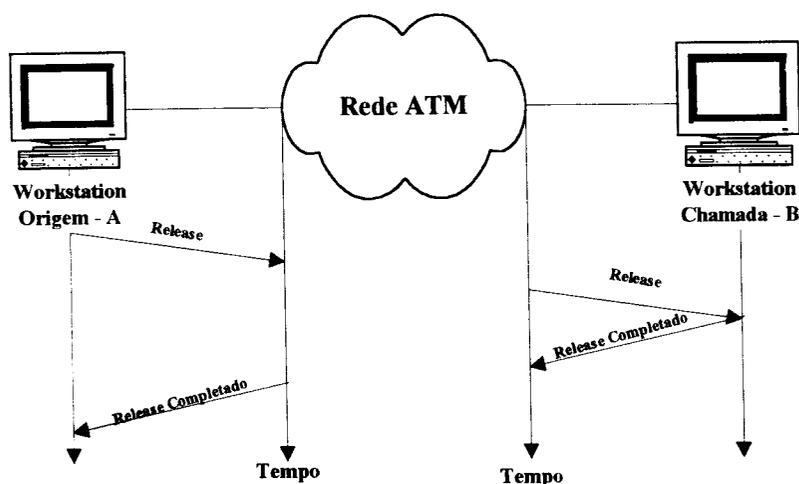


Figura 2. 20: Liberação de uma chamada ponto-a-ponto

### 2.5.2.3 Conexão ponto para multiponto

A figura 2.21 ilustra um exemplo de *setup* de uma chamada ponto-para-multiponto, com origem no nó A e destino nos nós B e C conectados a uma chave ATM local, e um terceiro nó D, conectado a uma UNI ATM separada. O nó A, inicia uma chamada ponto-para-multiponto, enviando uma mensagem de *setup* para a rede, solicitando *setup* de uma conexão ponto-para-multiponto, identificando o nó B, e a rede responde com uma mensagem *call proceeding*, como se fosse uma conexão ponto-a-ponto. A rede transporta a mensagem de *setup* para o nó B, associando os valores de VPI/VCI. O nó B, indica sua intenção de se unir à chamada, retornando para a rede uma mensagem *connect acknowledge*. A rede informa o nó A, do sucesso da adição do nó B, através de uma mensagem *connect* e o nó A retorna uma mensagem de *connect acknowledge*.

Na seqüência, o nó A solicita a adição do nó C, através de uma mensagem *add party*, que a rede retransmite através da UNI ATM, utilizada no nó B, a mensagem *add party*, para informar à chave local a solicitação de adição do nó C. O nó C responde com uma mensagem de *add party acknowledge*, a qual é propagada pela rede até o nó A. O nó A, solicita que o nó D seja adicionado com uma

mensagem *add party*. A rede roteia esta mensagem para a UNI conectada ao nó **D**. O nó **D**, responde com uma mensagem de *connect*, a qual a rede responde com uma mensagem de *connect acknowledge*.

Os nós de uma conexão ponto-para-multiponto, podem ser removidos da chamada pela mensagem *drop party* se uma ou mais partes pertencentes a uma mesma UNI quiserem permanecer na chamada, ou por uma mensagem de *release*, se for o último nó pertencente àquela UNI fazendo parte da conexão, segundo McDysan, Spohn (1994).

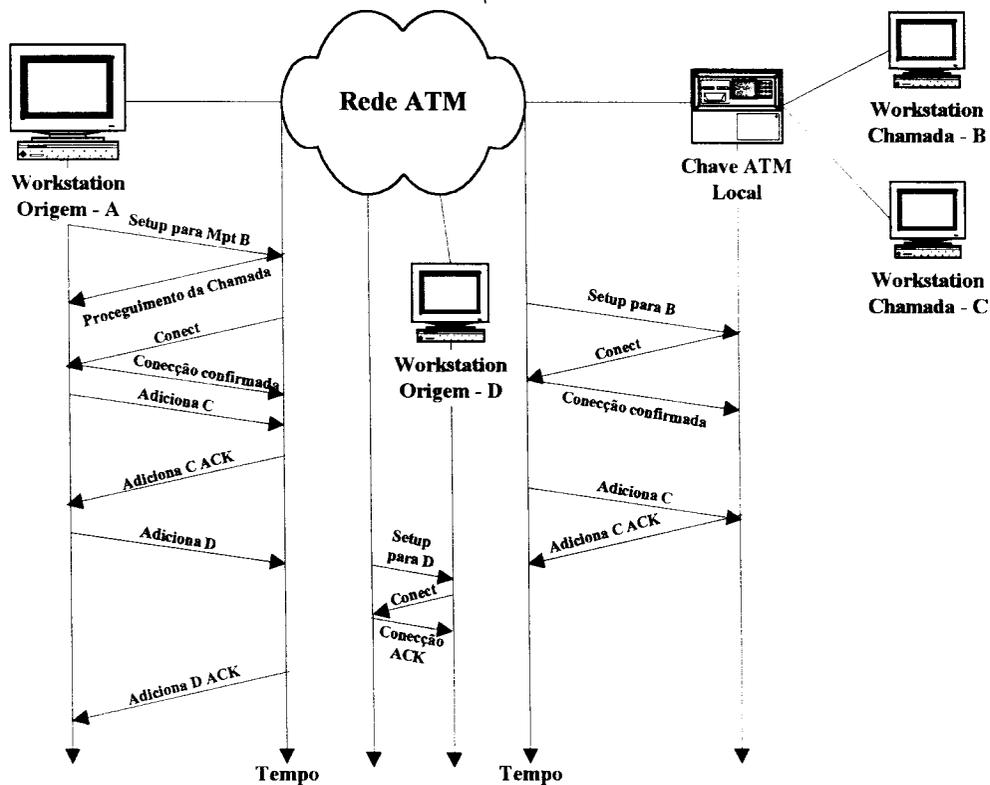


Figura 2. 21: Setup de uma chamada ponto-para-multiponto

## 2.6 Qualidade de Serviço

A qualidade do Serviço (QoS) é definida por parâmetros específicos, para células que estejam em conformidade com o contrato de tráfego. De um modo geral, um usuário solicita uma certa qualidade de serviço (QoS) e certas classes são definidas.

A qualidade do serviço (QoS), segundo McDysan; Spohn (1994), é definida em uma base fim-a-fim, sendo que, a medida é feita considerando-se:

- Células transmitidas por um usuário origem;
- Células transmitidas com sucesso para o usuário destino;
- Células perdidas que não chegam ao usuário destino;
- Células erradas que chegam ao destino mas com erros no *payload*;
- Células misturadas que chegam ao destino mas não foram enviadas pela origem. Isto pode ocorrer devido a um erro não detectado no *header* ou um erro de configuração;

Os parâmetros QoS são definidos em termos dos resultados acima através das seguintes definições:

$$TCP = CP / CT$$

$$TCE = CE / (CTS + CE)$$

$$TCEG = BCEG / TBCT$$

$$TCM = CM / IT$$

onde:

- TCP - taxa de células perdidas;
- CP - células perdidas;
- CT - células transmitidas;
- TCE - taxa de erro de células;
- CE - células erradas;
- CTS - células transmitidas com sucesso;
- TCEG - taxa de células com erros graves;
- BCEG - bloco de células com erros graves;
- TBCT - total de blocos de células transmitidos;

- TCM - taxa de células misturadas;
- CM - células misturadas;
- IT - intervalo de tempo.

O atraso na transferência de células é composto pelos componentes, ilustrados na figura 2.22:

- T1 = Atraso na codificação e decodificação;
- T11 = Atraso na codificação;
- T12 = Atraso na decodificação;
- T2 = Atraso na segmentação e remontagem;
- T21 = Atraso na segmentação no lado emissor da AAL;
- T22 = Atraso na acomodação e remontagem no lado AAL do receptor;
- T3 = Atraso na transferência de células (fim-a-fim);
- T31 = Atraso na transmissão dos nós ATM internos;
- T32 = Atraso no processamento de todos os nós ATM (devido a filas, roteamento, comutação, etc.).

Os atrasos podem ocorrer no lado emissor e no lado receptor de um terminal final, em nós intermediários, e sobre os *links* de transmissão das conexões ATM.

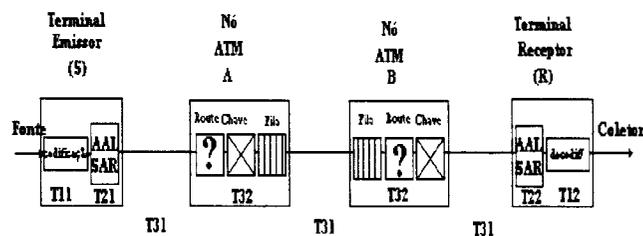


Figura 2. 22: Ilustração das fontes de atraso

Sendo assim, o atraso na transferência de células é o resultado de um componente aleatório T32, mais um atraso de propagação fixo T31. As classes de qualidade de serviço (QoS), segundo McDysan, Spohn (1994), foram definidas

principalmente em termos de alguns parâmetros definidos pelo CCITT I.350, para cada ATM VPC ou VCC:

- Atraso médio;
- Variação no atraso de células;
- Perdas sobre CLP = 0 para células ATM;
- Perdas sobre CLP = 1 para células ATM;
- Taxa de erros.

### 2.6.1 Classes QoS

Para simplificar as coisas para o usuário, um pequeno número de classes QoS foram definidas, conforme a tabela 2.2 pelo ATM Forum e especificadas pela UNI 3.1:

**Tabela 2. 2 :** *Classes QoS definidas pelo ATM Forum e especificadas pela UNI 3.1*

<b>Classe QoS</b>	<b>Parâmetros QoS</b>	<b>Aplicação</b>
<b>0</b>	<b>Não Especificados</b>	<b>Melhor esforço e risco</b>
<b>1</b>	<b>Especificados</b>	<b>Emulação de circuito, CBR</b>
<b>2</b>	<b>Especificados</b>	<b>Áudio/Vídeo VBR</b>
<b>3</b>	<b>Especificados</b>	<b>Orientado a Conexão</b>
<b>4</b>	<b>Especificados</b>	<b>Não Orientado a Conexão</b>

### 2.6.2 Classes QoS especificadas

Uma classe QoS especificada provê um desempenho para uma conexão virtual ATM (VCC ou VPC), como um subconjunto dos parâmetros de desempenho especificados para o ATM. Para cada classe QoS especificada, existe um valor *default* definido para cada parâmetro de desempenho, podendo estes serem alterados conforme a necessidade da conexão. Inicialmente, cada rede define os parâmetros de desempenho, para cada uma das classes QoS conforme a Recomendação I.362 do ITU-T, segundo Ginsburg (1996).

Classes de serviço de transporte:

- **Serviço classe A:** serviço orientado a conexão, com taxa fixa de bits e requisitos de tempo entre os pontos finais da comunicação, emulação de circuito e serviços vídeo a uma taxa de bit constante;
- **Serviço classe B:** serviço a taxa de bit variável do tipo áudio e vídeo;
- **Serviço classe C:** serviço orientado a conexão, com taxa variável de bits e sem requisitos de tempo entre os pontos finais da comunicação;
- **Serviço classe D:** serviço de transferência de dados não orientado a conexão;
- **Serviço classe X:** serviço orientado a conexão no qual todos os parâmetros de tráfego são definidos pelo usuário. Esta classe não é especificada pela Recomendação I.362 do ITU-T;
- **Serviço classe Y:** serviço a taxa de bits disponível, destinado a atender fontes, que podem variar sua taxa de transmissão, mas que requer uma garantia do serviço. Esta classe não é especificada pela Recomendação I.362 do ITU-T.

As seguintes classes QoS foram definidas pelo ATM Forum, e num futuro, outras classes QoS poderão ser definidas. Para uma dada classe de serviços tem-se:

- **Especificação QoS classe 0:** suporta serviço do tipo “melhor esforço”, também chamada de classe QoS não especificada, pois não há quantificação de qualquer dos parâmetros de qualidade;
- **Especificação QoS classe 1:** suporta uma QoS com função de serviço classe A como requisitos de desempenho. Esta classe pode atingir um desempenho comparável à desempenho das linhas digitais privadas. Pode ser utilizada, por exemplo, em aplicações que usem uma fonte de sinais de vídeo sem compressão de dados;
- **Especificação QoS classe 2:** suporta uma QoS com função de serviço classe B, como requisitos de desempenho. Esta classe

possui a intenção de empacotamento de vídeo e áudio em aplicações de videoconferência e multimídia;

- **Especificação QoS classe 3:** suporta uma QoS com função de serviço classe C, como requisitos de desempenho. Esta classe possui a intenção de interoperação de protocolos orientados a conexão, tal como o Frame Relay;
- **Especificação QoS classe 4:** suporta uma QoS com função de serviço classe D, como requisitos de desempenho. Esta classe possui a intenção de interoperação de protocolos não orientados a conexão, tais como IP ou SMDS.

Estas duas classificações não são independentes, nem todas as combinações são permitidas ou fazem sentido prático. A escolha da classe de transporte precede e condiciona a escolha da classe QoS. Por exemplo, não faz sentido especificar uma conexão de transporte Classe C e Classe 1 para QoS. É viável, no entanto, atribuir a uma conexão Classe X, quaisquer das classes QoS.

### 2.6.3 QoS não especificadas

Serviços usando classes QoS não especificadas podem possuir parâmetros de tráfego especificados explicitamente.

Um exemplo de aplicação de classes QoS não especificadas é o suporte de serviços de melhor esforço, onde efetivamente nenhum parâmetro de tráfego é especificado. Para este tipo de serviço “melhor esforço”, o usuário não especifica efetivamente qualquer parâmetro de tráfego e não conta com um compromisso de desempenho da rede. A interpretação e a definição clara do que venha a ser serviço do menor esforço, é uma das atividades do ATM Forum. O nome formal para este tipo de serviço é *Unspecified Bit Rate* (UBR). Um fluxo adaptativo, de serviços controlados foi definido pelo ATM Forum como serviço a *Available Bit Rate* (ABR), a qual é similar aos serviços ANSI Classe Y.

#### 2.6.4 Aplicações das diferentes QoS

Várias aplicações necessitam de diferentes QoS. Por exemplo: o atraso na transmissão de voz não pode ser superior a 15 ms, sem a presença de canceladores de eco. O empacotamento de voz pode aceitar quase que 1% de perdas de células sem conseqüências desagradáveis para o ouvinte. Nem toda aplicação possui seus requisitos bem definidos; porém, existem alguns requisitos gerais que podem ser resumidos nos parágrafos seguintes.

As aplicações de vídeo dependem de vários fatores: o algoritmo de codificação utilizado, o grau de movimento da imagem e a resolução desejada. A perda na generalidade causa alguma degradação na imagem, variando de porções distorcidas da imagem até a perdas de pacotes inteiros. Também, se as variações no atraso forem superior a 20 ou 40 ms, podem causar uma perceptível distorção na reprodução da imagem.

A combinação de áudio e vídeo é muito sensível ao atraso. A percepção humana é muito atenuada para uma correta correlação de áudio e vídeo, que está evidente em alguns filmes dublados. Já as aplicações de transferência de arquivos possui uma sensibilidade marcante a perdas que pode resultar em muitas retransmissões indesejadas ocasionando uma redução da largura de banda utilizável e às variações do atraso. A sensibilidade é descrita para o tempo de espera, o algoritmo de identificação de perda e as aplicações de estratégias de retransmissão.

Aplicações interativas são também sensíveis a perdas e variação no atraso, devido a retransmissão e resposta inconsistente no tempo, que podem diminuir a produtividade. Resposta consistente no tempo pode resultar em como os usuários irão sentir a qualidade dos serviços de dados.

Computação distribuída e aplicações de banco de dados podem ser muito sensíveis ao atraso absoluto, perdas e variações no atraso. O ideal para estes tipos de aplicações é uma largura de banda infinita sem latência.

## **2.7 Controle de tráfego em redes ATM**

### **2.7.1 Introdução**

O tráfego em uma rede ATM é estabelecido mediante o conceito de um contrato, seguindo as especificações do ITU-T I.371 e do ATM Forum. O controle de tráfego segundo McDysan, Spohn (1994), fornece os meios que permitem ao usuário assegurar que o fluxo de células apresentado, encontra-se na taxa especificada por seu contrato de tráfego, e os meios para que a rede garanta a QoS negociada com todas as conexões da rede.

### **2.7.2 Contrato de tráfego**

Em essência, um contrato de tráfego existe para cada Conexão de Caminho Virtual (VPC) ou Conexão de Canal Virtual (VCC). O contrato de tráfego é um arranjo entre um usuário e uma rede através da UNI, considerando os seguintes aspectos para qualquer fluxo de célula ATM em um VPC ou VCC:

- Qualidade do serviço (QoS);
- Os parâmetros do tráfego que especificam as características do fluxo de células;
- A regra de checagem utilizada para interpretar os parâmetros do tráfego;
- A definição da rede para o compromisso da conexão.

A definição do compromisso da conexão permite alguma liberdade na realização da verificação do fluxo de células do usuário.

#### **2.7.2.1 Modelo de referência**

A base do contrato de tráfego é uma configuração chamada terminal equivalente, que pode ser visualizado na figura 2.23.

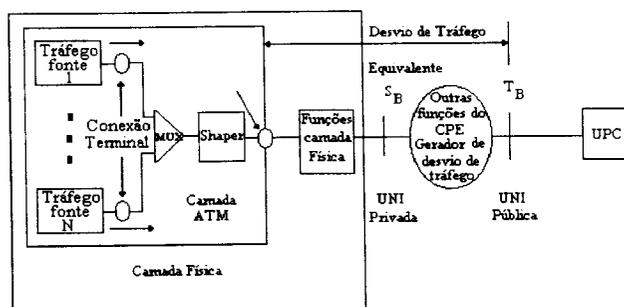


Figura 2. 23: Modelo de um terminal equivalente

Como pode ser visto um terminal equivalente não necessariamente se trata de um dispositivo real; na verdade ele pode ser uma coleção de dispositivos, conforme mostrado na figura 2.23. O tráfego de células ATM é gerado por um certo número de fontes de células, por exemplo, um certo número de *Workstations*, onde cada uma possui seu próprio VPC ou VCC conectado a um terminador. Estes são todos conectados a um multiplexador de células, o qual dentro de uma implementação distribuída pode ser uma chave ATM local, um roteador ou um Hub. Associado com as funções de multiplexação está o controlador de tráfego, que assegura que a fila de células esteja em conformidade com o conjunto de parâmetros de tráfego, definidos por um particular algoritmo de verificação. A saída do controlador de tráfego está na camada física PHY, no seu *Service Access Point* (SAP).

Após a função de controle de tráfego, algumas funções da camada física (e outras), podem alterar o fluxo real de células emitido sobre uma UNI ATM privada (ou ponto de referência  $S_B$ ). Esta fila de células ATM pode então ser chaveada através de outro CPE, para um pequeno *backbone* ATM, antes que ela seja liberada para uma rede pública ATM UNI (ou ponto de referência  $T_B$ ).

O modelo de referência QoS fim-a-fim pode conter uma ou mais redes separadas, cada uma com múltiplos nós conforme mostrado na figura 2.24. Cada uma dessas redes pode introduzir uma flutuação adicional no fluxo de células, devido a multiplexação e chaveamento, comprometendo desse modo a QoS. A princípio, o usuário não pode estar preocupado a respeito de quantas redes interligadas existam e

quais suas características. Não cabe ao usuário se preocupar com a forma que a rede se encontra interligada, ou quais as características que ela possui; o que importa segundo Prycker (1994) é que a rede seja capaz de garantir a QoS para todas as conexões. Porém, isto nem sempre é possível, quando se tem múltiplas redes interligadas com suas próprias características. Quando isto ocorre, o processo passa por uma renegociação na tentativa de atender a conexão da melhor forma possível, não havendo um acordo a conexão é momentaneamente recusada.

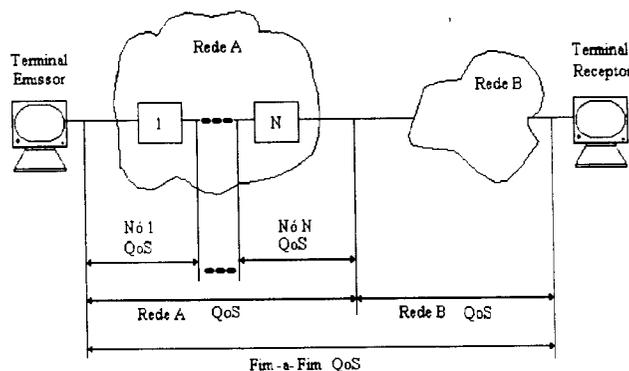


Figura 2. 24: Modelo de referência QoS fim-a-fim

### 2.7.2.2 Descritor de tráfego

O descritor de tráfego, é uma lista de parâmetros, que captura as características intrínsecas da fonte de tráfego. A seguir tem-se uma lista de parâmetros definidos pelo ATM Forum especificação UNI versão 3.0. McDysan; Spohn (1994)

- Uma taxa de pico de células (PCR) em células/segundo em conjunto com um CDV, tolerância em segundos;
- Uma taxa de células de manutenção (SCR) opcional, em células por segundo (sempre menor ou igual ao PCR), em conjunto com o máximo tamanho da rajada de células (MBS).

A figura 2.25 ilustra alguns conceitos importantes que caracterizam um contrato de tráfego. As afirmações seguintes resumem o estado corrente do ATM Forum:

- Taxa de pico de células (PCR) =  $1/T$  em unidades de células por segundo, onde  $T$  é o mínimo espaçamento entre células por segundo (por exemplo: o intervalo de tempo do primeiro bit de uma célula e o primeiro bit da próxima célula);
- ⊙ Variação do atraso de células (CDV) Tolerância =  $\tau$  em segundos. Este parâmetro de tráfego normalmente não pode ser especificado pelo usuário, mas sim pela rede. O número de células que podem ser enviadas *back-to-back*, a taxa de acesso à linha é de  $\tau/T + 1$  como mostrado na figura 2.25;
- Taxa de manutenção das células (SCR) é a taxa média máxima da rajada, em que uma fonte de tráfego *on-off* pode estar enviando a uma taxa de pico, tal como o ilustrado na figura 2.25;
- Tamanho máximo da rajada (MBS) é o número máximo de células que podem ser enviadas na taxa de pico.

A figura 2.25 descreve o intervalo mínimo de rajada como sendo  $T_i$ , o qual está relacionado com SCR e MBS conforme o equacionamento descrito na figura 2.25. A máxima duração de uma rajada é definida como  $T_b$ . Estas informações podem ser úteis para compreender os parâmetros de tráfego, mas não são parte de um contrato formal de tráfego.

A recomendação I.371 do ITU-T, especifica apenas a taxa de pico das células. O ATM Forum adicionou a especificação da taxa de manutenção das células e do tamanho máximo da rajada, de forma padronizada depois da definição do PCR, para um melhor modelo do tráfego de dados.

A figura 2.25, porém, não representa uma definição rigorosa dos parâmetros de tráfego.

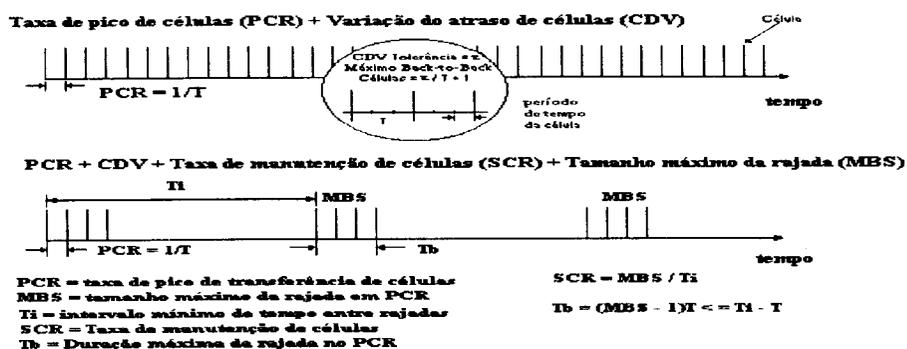


Figura 2. 25: Ilustração dos principais parâmetros de tráfego em redes ATM

### 2.7.3 Parâmetros de controle Usuário/Rede

Todas as ações assumidas por uma rede ATM com relação ao fluxo de células de um usuário, ou outra rede, são chamadas respectivamente de: *Usage Parameter Control (UPC)* e *Network Parameter Control (NPC)*. Genericamente poderia se empregar o termo *policing*, como uma analogia à sociedade civil, onde policiais são os agentes responsáveis por garantir o cumprimento das leis Prycker (1994).

Os mecanismos UPC/NPC devem garantir os recursos de largura de banda e espaço reservado de buffer, para cada usuário, de acordo com o que foi estabelecido em seu contrato de tráfego. Na ausência do mecanismos de *policing*, situações onde um usuário utilize recursos além daqueles que tem direito, mediante seu contrato de tráfego, podem ocorrer.

Não existe um padrão que especifique exatamente, como as funções do UPC ou NPC são implementadas; de fato o desempenho de qualquer implementação UPC/NPC é especificado apenas com base no algoritmo *Leaky Bucket*. Existe um compromisso, no contrato de tráfego, que especifica o quão flexível deve ser o UPC. Um outro requisito é que o UPC, não deve exercer as ações de *policing* sobre mais que uma fração das células que estão em conformidade com as regras do *Leaky Bucket*; em outras palavras, o UPC, não pode ser muito justo.

O UPC, deve se moldar as flexibilidades do contrato de tráfego, o que inviabiliza muitas vezes sua implementação utilizando o *Leaky Bucket*. A seguir serão dados três exemplos de implementação de UPC/NPC.

### 2.7.3.1 O mecanismo *Leaky Bucket* UPC/NPC

A figura 2.26, ilustra um exemplo de *Leaky Bucket*. Nela três duendes: *Identificador*, *Descartador* e *Monitor*, juntos com o duende *Descartador*, ilustram as possíveis ações do UPC. A seqüência de células que estão chegando, está ilustrada no eixo horizontal, no topo da figura 2.26, com o duende *Descartador* despejando o fluído de células que não estão em conformidade com o contrato de tráfego. O *Identificador*, *Descartador* e *Monitor*, operam todos em conjunto. *Identificador*, seta o bit CLP para 1 (indiferente do seu valor de entrada), e permite que a célula passe com sua posição inalterada. O *Descartador* simplesmente não permite que a célula seja transmitida. O *Monitor* registra em seu bloco de anotações quantas células não estão em conformidade com seu contrato de tráfego.

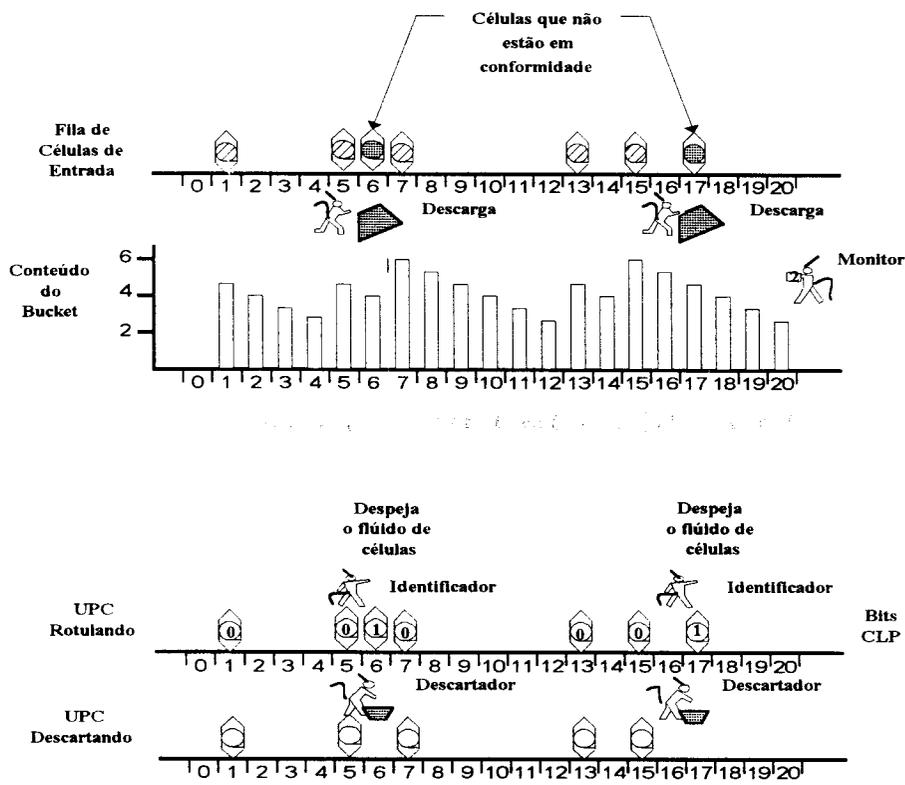


Figura 2. 26: Exemplo de UPC empregando Leaky Bucket

### 2.7.3.2 O mecanismo *Sliding and Jumping Window* UPC

Os dois mecanismos UPC baseados no conceito de janelas, são comparados com o *Leaky Bucket*, para ilustrar como diferentes implementações, designadas para policiar a mesma porção de células que não estão em conformidade, podem policiar células diferentes, e até mesmo porções diferentes de células que não estão em conformidade. Para qualquer parâmetro de um *Leaky Bucket* simples existe uma relação equivalente para uma janela de  $M$  células, em  $N$  tempos de células. Na figura 2.27,  $M = 3$  e  $N = 10$ . Para os dois mecanismos UPC, em janelas, dois novos duendes foram criados: *Slide* e *Jump*.

No protocolo *Sliding Window*, uma janela de  $N$  tempo de células, se move para a direita por “*Slide*”, a cada tempo de célula. Se houver menos de  $M$  células em  $N$  intervalos de células, nenhuma ação do UPC será realizada. Porém, se

por uma unidade de deslocamento da janela para a direita,  $M$  ou mais células forem incluídas dentro da janela, então, o UPC pega todas as células a partir da  $M$ -ésima célula dentro da janela. Este processo se repete a cada tempo de célula.

No *Jumping Window*, uma janela se move em saltos de  $N$  unidades para a direita, a cada  $N$  tempo de células. A mesma regra  $M$ , fora  $N$ , é aplicada. De um modo geral, um mecanismo UPC que emprega *Jumping Window*, é mais tolerante que um *Slide Window*.

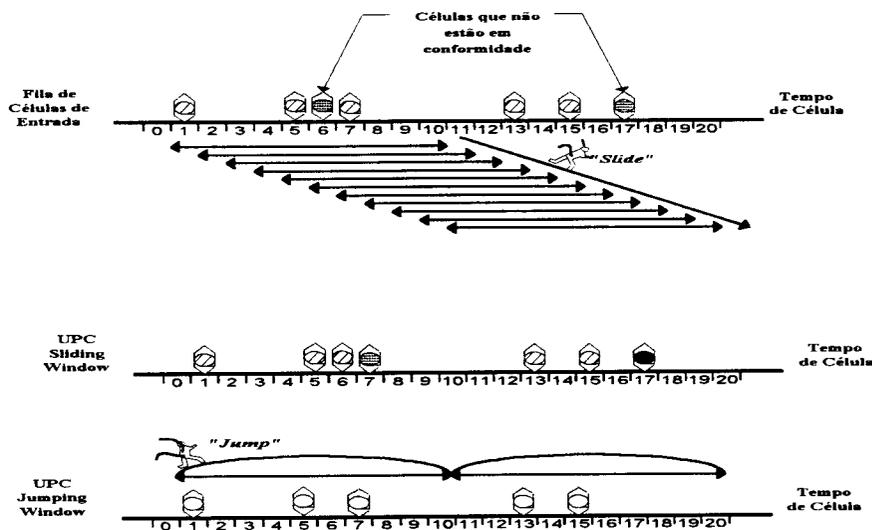


Figura 2. 27: Exemplo de UPC utilizando janelas

No exemplo da figura 2.27, o UPC *Jumping Window*, foi mais flexível que os métodos UPC, que empregam *Leaky Bucket* ou *Slide Window*.

O fato de que diferentes algoritmos UPC, ou o mesmo algoritmo, pode policiar diferentes células, é chamado de medida de distorção (*skew*), e esta é a razão do desempenho do UPC, se originar a partir da fração de células que estão em conformidade. A figura 2.28 ilustra um caso patológico, onde a diferença entre um UPC *Leaky Bucket*, *Sliding Window* e *Jumping Window* é mais acentuada.

Os parâmetros de cada algoritmo foram definidos para admitir uma célula a cada três tempos de células, e permitir no máximo duas células *back-to-back*. O *UPC Leaky Bucket* teve um incremento de três células, e seu *bucket*, uma profundidade de cinco células, enquanto que, os algoritmos de janelas, foram especificados com os seguintes parâmetros:  $M = 2$  e  $N = 6$ . A figura 2.28, ilustra a chegada de 9 células. O *UPC Leaky Bucket*, identifica 20% como não estando em conformidade com o contrato de tráfego, o *Sliding Window* identifica 40% com não estando em conformidade, e o *Jumping Window* identifica 30% das células como não estando em conformidade.

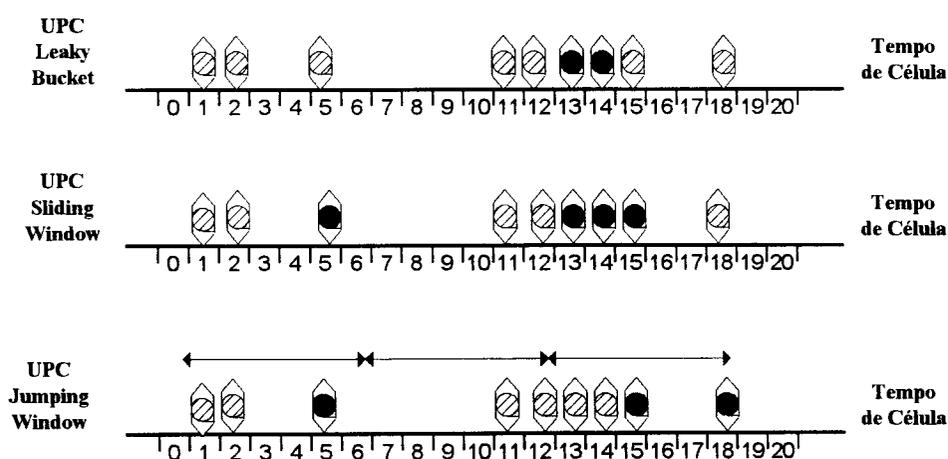


Figura 2. 28: Exemplo patológico das diferenças dos UPCs

### 2.7.3.3 Outros métodos de controle de tráfego

Existem muitos outros métodos de controle de tráfego em redes ATM. Novas definições de parâmetros de tráfego, funções de policiamento, e prioridade de filas, estão sendo desenvolvidas, e incluídas nas novas linhas de produtos ATM. Muitos métodos já desenvolvidos se mencionados, necessitariam de um tratamento todo especial, o que está fora do escopo deste trabalho. Os métodos aqui apresentados procuram dar uma visão geral do controle de tráfego nas redes ATM.

## **Capítulo 3**

*Este capítulo analisa uma chave ATM, a distribuição de seus blocos funcionais e destaca as funções do B-ISDN, diretamente relacionadas com a retransmissão de células. É feito também uma pequena abordagem a respeito do SONET e da organização das células ATM no payload SONET.*

### **A chave ATM**

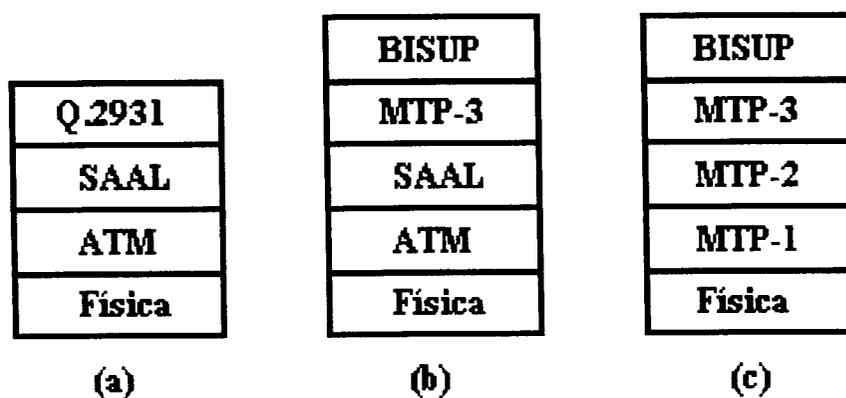
#### **3.1 Introdução**

As chaves ATM, embora sendo uma evolução das tradicionais chaves de telefonia, são muito mais que simples estruturas de roteamento e buffers de células. Em uma rede ATM, a chave se destaca por ocupar uma posição central, ou seja, toda conexão entre dois usuários finais deve, obrigatoriamente, passar por ao menos uma chave. Além das funções de retransmissão de células, a chave deve realizar as funções de controle e gerenciamento, relativas ao Plano de Controle e Plano de Gerenciamento do B-ISDN, respectivamente. A distribuição de todas estas funções segundo Chen, Liu (1994) e Kim et al. (1994) dentro de um sistema compõe uma chave ATM.

#### **3.2 Plano de Controle**

O Plano de Controle é o responsável pelas funções de controle relacionadas com: estabelecimento da chamada e supervisão e liberação dos VPCs/VCCs. Uma chamada pode consistir de múltiplas conexões, e qualquer conexão pode ser estabelecida, modificada ou finalizada durante uma chamada.

O Plano de Controle é dividido em camadas, e compartilha as mesmas camadas física e ATM com os demais planos do B-ISDN (figura 1.12). Como consequência, as células de sinalização são transportadas na camada ATM, da mesma forma que as células de dados do usuário. Na chave, as células de dados dos usuários são roteadas de uma porta de entrada para a porta de saída apropriada mediante o processamento do seu *header*. As células de sinalização no entanto, devem ser processadas pela chave. A estrutura do Plano de Controle segundo CCITT I.321 (1991), Kim et al. (1994) e Chen, Liu (1995), pode ser vista na figura 3.1. A função da SAAL é adaptar as informações de sinalização das camadas superiores em células ATM de sinalização. O protocolo de sinalização das camadas superiores padronizado pelo ITU-T é o Q.2931 na UNI e o BISUP na NNI. Existem duas possibilidades para suportar o BISUP. A figura 3.1b, mostra o BISUP sendo suportado sobre o ATM pela SAAL e o *Message Transfer Part level 3* (MTP-3). A outra possibilidade mostra o BISUP sendo suportado pelos MTP níveis de 1 a 3, conforme a figura 3.1c.



**Figura 3. 1:** *Suíte de protocolos (a) sinalização de acesso, (b) sinalização entre chaves sobre ATM, e (c) sinalização entre chaves sobre SS7*

As mensagens de sinalização são transportadas pela camada ATM em células de sinalização, identificadas por seus campos VPI e VCI. Na UNI e NNI, os valores reservados para VCI e VPI numa sinalização ponto-a-ponto são: VPI = 0 e VCI = 5, sendo os mesmos valores em ambas as direções.

Sinalização de configuração ponto-para-multiponto, são suportadas pela UNI, quando múltiplos usuários compartilham a mesma UNI. Um VC geral de sinalização *broadcast* pode ser usado por uma chamada que oferece serviço ponto-para-multiponto. Um VC geral de sinalização *broadcast* é indicado por VCI = 4 em qualquer VP.

O conjunto de mensagens Q.2931 para chamada e controle de conexão está listada na tabela 3.1 Mensagens adicionais para conexão ponto-para-multiponto foram incluídas. Os elementos que compõem uma mensagem Q.2931 são exibidos na figura 3.2 e descritos a seguir conforme Coutini et al. (1992), Kim et al. (1994) e Chen, Liu (1995):

	Discriminador de Protocolo	Referência da Chamada	Tipo de Mensagem	Tamanho da Mensagem	Mensagem
bytes:	1	4	2	2	Variável

Figura 3. 2: Formato de uma mensagem Q.2931

- Discriminador de protocolo (1 byte): para identificar mensagens de chamada de controle usuário-rede ou outros propósitos;
- Valor de referência da chamada (4 bytes): para identificar a chamada à UNI local para a qual a mensagem se destina;
- Tipo de mensagem (2 bytes): indica a função da mensagem (tabela 3.1);
- Tamanho da mensagem (2 bytes): indica o número de bytes na mensagem;
- Mensagem (tamanho variável): elementos de informação específica.

O conjunto de mensagens BISUP para chamada e controle de conexão são listados na tabela 3.2. A figura 3.3 ilustra o formato de uma mensagem BISUP segundo Coutini et al. (1992), Kim et al. (1994), Chen; Liu (1995).

	Rótulo de Roteamento	Tipo de Mensagem	Comprimento da Mensagem	Compatibilidade da Mensagem	Conteúdo da Mensagem
bytes:	4 ou 7	1	1 ou 2	1	Variável

Figura 3.3 : Formato de uma mensagem BISUP

- Rótulo de roteamento (4 a 7 bytes): especifica os pontos origem e destino;
- Código do tipo de mensagem (1 byte): define a função e formato da mensagem;
- Tamanho da mensagem (1 a 2 bytes): especifica o número de bytes da mensagem;
- Informação de compatibilidade da mensagem (1 byte): para diferenciar entre diferentes versões de protocolos;
- Conteúdo da mensagem (comprimento variável): parâmetros específicos da mensagem.

**Tabela 3. 1:** Mensagens de sinalização Q.2931 de acesso para chamada (controle de conexão)

Mensagem	Função
<i>Alerting</i>	Enviado pelo destinatário à rede ou pela rede à origem , para indicar que o alerta foi iniciado.
<i>Call proceeding</i>	Enviado pelo destinatário à rede ou pela rede à origem , para indicar que o <i>setup</i> da chamada foi iniciado e que mais informação de <i>setup</i> não é necessária.
<i>Connect</i>	Enviado pelo destinatário à rede ou pela rede à origem, para indicar que a chamada foi aceita pelo destino.
<i>Connect ack</i>	Enviada pela rede ao destinatário para indicar que a chamada foi bem sucedida.
<i>Setup</i>	Enviada pelo usuário à rede e pela rede ao destinatário, para iniciar uma chamada (estabelecimento de uma conexão).
<i>Release</i>	Enviada pelo usuário para solicitar a liberação da conexão, ou pela rede para indicar que a conexão foi liberada.
<i>Release complete</i>	Enviada pelo usuário ou pela rede, para indicar qual o equipamento que fez referência à chamada e o identificador da conexão.
<i>Notify</i>	Enviado pelo usuário ou pela rede para indicar a informação que pertence a uma chamada/conexão.
<i>Status</i>	Enviado pelo usuário ou pela rede para responder a uma solicitação de status ou registrar certas condições de erro a qualquer momento.
<i>Status Enquiry</i>	Enviado pelo usuário ou pela rede para solicitar o status
<i>Add party</i>	Enviado pelo usuário ou pela rede para adicionar uma novo usuário à conexão existente.
<i>Add party ack</i>	Enviado pelo usuário ou pela rede para confirmar que a adição de um novo usuário teve sucesso.
<i>Add party reject</i>	Enviado pelo usuário ou pela rede para indicar que a adição de um novo usuário não teve sucesso.
<i>Drop party</i>	Enviado pelo usuário ou pela rede para liberar um usuário de uma conexão ponto-para-multiponto
<i>Drop party ack</i>	Enviado pelo usuário ou pela rede para confirmar que o pedido para desconectar usuário foi bem sucedido.

**Tabela 3. 2:** Mensagens BISUP para chamadas (Controle da Conexão)

<b>Mensagem</b>	<b>Função</b>
<i>Address complete (ACM)</i>	Retornado pela chave destino, para confirmar que a chamada está prosseguindo ao usuário chamado e indicar o status do usuário chamado.
<i>Answer (ANM)</i>	Retornado pela chave destino, para indicar que usuário chamado aceitou a chamada e que a informação pode começar a ser transferida.
<i>Call progress (CPG)</i>	Enviado em qualquer direção, para indicar que um evento para iniciar ou finalizar o acesso ocorreu.
<i>Initial address (IAM)</i>	Enviado pela chave origem para propagar a solicitação de chamada para a chave destino.
<i>IAM ack (IAA)</i>	Retornada por uma chave para confirmar o recebimento da mensagem IAM.
<i>IAM reject (IAR)</i>	Retornado pela chave para rejeitar a mensagem IAM.
<i>Release (REL)</i>	Solicita o término da conexão e libera o identificador de conexão e a referência da chamada.
<i>Release complete (RCL)</i>	Resposta a mensagem de <i>Release</i> , para confirmar a liberação da referência da chamada e o identificador de conexão.
<i>Resume (RES)</i>	Solicita a retomada de uma comunicação após uma suspensão temporária.
<i>Subsequent address (SAM)</i>	Enviado pela chave origem seguindo um IAM para transportar um número adicional de informações dos usuários chamados.
<i>Suspend (SUS)</i>	Pedido de suspensão temporária de comunicação, sem liberação da chamada.

### 3.3 Plano de Gerenciamento

Para sustentar a QoS desejada pela conexão, é necessário monitorar e controlar a rede ATM de tal forma a assegurar uma correta operação e uma utilização eficiente dos recursos da rede. São de competência do Plano de Gerenciamento, as funções de monitorar, controlar e realizar a manutenção da rede. São funções do Plano de Gerenciamento segundo Kim et al. (1994), CCITT i.321 (1994) e Chen, Liu (1995):

- **Gerenciamento de faltas:** detecta dinamicamente, isola e corrige falhas;

- **Gerenciamento de performance:** monitora continuamente, notifica e avalia o comportamento dos elementos da rede;
- **Gerenciamento de configuração:** inicializa serviços e monitora alterações em seu status;
- **Gerenciamento de contabilidade:** coleta e processa informações dos serviços utilizados, para emissão da fatura;
- **Gerenciamento de segurança:** regula e controla o acesso aos bancos de dados da rede.

As operações de gerenciamento empregam células especiais chamadas *Operation and Maintenance (OAM)*. As células OAM, possuem o formato especificado na figura 3.4.

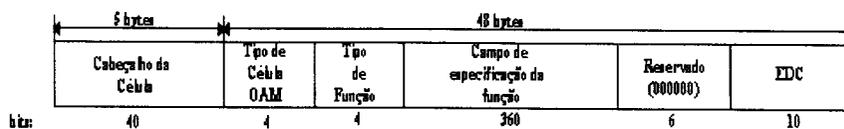


Figura 3. 4: Formato de uma célula OAM

O campo tipo de célula OAM indica o tipo de função de gerenciamento, conforme especificado na tabela 3.3. O campo tipo de função indica a função específica realizada pela célula. O campo EDC é um CRC de grau 10, para detecção de erros.

Tabela 3. 3: Código das células OAM

Tipo de Célula OAM	Valor	Tipo de Função	Valor
<i>Fault Management</i>	0001	<i>Alarm indication signal (AIS)</i>	0000
		<i>Remote defect indicator (RDI)</i>	0001
		<i>Continuity check</i>	0100
		<i>Cell loopback</i>	1000
<i>Performance Management</i>	0010	<i>Forward monitoring</i>	0000
		<i>Backward reporting</i>	0001
		<i>Monitoring &amp; reporting</i>	0010
<i>Activation/deactivation</i>	1000	<i>Performance monitoring</i>	0000
		<i>Continuity check</i>	0001

### 3.3.1 Interface de gerenciamento local (*Interim Local Management (ILMI)*)

Como muitos detalhes a respeito das funções do Plano de Gerenciamento estão em fase de estudo pelo ITU-T, o ATM Forum definiu uma interface de gerenciamento local ILMI baseada no *Simple Network Management Protocol* (SNMP) e um padrão ATM UNI chamado *Management Information Base* (MIB), que permite ao usuário ATM obter informações de status e controle a respeito dos VPCs/VCCs de sua UNI.

De acordo com a especificação ILMI, cada dispositivo de usuário ATM possui uma *UNI Management Entity* (UME), que suporta as funções ILMI. As células ILMI, são identificadas pelos valores de VPI = 0 e VCI = 16.

A ATM UNI MIB de acordo com Prycker (1994) e McDysan, Spohn (1994), contém informações a respeito da camada física, camada ATM, estatísticas da camada ATM e VPCS/VCCS.

### 3.4 Características de uma chave ATM

Toda chave ATM deve ser capaz de desempenhar duas funções básicas. A primeira delas, é a comutação espacial, que garante que toda informação colocada em uma determinada porta de entrada será dirigida para uma determinada porta de saída específica, tendo sido objeto de uma conexão prévia. Um aspecto importante da comutação espacial é o roteamento, isto é, como a informação será internamente roteada da porta de entrada para a porta de saída da chave. A Segunda função da chave é a comutação temporal, que pode ser comparada à comutação de *slot* de tempo em um tempo de chave. A chave ATM pode realizar comutação temporal, ou seja, a chave possui autonomia para alterar a posição na escala de tempo ocupada por um determinado conjunto de células. É por isso que a identificação de tempo fixa no frame STM é substituída por um identificador de canal lógico na célula conforme Prycker (1994) e McDysan, Spohn (1994).

A figura 3.5 exhibe as duas funções básicas de uma chave ATM. Os tópicos a seguir descrevem uma arquitetura de chave ATM.

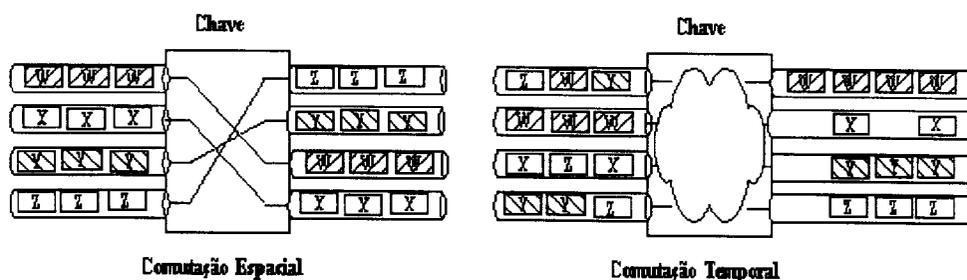


Figura 3.5 : Característica de uma chave ATM sob o ponto de vista do usuário

### 3.5 Arquitetura de uma Chave ATM

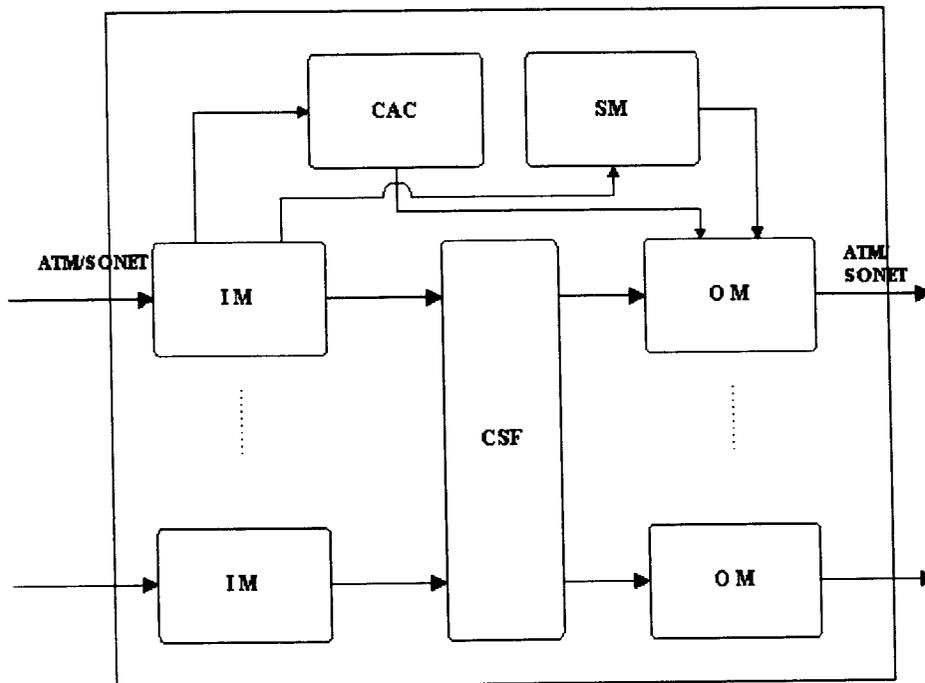
De um modo geral, uma chave é constituída por um conjunto de portas de entrada e saída, através das quais a chave é interconectada a usuários e outros elementos da rede. A chave ATM tem por função prover o serviço de retransmissão de células, e deve suportar todas as funções de controle e gerenciamento necessárias para a realização deste serviço.

Como a informação contida no *payload* das células de dados dos usuários é transportada transparentemente pela rede ATM, a chave processa apenas o *header* da célula, pois é nele que se encontram as informações necessárias para a retransmissão da célula. As informações derivadas dos campos VPI/VCI no *header* das células, são usadas para rotear as células através da chave para que esta encontre a porta de saída apropriada. Na porta de saída, as células são preparadas para a retransmissão,

As funções realizadas por uma chave ATM segundo Chen, Liu (1994), são distribuídas em cinco módulos, conforme a figura 3.6:

- **Input Module (IM)** – recebe as células e as prepara para serem roteadas através da chave;
- **Cell Switch Fabric (CSF)** – responsável pelo roteamento das células no interior da chave. Dependendo do projeto da chave, pode fornecer rotas alternativas para o transporte de células de sinalização e gerenciamento para o CAC e SM respectivamente, através de portas especiais;

- **Output Module (OM)** – preparam as células para serem retransmitidas;
- **System Management (SM)** – realiza todas as funções de gerenciamento e controle de tráfego, para assegurar uma correta e eficiente operação da chave;
- **Connection Admission Connection (CAC)** – processa e interpreta as informações de sinalização e decide a respeito da admissão ou rejeição de uma nova conexão.



**Figura 3.6:** Representação em diagrama de blocos dos cinco módulos funcionais de uma chave ATM

A função básica de cada módulo de entrada, considerando um sinal SONET, é finalizar o sinal de entrada e extrair a fila de células ATM. Para isso, deve ser capaz de realizar algumas funções como:

- Conversão do sinal óptico em elétrico;
- Recuperação da fila de bits digitais;

- Processamento do *overhead* SONET;
- Delineação das células;
- Descartar células vazias.
- Posteriormente, ainda no módulo de entrada, as células devem ser preparadas para serem roteadas através da CSF. Para isso, cada célula passa por:
  - Verificação de erro no *header*, através do campo HEC;
  - Validação dos valores VPI/VCI;
  - Classificação das células de sinalização para serem enviadas para o CAC;
  - Classificação das células de gerenciamento para serem enviadas para o SM;
  - Estabelecimento dos parâmetros de controle usuário/rede para cada VPC/VCC a ser policiado;
  - Adição de um rótulo para roteamento interno.

Os rótulos adicionados às células no módulo de entrada são removidos no módulo de saída. Eles podem conter dois tipos de informações: roteamento interno e parâmetros de gerenciamento interno. Para roteamento interno, o rótulo deve conter campos relativos a porta de saída destino, tolerância a perdas, prioridade de atraso, *time stamp*, limite máximo de atraso, tolerância na variação do atraso, identificador de conexão *broadcast/multicast*. O gerenciamento interno deve monitorar o desempenho interno, podendo conter também campos para a identificação do usuário fonte, um número que possibilite identificar a seqüência das células, e o tipo da célula. Como este rótulo só existe internamente, sua especificação fica por conta do projeto da chave conforme Kim (1993), Huang et al. (1994) e Chen, Liu (1994).

As funções do módulo de saída, na sua quase totalidade, consistem em desfazer as funções realizadas pelo módulo de entrada, sendo que a sua principal função é a de preparar as células ATM para serem transmitidas pela camada física. Para isso realiza as funções:

- Remover e processar o rótulo interno de cada célula;

- Transladar, se necessário, os valores VPI/VCI;
- Gerar o campo HEC e inclui-lo no *header* da célula;
- Possibilitar a mistura das células de sinalização e gerenciamento provindas do CAC e do SM respectivamente, com a fila de células de dados dos usuários;
- Desacoplamento de células (adição de células vazias);
- Mapear as células no *payload* SONET;
- Gerar o código extra SONET;
- Converter a fila de bits digitais em sinais óticos;

De uma forma geral, a CSF é a principal responsável pela transferência de células entre os demais blocos funcionais da chave. São funções da CSF segundo Kim (1993), Huang et al. (1994) e Chen; Liu (1994):

- *Buffer* de células;
- Concentração do tráfego e multiplexação;
- Redundância para tolerância a falhas;
- *Multicasting* ou *Broadcasting*;
- Descarte seletivo de células com base na prioridade de perdas;
- Monitorar o congestionamento;
- Escalonar as células com base na prioridade de atraso;

O CAC realiza todas as funções relativas ao estabelecimento, modificação e finalização da conexão. Mais especificamente, o CAC é responsável por segundo Kim (1993), Huang et al. (1994) e Chen; Liu (1994):

- Protocolo de sinalização das camadas superiores;
- Funções de sinalização AAL para interpretar ou gerar células de sinalização;
- Interface com a sinalização da rede;
- Negociar o contrato de tráfego com usuários que requisitam novos VPC(s)/VCC(s);
- Renegociar com os usuários para estabelecer trocas de VPC(s)/VCC(s);

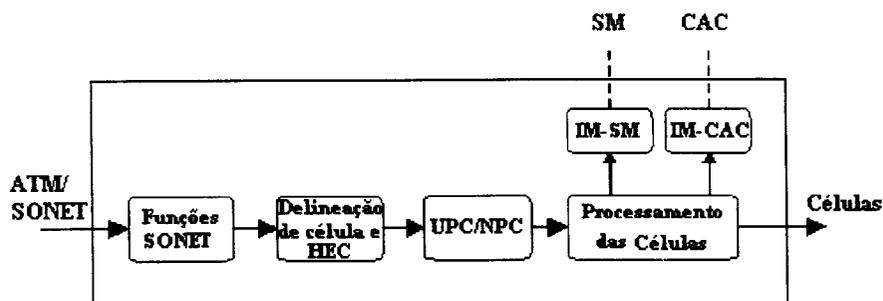
- Alocar recursos da chave para VPC(s)/VCC(s), incluindo seleção de rotas;
- Decidir sobre a admissão/rejeição de VPC(s)/VCC(s) requisitados;
- Gerar parâmetros UPC/NPC.

O Sistema de Gerenciamento é muito complexo de acordo com Kim (1993), Huang et al. (1994) e Chen; Liu (1994), e nem todas as suas funções foram padronizadas. Algumas de suas funções são listadas a seguir:

- Camada física OAM;
- Camada ATM OAM;
- Gerenciamento e configuração dos componentes da chave;
- Controle de segurança para o banco de dados da chave;
- Gerenciamento do tráfego;
- Administrar uma base de informações de gerenciamento;
- Gerenciamento dos clientes da rede;
- Interface com o sistema operacional ou *Telecommunications Management Network* (TMN<sup>12</sup>);
- Medir a utilização dos recursos da chave para emissão de fatura.

### **3.6 Módulo de entrada (IM)**

É o módulo mais complexo da chave, sendo composto por seis blocos funcionais básicos Funções SONET, Delineação de Células & HEC, UPC/NPC, Prócessamento das Células IM-SM (opcional) e IM-CAC (opcional), que realizam as funções da camada física, da camada ATM, do CAC e do SM. A figura 3.7 apresenta o diagrama esquemático dos blocos funcionais do módulo de entrada.



**Figura 3. 7:** Diagrama de bloco dos componentes funcionais do Módulo de Entrada de uma chave ATM

As funções da camada física são realizadas pelos blocos Funções SONET e Delineação de Células e HEC. As funções da camada ATM são de competência dos blocos UPC/NPC e Processamento de Células.

O padrão *Synchronous Optical Network* (SONET), é a versão americana do ITU-T para o *Synchronous Digital Hierarchy* (SDH) europeu. SONET e SDH são compatíveis na sua quase totalidade, sendo a taxa básica da hierarquia SONET-STS<sup>13</sup> de 51.84Mbps (STS-1), ao passo que a hierarquia SDH-STM têm, como taxa básica, 155.52Mbps (STM-1). A taxa de transferência do SONET está entre 51.84 Mbps à 2.488 Gbps, podendo ser estendida para mais de 13 Gbps, o que lhe garante muita flexibilidade em suportar futuras expansões de serviços em banda larga. As considerações a respeito do SONET se referem ao Manual da Hewlett Packard – Introduction to SONET Network and Test (1994).

A figura 3.8 mostra a composição de um *frame* SONET STS<sub>N</sub>, onde N indica um múltiplo de 51,84 Mbps dentro da hierarquia SONET (ex. STS<sub>3</sub> SONET a 155,52 Mbps). Um frame STS consiste de duas partes: o *Synchronous Payload Envelope* (SPE) e o *Transport Overhead*. A ordem de transmissão de um frame STS é linha a linha, da esquerda para a direita, sendo que, em cada byte o bit mais significativo é transmitido primeiro.

<sup>12</sup> TMN: é especificado pelo ITU-T na REC. M.3010, Principles for a Telecommunications Management Network.

<sup>13</sup> STS<sub>N</sub>: A hierarquia STS, diz respeito ao emprego de cabos de cobre como meio físico (sinal elétrico). Quando SONET é empregado em fibra ótica, utiliza-se a hierarquia OC<sub>N</sub>.

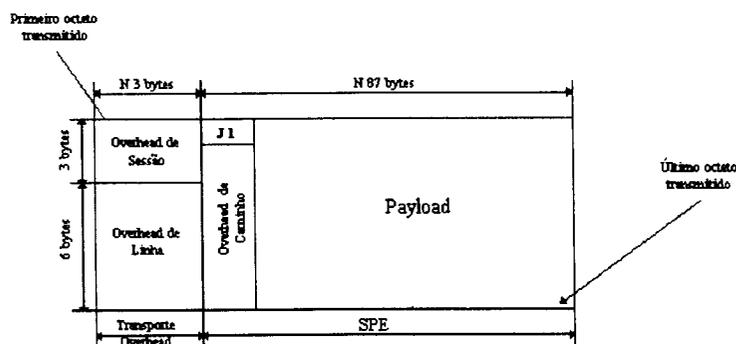


Figura 3. 8: Formato de um frame SONET

Um *frame* STS<sub>1</sub> é composto por 9 linhas e 90 colunas, formando uma matriz de bytes com duração de 125  $\mu$ s, o que equivale a dizer que cada byte no STS<sub>1</sub> pode ser considerado como um canal de 64 kbps (8.000 quadros por segundo).

A estrutura do *overhead* SONET é tal que os elementos da rede têm acesso apenas às informações necessárias para realizações de suas funções, o que permite a simplificação de alguns elementos de rede, porque nem todo *overhead* precisa ser processado. A característica do *overhead* SONET, que permite a simplificação dos elementos da rede é o fato de que seus ponteiros estão organizados de tal forma a atender as especificações de cada camada SONET (Física, Sessão, Linha e Caminho), sendo criado ou processado apenas pelos elementos da rede que finalizam as funções daquela camada.

No *payload* SONET, encontram-se o conjunto de células ATM, com 53 octetos cada. Estas células estão mapeadas no *payload* como se ocupassem um *container* imaginário de 2340 octetos (considerando um *frame* STS<sub>3</sub>). Sendo assim, o *container* pode ter início em qualquer parte do *payload*. Haverá um problema de alinhamento das células dentro do *container*, pois 2340 não é múltiplo de 53, ou seja o *container* terá capacidade para 44,15 células ATM. Uma importante informação contida no *overhead* de caminho diz respeito ao ponto do *container* onde tem início a primeira célula.

Cabe à subcamada de convergência de transmissão do modelo ATM, a missão de montar e desmontar o *frame* SONET/SDH.

A figura 3.9 esquematiza o *container* SONET/SDH.

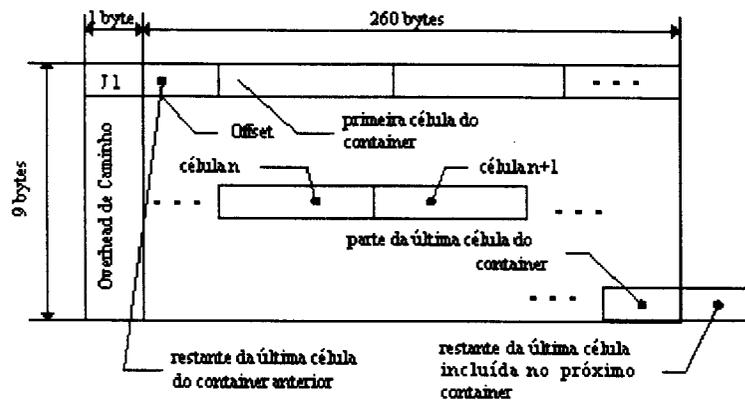


Figura 3. 9: Diagrama esquemático do container SONET/SDH

### 3.6.1 Funções SONET

É o bloco encarregado de extrair o *payload*  $STS_N$  e passar seu conteúdo para o bloco de Delineação de Células, para extração e delineação das células ATM.

Após a delineação, as células estão prontas para serem processadas pela camada ATM. O processamento da camada ATM é comum a todos os tipos de células: células do usuário, células de sinalização e células de gerenciamento.

São dois os blocos responsáveis pelas funções da camada ATM: UPC/NPC e Processamento de Células.

O gerenciamento da camada física é supervisionado pelo SM ou IM-SM. O SM ou IM-SM, recebe o alarme SONET, registra as informações do *overhead*, e coordena os teste das funções SONET segundo Obara (1990), Balboni et al. (1992), Chen; Liu (1994) e McDysan; Spohn (1994).

### 3.6.2 Delineação de Células e Controle de erros no *Header*

O bloco Delineação de Células e Controle de Erros no *Header*, realiza três funções da camada física descritas a seguir segundo Obara (1990), Balboni et al. (1992) e Chen; Liu (1994):

- Delineação da célula;
- Controle de erros no *header*;
- Desacoplamento de células.

O campo HEC contido no *header* da célula ATM permite a correção de um único bit no *header* da célula e a detecção de múltiplos erros nos cinco bytes do *header*.

A figura 3.10 mostra as células ATM:

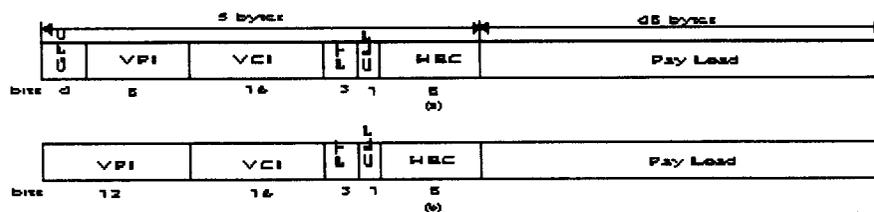


Figura 3. 10: Células ATM a) UNI e b) NNI

Para que uma célula possa ser transmitida, é necessário que o campo HEC seja gerado após os 4 primeiros bytes do *header*. Para se obter o HEC, a princípio os 8 bits são zerados e o *header* convertido em um polinômio de grau 39. Em seguida, este polinômio é dividido por um polinômio gerador de grau 8 para se obter os 8 bits que serão adicionados ao código padrão “01010101”, para compor o HEC da célula.

A figura 3.11 mostra o diagrama de estados do procedimento para delimitação de células.

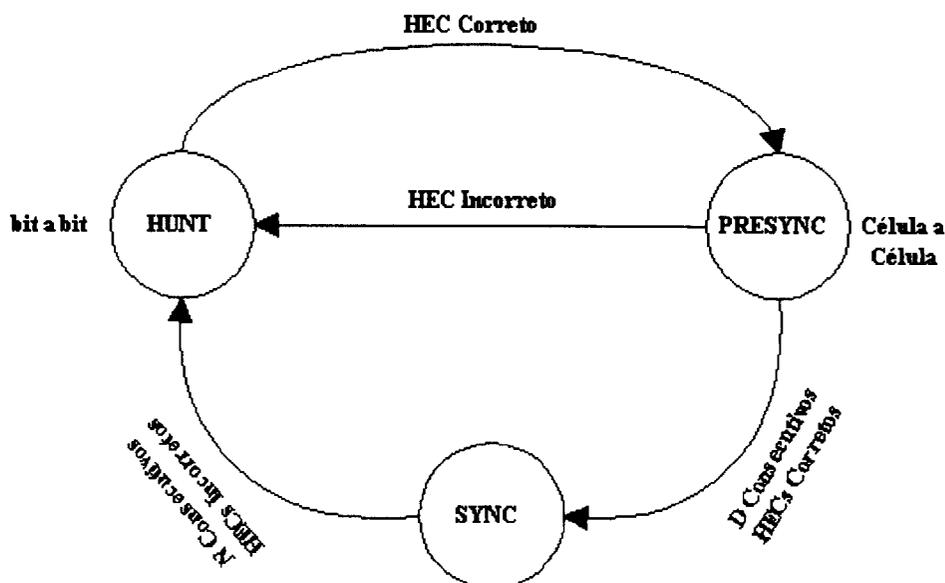


Figura 3. 11: Diagrama de estados do bloco Delineador de Células

O procedimento para delimitação de células inicia-se no estado de HUNT num *Sliding Window* de 5 bytes.

Quando uma célula é recebida, o byte “01010101” é subtraído do campo HEC. O byte HEC resultante, junto com os outros 4 bytes do *header*, formam um polinômio de grau 39. Este polinômio é dividido pelo polinômio gerador para se obter os 8 bits restantes, conhecido como *syndrome*<sup>14</sup>.

Um *syndrome* igual a zero, é um forte indicador de que os limites da célula (fronteiras) foram encontradas e que nem um erro no *header* foi encontrado de acordo com Obara (1990), Balboni et al. (1992) e Chen; Liu (1994).

Um *syndrome* diferente de zero pode significar:

- As fronteiras das células estão fora de alinhamento;
- O *header* da célula contém erros;
- Ambos.

<sup>14</sup> **Syndrome:** sempre que calculado o *syndrome* é verificado com 0, sendo o resultado desta verificação um indicativo de que um bom *header* pode ter sido encontrado.

O estado de HUNT calcula o *syndrome* a cada 5 bytes e faz a verificação com zero. Quando obtém êxito, considera que um bom *header* foi encontrado e passa para o estado PRESYNC.

No estado PRESYNC, o processo de delimitação das células entra na fase de confirmação. Se nas próximas D células consecutivas todos os *syndromes* forem iguais a zero, a delimitação é considerada como estabelecida e o processo entra no estado de SYNC; Caso contrário, o processo retorna para o estado de HUNT. Durante os estados de HUNT e PRESYNC as células podem ser descartadas.

No estado SYNC, três tarefas são realizadas:

- Identificação de células com erros no *header* não detectados são marcadas para análise de erros ou descarte;
- Declaração da perda dos limites da célula e retorna para o estado de HUNT quando N consecutivas células com “*syndrome*” diferente de zero forem recebidas. Neste caso apenas a primeira das N células é corrigida (se possível), e as seguintes N – 1 células são descartadas;
- Todas as células recebidas com *syndrome* igual a zero, são passadas para remontagem de células e todas as células vazias ou não associadas são descartadas.

### 3.6.3 Bloco UPC/NPC

O bloco UPC/NPC é responsável por realizar o controle dos parâmetros do usuário e da rede, para UNI e NNI respectivamente. Ele monitora e regula o tráfego através das interfaces de acordo com algum algoritmo de policiamento, como por exemplo o *Leaky Bucket*.

O banco de dados UPC/NPC, recebe informações do CAC (central ou local), a respeito dos VPC(s)/VCC(s) ativos e seus parâmetros de tráfego.

O bloco funcional *Header Content Check* examina todos os *headers* para validade do campo de código. Células com conteúdo inválido são extraídas da fila de células, contadas e registradas para análise.

Como o UPC/NPC é definido para VPC(s)/VCC(s) individuais, cada VPC ou VCC ativo possui um mecanismo UPC/NPC correspondente.

A figura 3.12 representa o diagrama funcional do bloco UPC/NPC. O bloco de Configuração do Tráfego é opcional e sua função, quando presente, é suavizar o fluxo de células da chave, reduzindo a possibilidade de congestionamento causado por um possível surto na entrada. As células suavizadas são enviadas para o bloco de processamento de células.

O bloco Contador de Erros no *Header*, recebe células com erros no *header* detectadas pelo bloco de verificação de erros no *header*, conta o número de células por interface e armazena o *header* destas células com erros para uma análise futura.

Do mesmo modo, células que não estão em conformidade e que são descartadas pelo mecanismo UPC/NPC, são contadas e registradas para análise futuras.

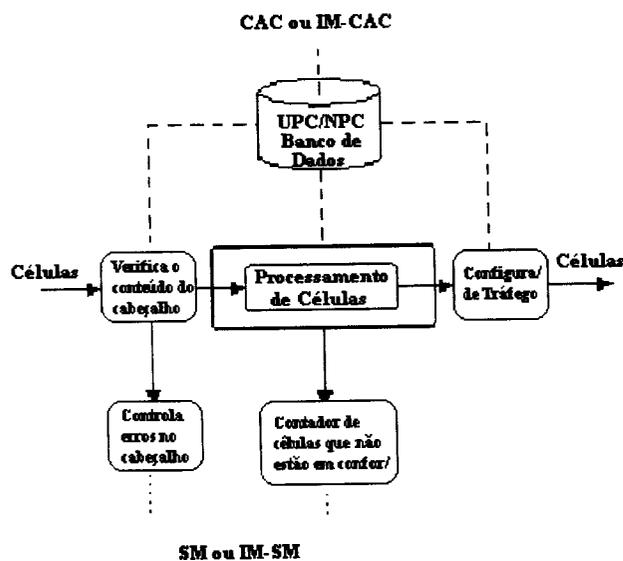


Figura 3. 12: Distribuição das funções do módulo de entrada

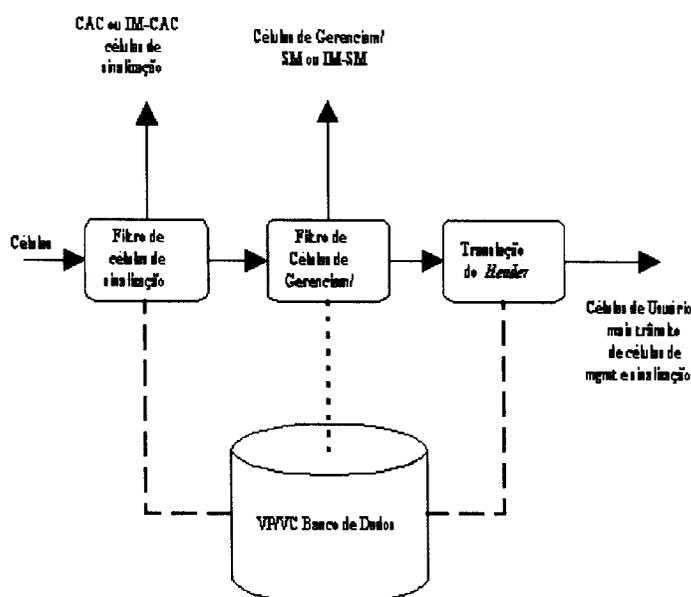
### 3.6.4 Processamento de Células

É o módulo onde as células do usuário, células de gerenciamento e células de sinalização são diferenciadas na chave. Após o processamento das

informações do protocolo ATM no *header* das células, várias ações são tomadas, dependendo do tipo da célula. As células de usuários são preparadas para serem roteadas através da estrutura de roteamento da chave, pela translação do VPI/VCI e a adição de um rótulo interno. Células de sinalização e gerenciamento podem ser extraídas e desviadas para o CAC ou IM-CAC e SM ou IM-SM respectivamente, para processamento, caso necessário. A figura 3.13 representa o diagrama esquemático do bloco funcional de Processamento de Células segundo Obara (1990), Balboni et al. (1992) e Chen, Liu (1994).

O processamento de células é composto por quatro blocos:

- Banco de dados VP/VC;
- Filtro de células de sinalização;
- Filtro de células de gerenciamento;
- Translação do *header* da célula.



**Figura 3. 13:** Diagrama esquemático do bloco de processamento de células

O banco de dados VP/VC é essencialmente uma tabela de consulta com informações a respeito de novos valores VPI/VCI, e porta de saída para cada VPC/VCC ativo. Inclui também informações a respeito dos VPI(s)/VCI(s) pré associados e VPI(s)/VCI(s) associados pelo CAC ou gerenciamento da rede.

Diferente do filtro de células de sinalização, as células do usuário não passam transparentemente pelo filtro de células de gerenciamento. O filtro de células de gerenciamento, pode necessitar monitorar as células de dados do usuário.

O bloco de translação do *header* extrai os novos valores de VPI/VCI do banco de dados VP/VC bem como as outras informações a respeito de cada célula. Os valores de VPI/VCI do *header* das células são transladados para novos valores. Outras informações a respeito de cada célula podem ser encontradas em Obara (1990), Balboni et al. (1992) e Chen, Liu (1994):

- . Usuário fonte;
- . Usuário destino;
- . Porta de saída;
- . Tipo de célula;
- . Atraso de células e tolerância a perdas;
- . Prioridade de atraso;
- . Identificador de conexão *Broadcast/Multicast*.

Estas informações podem ser incluídas em um rótulo interno que é preso a cada célula. As informações de roteamento especificam como as células serão tratadas dentro da estrutura de roteamento da chave. Informações locais como por exemplo usuário fonte ou usuário destino são úteis para monitorar o desempenho interno e também para propósito de contabilidade. O bloco de translação do *header* pode gerar informações internas adicionais como: número de seqüência de células e verificação de erro para o *payload* da célula. Como este rótulo possui significado local, seu contudo irá depender apenas do projeto da chave.

### 3.6.5 O IM-CAC

Se o CAC for centralizado na chave, o módulo de entrada deve ser capaz de reconhecer as células de sinalização e roteá-las diretamente para o CAC ou através de uma porta especial na Estrutura de Célula da Chave (CSF). O problema é que o CAC centralizado pode rapidamente se tornar um ponto de estrangulamento,

sendo conveniente distribuir parte de suas funções para o módulo de entrada IM-CAC.

O CAC realiza as funções da SAAL sobre as células de sinalização da camada superior e toma as decisões de alocação de recursos.

### 3.6.6 IM-SM

Se o SM for centralizado na chave, o módulo de entrada deve ser capaz de reconhecer as células de gerenciamento e roteá-las diretamente para o SM ou através de uma porta especial na Estrutura de Célula da Chave (CSF). Assim como o CAC, o SM centralizado pode rapidamente se tornar um ponto de estrangulamento, sendo conveniente distribuir parte de suas funções para o módulo de entrada IM-SM.

O IM-SM é responsável por todas as funções de controle e gerenciamento da chave para assegurar uma operação correta e eficiente conforme Obara (1990), Balboni et al. (1992) e Chen, Liu (1994). São atributos do IM-SM:

- O UPC/NPC marca o ingresso das células para suportar o gerenciamento da contabilidade;
- O filtro de células OAM suporta as funções da camada ATM OAM. Ele monitora o desempenho, fazendo a contagem de células e a verificação de erros nos blocos de células dos usuários;
- A monitoração do protocolo ATM acontece quando o módulo de entrada processa o *header* da célula.;
- Manipular as funções da camada física OAM, quando o *overhead* SONET é processado.

### 3.7 Módulo de Saída

A primeira responsabilidade do módulo de saída é receber as células da estrutura de roteamento e prepará-las para transmissão no *link* de saída. Muitas das funções da camada física e da camada ATM do módulo de entrada IM, devem ser realizadas no sentido inverso no módulo de saída OM. Dependendo do projeto da

chave pode ser interessante que algumas funções do CAC e SM sejam distribuídas no módulo de saída conforme Obara (1990), Balboni et al. (1992) e Chen; Liu (1994).

O módulo de saída consiste de 5 blocos funcionais chamados:

- Processamento de Células;
- Convergência da Transmissão;
- Funções SONET;
- OM-CAC (opcional);
- OM-SM (opcional).

As funções da camada ATM, são suportadas pelo bloco de Processamento de Células. Os blocos Convergência da Transmissão e SONET realizam funções da camada física. A figura 3.14 apresenta o diagrama de blocos do módulo de saída de uma chave ATM.

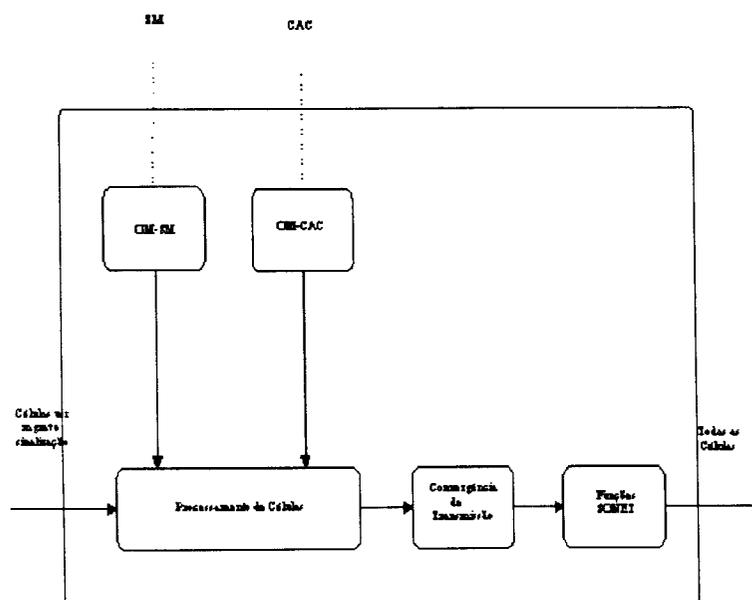


Figura 3. 14: Diagrama de blocos do Módulo de Saída de uma chave ATM

### 3.7.1 Análise do bloco de Processamento de Células

Como visto no item 3.6.4 o bloco de Processamento de Células, é o encarregado por realizar todas as funções de processamento relativas a camada

ATM, necessárias para preparar as células para a transmissão pela camada física. A figura 3.15 apresenta em diagrama de blocos o bloco funcional de Processamento de Células segundo Obara (1990), Balboni et al. (1992) e Chen, Liu (1994).

Ele é composto de quatro componentes:

- Banco de dados VP/VC;
- Inserção de células de sinalização;
- Inserção de células de gerenciamento;
- Processamento e Manutenção.

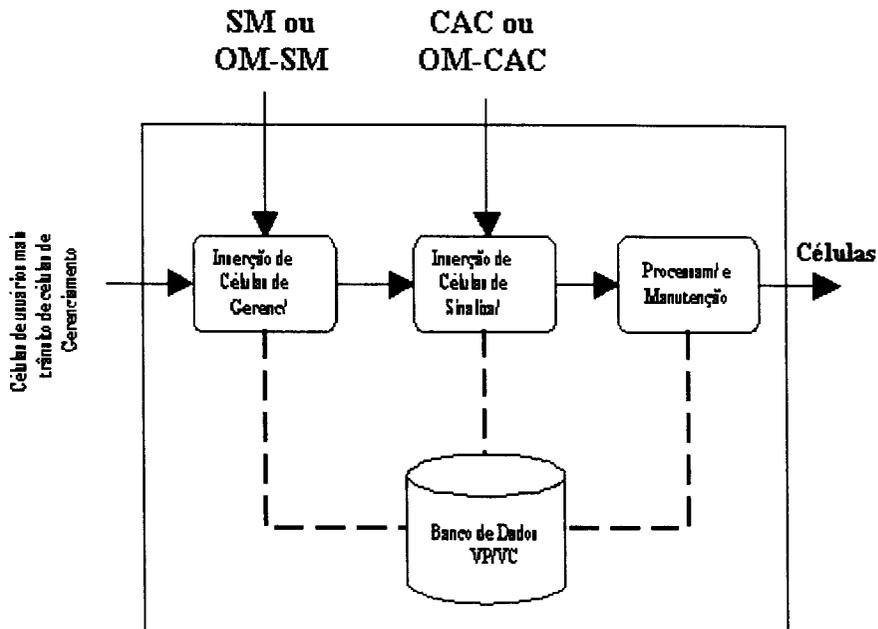


Figura 3.15: Diagrama esquemático do bloco funcional de Processamento de Células

### 3.7.1.1 Banco de Dados VP/VC

Mantém informações a respeito das conexões VP/VC, e, principalmente, os novos valores de VPI/VCI para as células *Multicast*.

O banco de dados VP/VC pode conter informações a respeito da taxa de tráfego de cada conexão, que pode ser utilizada para configuração da fila de saída

das células. Finalmente, o banco de dados pode registrar informações a respeito do tráfego de células na porta de saída, como a contagem de células ou utilização do *link* pelo sistema de gerenciamento

#### **3.7.1.2 Inserção de Células de Gerenciamento**

O bloco de Inserção de Células de Gerenciamento realiza a função inversa do filtro de células de gerenciamento no IM.

As células roteadas para o SM são processadas e o SM pode gerar novas células de gerenciamento e enviá-las para o módulo de saída. É também o bloco que realiza a mistura de células de gerenciamento com a fila de células de dados dos usuários no módulo de saída da chave.

#### **3.7.1.3 Inserção de Células de Sinalização**

O bloco de Inserção de Células de Sinalização realiza a função inversa do filtro de células de sinalização no IM. O CAC interpreta as mensagens de sinalização das camadas superiores transportadas dentro das células de sinalização. Seguindo o protocolo de sinalização, o CAC pode gerar novas células de sinalização, que serão enviadas para o módulo de saída. É também o bloco que realiza a mistura das células de sinalização com as células de dados dos usuários na porta de saída da chave.

#### **3.7.1.4 Processamento e Manutenção**

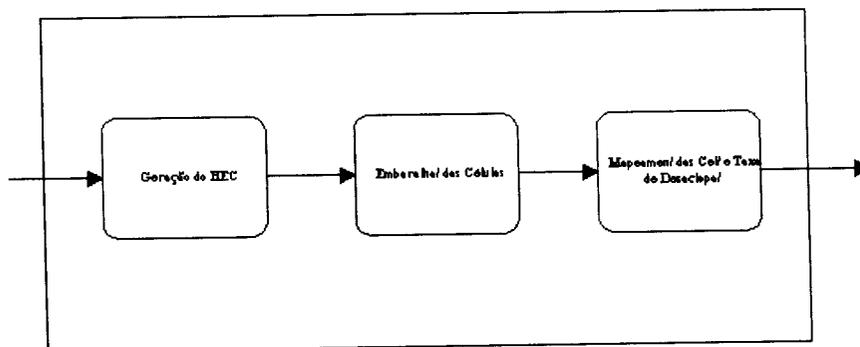
O bloco de Processamento e Manutenção realiza todas as funções da camada ATM antes das células estarem prontas para transmissão. Primeiro, usando o banco de dados VP/VC, os novos valores de VPI/VCI para as células *Multicast* são determinados e inseridos no *header* da célula. Todas as células de saída podem ter novos valores de VPI/VCI a partir deste ponto.

Outra tarefa importante realizada por este bloco funcional é a remoção e o processamento do rótulo interno fixado em cada célula pelo módulo de entrada.

As informações de manutenção contidas no rótulo são processadas no módulo de saída, para a monitoração do desempenho interno da chave.

### 3.7.2 Convergência da Transmissão

O bloco de Convergência da Transmissão, manipula as funções da camada física que são independentes do meio de transmissão. É composto por três componentes que representam os passos para preparação de uma célula para a transmissão SONET. A figura 3.16 apresenta um diagrama de blocos do bloco funcional de Convergência da Transmissão segundo Obara (1990), Balboni et al. (1992) e Chen, Liu (1994).



**Figura 3. 16:** Diagrama esquemático do bloco funcional de convergência da transmissão

Os dois primeiros passos dados pelo bloco de Convergência da Transmissão são a geração do HEC usando os quatro primeiros bytes do *header* da célula ATM e o embaralhamento das células. O novo campo HEC calculado, é inserido no quinto byte do *header* da célula. O embaralhamento da célula é feito no campo de informação da célula. Como estes dois campos atuam em partes distintas da célula, é possível a inversão de seqüência destes blocos sem qualquer prejuízo para o desempenho da chave.

O terceiro passo é o mapeamento e a taxa de desacoplamento do pacote de células ATM sobre o *payload* SONET. Para preencher os espaços entre as células, células ATM não associadas, como células contendo VPI = 0, VCI = 0, são

geradas e misturadas na saída com as células associadas, para garantir um fluxo contínuo de células necessário para o preenchimento do *payload* SONET.

### 3.7.3 Funções SONET

O bloco funcional SONET, é responsável pela transmissão do *frame* SONET sobre *links* de fibra ótica. É constituído por quatro blocos funcionais que estão esquematizados na figura 3.17 conforme Obara (1990), Balboni et al. (1992) e Chen, Liu (1994):

- Geração do *Frame* SONET;
- Geração do *Overhead* SONET;
- Embaralhamento do *Payload*;
- Conversão do sinal elétrico para ótico.

O módulo de saída será responsável pela geração e transmissão de sinais de alarme, através do *Data Communication Channel* (DCC), realização de testes e monitoração do equipamento.

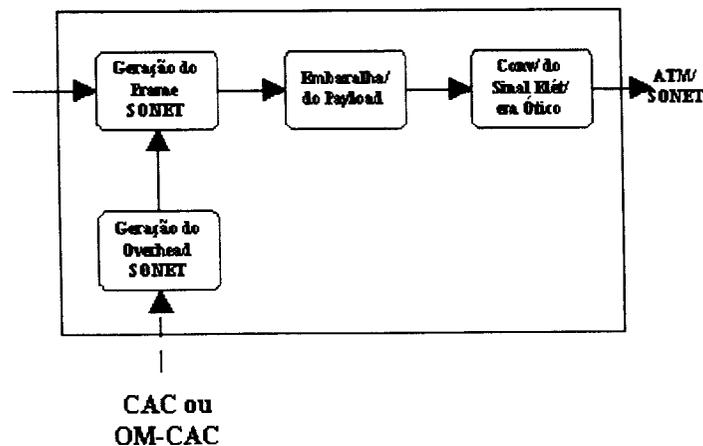


Figura 3. 17: Diagrama esquemático do bloco funcional SONET

### 3.7.4 OM-CAC e OM-SM

Dependendo do projeto da chave, pode ser interessante a distribuição de algumas funções do CAC e SM no módulo de saída. Por exemplo o OM-CAC poderia manipular as funções da SAAL para encapsular as mensagens de sinalização das camadas superiores em células de sinalização. O OM-SM poderia manipular a geração de novas células OAM.

A distribuição exata das funções do CAC e SM no módulo de saída, pode variar de chave para chave, sendo moldada conforme as necessidades do projeto.

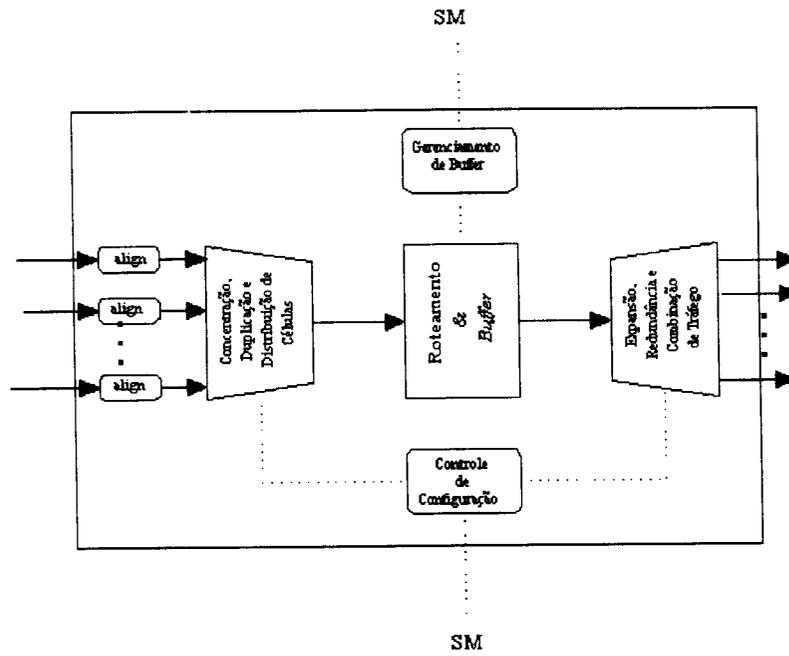
### 3.8 Estrutura de Célula da Chave – CSF

É a parte central de uma chave ATM, e é a responsável pela transferência de células entre os outros blocos funcionais da chave. Sua principal função é a transferência de células do usuário da porta de entrada para a porta de saída destino. É possível utilizar a CSF para rotear células de sinalização e gerenciamento para o CAC e SM respectivamente, através de portas especiais.

A complexidade da CSF aumenta quando os requisitos segundo Pattavina; Bruzzi (1993), Chen, Liu (1994), Gaïti, Pujolle (1996), Lin, Hughes (1997), são:

- *Multicasting* (assunto dos próximos capítulos);
- Tolerância a falhas;
- Prioridade de perdas e atrasos.
- Quando estes requisitos são considerados funções adicionais são incorporadas pela Estrutura de Célula da Chave CSF como:
  - Concentração de células (*buffer*);
  - Duplicação de células;
  - Escalonamento de células;
  - Descarte seletivo de células;
  - E monitoração de congestionamento.

A figura 3.18 apresenta uma representação esquemática dos blocos funcionais de uma Estrutura de Célula de Chave típica.



**Figura 3. 18:** Diagrama funcional das funções da Estrutura de Célula da Chave

### 3.8.1 Roteamento & Buffer

O módulo de Roteamento & Buffer, tem por função rotear as células que estão chegando dos módulos de entrada, para os módulos de saída apropriados, de acordo com o rótulo de roteamento interno fixado em cada célula no módulo de entrada.

Como muitas células podem estar endereçadas para a mesma porta de saída, situações de disputa podem ocorrer; daí a necessidade dos buffers.

Um dos primeiros protótipos de chaves foi o *Starlite* de Huang; Knauer (1984), que utilizava uma rede *Batcher Banyan* como estrutura de roteamento e empregava uma tecnologia de *buffer* circular. Apesar de interessante este protótipo esbarrou na dificuldade de se implementar na prática este tipo de *buffer*.

Na tentativa de minimizar os problemas de se implementar o *buffer* circular, foi proposta a utilização de um *buffer* central compartilhado, que acabou por resultar na redução da quantidade total de memória necessária. Atualmente expandiu-se este conceito e alguns fabricantes como a *Fore*, empregam o conceito de *buffer* particionado distribuído, aumentando consideravelmente o desempenho da chave [*ForeRunner ATM Switch Architecture* (1997)].

Devido a grande importância da Estrutura de Células da Chave, os tópicos a seguir descrevem dois projetos clássicos para a CSF, classificados como:

- Totalmente interconectada;
- Divisão de espaço.

### 3.8.1.1 Totalmente interconectada

A figura 3.19 mostra um diagrama de blocos de uma CSF com abordagem totalmente interconectada. Neste tipo de projeto existem caminhos independentes entre todas  $N$  portas de entrada com as  $N$  portas de saída.

Neste tipo de abordagem, todas as células que chegam em *Broadcast* em barramentos separados com todas as portas de saídas. Os filtros de endereço em cada porta de saída são necessários para determinar se as células são, de fato, destinadas àquela porta de saída. Após passarem pelo filtro, as células são armazenadas no *buffer* de saída.

Os filtros de endereço e os *buffers* são simples de serem implementados, podendo operar à mesma velocidade da porta de saída segundo Nojima et al. (1987). Mais que isto, todo hardware opera na mesma velocidade.

Infelizmente o crescimento  $N^2$  do *buffer* deve ser limitado por razões práticas. A chave *Knockout* emprega este tipo de abordagem, acrescentando um mecanismo de concentração de células, que permite a redução dos  $N$  *buffers* por porta para  $L$ , onde o valor de  $L$ , estatisticamente calculado para garantir uma taxa de perdas de células inferior a  $10^{-10}$ , para qualquer valor de  $N$ , é 12 segundo Pattavina, Bruzzi (1993) e Suh, Jun (1994).

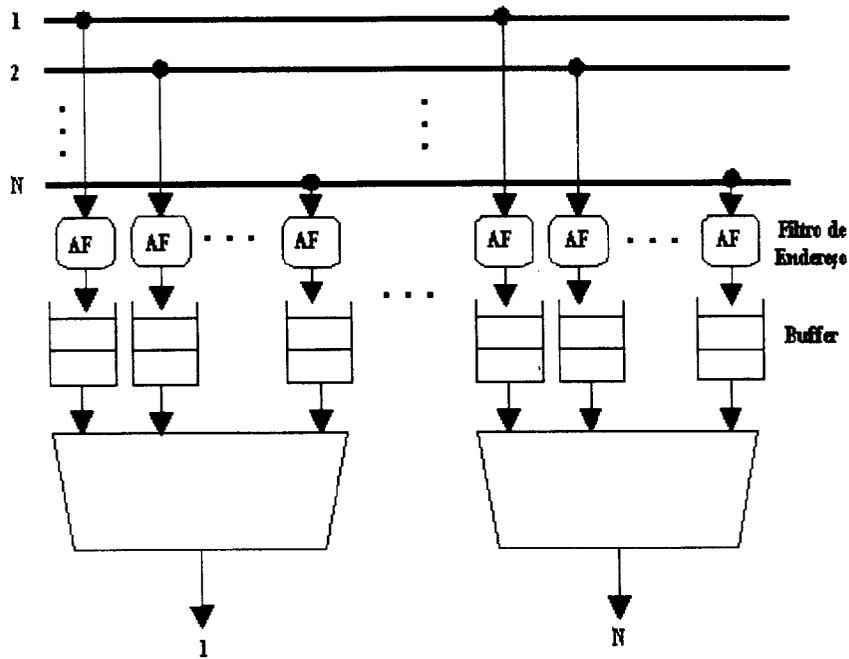


Figura 3. 19: Diagrama esquemático da Abordagem Totalmente Interconectada

### 3.8.1.2 Divisão de Espaço

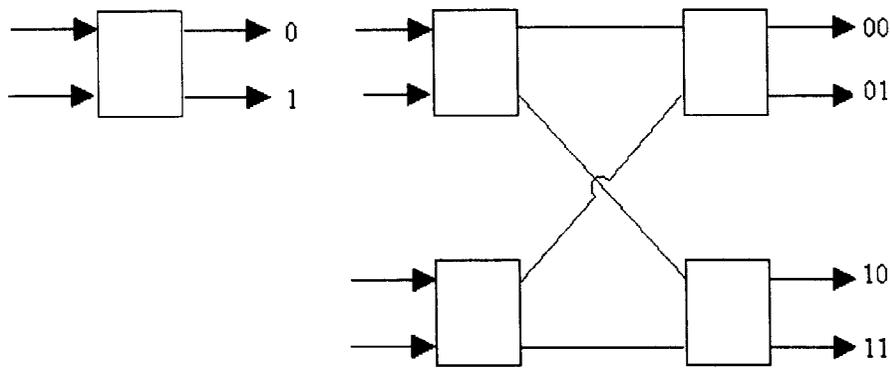
A abordagem por Divisão de Espaço pode ser encontrada nas chaves que utilizam o conceito de barramento cruzado (*crossbar*), que conectam fisicamente qualquer das  $N$  entradas com qualquer das  $N$  saídas.

Dentro desta abordagem, tem-se dado uma ênfase muito grande no estudo das *Multistage Interconnection Network* (MINs), como uma alternativa para as chaves com barramento cruzado utilizadas na telefonia Chen; Liu (1995).

A figura 3.20 mostra uma Rede Banyan, construída recursivamente a partir de elementos de chaveamento  $2 \times 2$ , freqüentemente utilizada como uma MIN em chaves ATM conforme Chen, Liu (1995):

Nas redes Banyan, cada elemento  $2 \times 2$  roteia as células de acordo com o bit de entrada (bits de endereço acrescentado às células no IM). Se o bit for

zero, independente da entrada, a célula será roteada para a saída superior (endereço 0); caso contrário a célula será roteada para a saída inferior (endereço 1).



**Figura 3. 20:** Diagrama esquemático de uma Rede Banyan

As redes *Banyan* se mostraram muito atrativas em função de razões como as enunciadas por Chen; Liu (1995) e descritas a seguir:

- O roteamento é executado por pequenos elementos de chaveamento;
- As células são roteadas em paralelo;
- Todos os elementos operam na mesma velocidade;
- Uma grande estrutura de roteamento pode ser construída de forma modular e recursiva;
- Pode facilmente ser implementada em hardware.

Como as redes Banyan possuem menos de  $N^2$  pontos de cruzamento, pode acontecer que duas células com destinos diferentes venham a disputar um mesmo canal antes do último estágio da chave (esta situação é chamada de bloqueio interno da chave). Neste caso, apenas uma célula irá passar para o próximo estágio.

Uma solução para este problema foi adicionar uma rede de classificação *Batcher*, que rearranja as células por ordem crescente de seu endereço antes da rede Banyan, formando assim a combinação *Batcher Banyan*. A rede *Batcher* tem sido utilizada até os dias de hoje com uma rede de classificação. Esta

abordagem resolve o problema da disputa interna sendo portanto, uma estrutura internamente não bloqueada. No entanto, ainda pode ocorrer contenção na saída, caso duas células sejam endereçadas para a mesma porta de saída. Para solucionar este problema, é necessário adotar-se alguma estratégia de *buffer*.

Uma boa solução é a utilização de um *buffer* circular, externo à estrutura, como visto anteriormente. Esse tipo de *buffer* requer um complexo mecanismo de gerenciamento e uma complexa implementação em *hardware*.

A utilização de *buffers* na saída, além de simplificar o projeto da estrutura, tem-se mostrado muito eficiente, e pode facilmente ser implementado sem muita modificação na estrutura. Alternativamente, a utilização de *buffer* na entrada ou internos na estrutura levariam ao *head-of-line-blocking*<sup>15</sup>, reduzindo drasticamente o *throughput* da chave conforme Pattavina, Bruzzi (1993), Meyer et al. (1993), Liew (1994) e Jaccob, Kumar (1996).

A situação de bloqueio interno é causada por uma disputa de um *link* interno da Estrutura de Célula da Chave, por duas células destinadas a portas de saída diferentes. A figura 3.21 mostra uma situação onde duas células endereçadas para as portas de saída (010) e (011), tentam seguir pelo mesmo *link*.

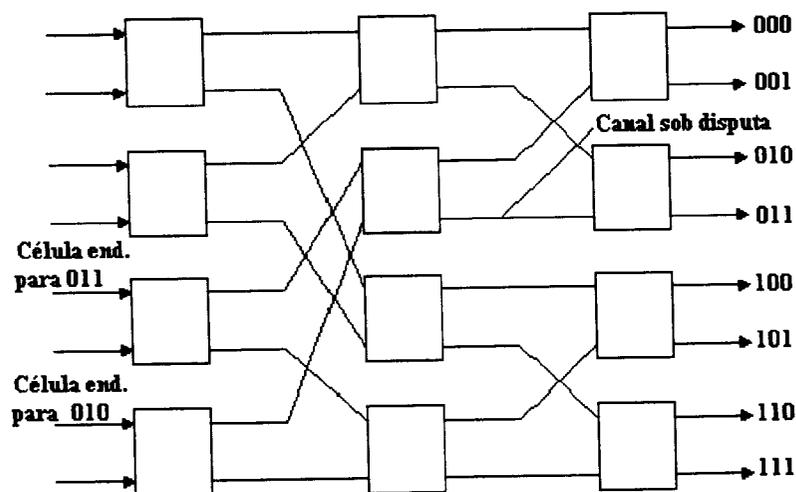
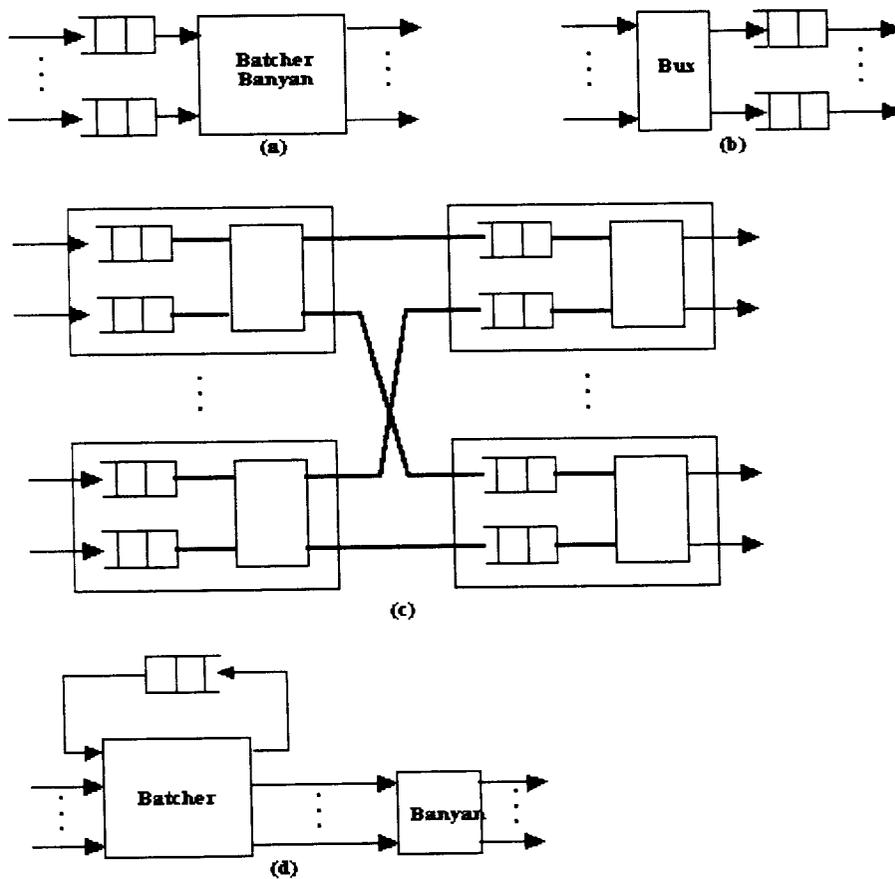


Figura 3. 21: Exemplo de bloqueio interno

<sup>15</sup> **head-of-line-blocking**: situação de bloqueio causada por células enfileiradas que não se destinam a mesma saída e no entanto precisam esperar sua vez.

O problema da contenção de saída é causado por uma situação de disputa, onde 2 células com destino à mesma porta de saída, após terem passado por todos os estágios da MIN com sucesso, chegam juntas à mesma porta de saída. Neste ponto, apenas uma das células será atendida, o que torna evidente a necessidade de *buffer*. A figura 3.22 mostra as quatro estratégias clássicas de *buffer* empregadas em chaves ATM. É claro que uma combinação destes modelos clássicos é perfeitamente possível segundo Meyer et al. (1993), Pattavina, Bruzzi (1993), Suh, Jun (1994), Jacob, Kumar (1996) e Chen, Liu (1995).



**Figura 3. 22:** Exemplos de *buffer*: (a) *buffer de entrada* (b) *buffer de saída* (c) *buffer interno* (d) *buffer circular*

### 3.8.2 Concentração e Expansão

O bloco funcional Concentração e Expansão é classificado como opcional no ambiente da chave, podendo suas funções serem realizadas por um dispositivo externo. A chave ATM em um contexto prático pode ser constituída de múltiplos planos em paralelo para facilitar a distribuição do tráfego, garantir o *throughput*, mesmo em caso de faltas em qualquer dos planos da estrutura de roteamento.

Como uma chave ATM pode estar conectada diretamente aos clientes, é improvável que a maioria das interfaces estejam ativas durante a maior parte do tempo, o que viabiliza um estágio de concentração do tráfego antes do estágio de roteamento da chave conforme Balboni et al. (1992), Pattavina, Bruzzi (1993) e Chen, Liu (1995).

A quantidade de concentração irá depender das características do tráfego que na maioria das vezes é desconhecida. Para Banwell et al. (1991), uma taxa de concentração de 10:1 é um bom valor. Já para Fischer et al. (1989), uma taxa entre 1:1 e 31:1 pode ser facilmente configurada em um concentrador remoto.

Outra função importante do bloco funcional de concentração é a distribuição do tráfego em múltiplos planos de roteamento e estruturas de *buffer*, sendo que esta distribuição deve ser ajustada dinamicamente para garantir o *throughput* da chave.

Por outro lado, a chave pode possuir interface com muitos assinantes, o que pode resultar na necessidade de expandir o tráfego após a estrutura de roteamento e *buffer*, realizando a função inversa do concentrador segundo Balboni et al. (1992), Pattavina, Bruzzi (1993) e Chen; Liu (1995).

### 3.8.3 Cópia e *Multicast*

Serviços como por exemplo vídeo *broadcast*, requer muito mais que a simples transferência de informação de uma porta de entrada para uma porta de saída já que nos serviços de *broadcast*, a informação em uma porta de entrada é direcionada para todas as portas de saída. No *multicast* a informação é direcionada

para um grupo seletivo de portas de saída segundo Huang, Knauer (1984), Pattavina, Bruzzi (1993), Meyer et al. (1993), Chen, Liu (1994).

A necessidade de realizar *multicast* e *broadcast* na chave ATM, resulta em uma série de fatores como aumento da complexidade de controle, aumento de memória e redução do *throughput*. A complexidade deste processo foi o que motivou a realização deste trabalho sendo este o assunto dos próximos capítulos.

#### 3.8.4 Gerenciamento de *Buffer*

A questão de gerenciamento de *buffer* ganha importância e cresce em complexidade quando deixa-se de assumir que o tráfego em uma rede ATM é bem comportado e seus requisitos são os mesmos para todos os usuários segundo Liew (1994), Suh, Jun (1994), Chen, Liu (1994) e Gaïti; Pujolle (1996), e passa-se a levar em conta a variedade de classes QoS que compõem o tráfego ATM, incluindo serviços do tipo de melhor esforço, definido pelo ATM Forum como um serviço a taxa de bits disponível.

A CSF deve manipular estas diferentes classes de serviço de acordo com seus requisitos de QoS. A utilização de *buffer* na saída, embora sendo uma opção bem aceita, não mascara o problema da CSF ser obrigada a manipular múltiplos *buffers* em cada porta de saída e um *buffer* para cada classe QoS. Estes *buffers* podem ser fisicamente separados ou um único *buffer* separado logicamente. Cada *buffer* é tipicamente uma FIFO, mas a disciplina da fila não necessariamente requer que seja uma FIFO segundo Chen, Liu (1995).

O gerenciamento refere-se ao policiamento e ao descarte das células que estão entrando no *buffer* e à política de escalonamento adotada para as células que estão saindo dos *buffers*. O gerenciamento de *buffer* envolve o escalonamento de células, espaço no *buffer*, descarte de células, prioridade de atrasos e indicação de congestionamento conforme Huang, Knauer (1984), Balboni et al. (1992) e Chen, Liu (1995).

### 3.9 Controle e Admissão da Conexão – CAC

Sua função é realizar o controle de tráfego. Consiste de funções de controle para o estabelecimento, supervisão e liberação da conexão virtual (conexões virtuais semi permanentes são controladas pelo gerente da rede). O CAC troca informações de sinalização entre o usuário e o nó da rede.

No B-ISDN, chamadas e conexões podem ser controladas separadamente. Conexões podem ser: ponto-a-ponto ou ponto-para-multiponto, simétrica ou assimétrica; elas são caracterizadas por uma classe QoS específica e um *throughput*. Uma chamada pode consistir de múltiplas conexões, e qualquer conexão pode ser estabelecida, modificada ou finalizada durante a chamada. A figura 3.23 apresenta um diagrama de blocos do bloco funcional de Controle e Admissão da Conexão.

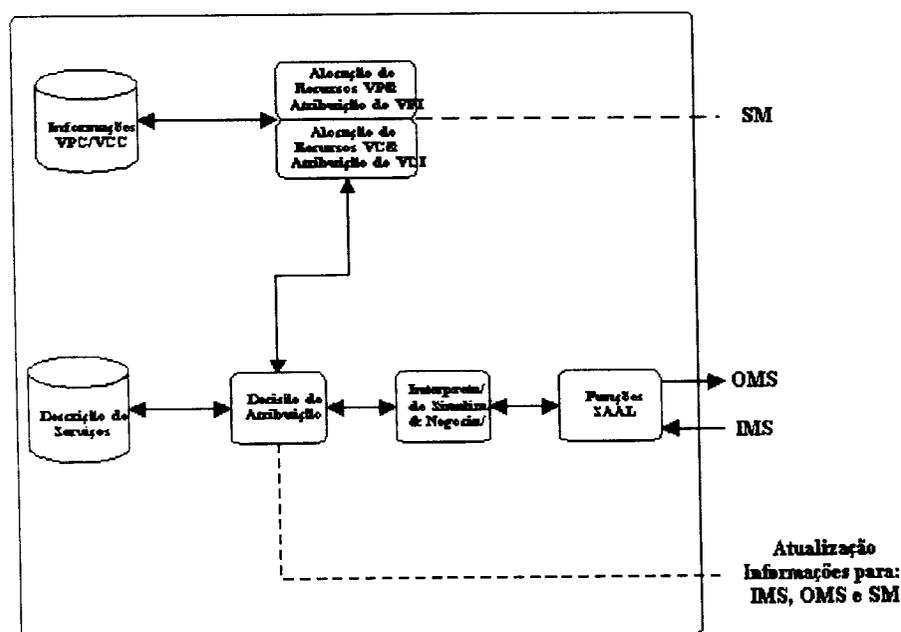


Figura 3. 23: Diagrama esquemático do bloco funcional de Controle e Admissão da Conexão

Quando o CAC recebe células de sinalização do módulo de entrada, as informações de sinalização das camadas superiores são extraídas pelas funções de sinalização AAL (SAAL) e interpretadas por um bloco funcional de negociação que

segue os protocolos de sinalização das camadas superiores (ex.: Q.2931 e BISUP). As funções SAAL também trabalham no sentido reverso, ou seja, encapsulando mensagens de sinalização em células ATM e enviando-as para o módulo de saída.

A decisão a respeito de uma nova conexão segue uma política de admissão de conexão especificada pelo bloco funcional de tomada de decisão.

Como parte do processo de decisão de admissão, os recursos necessários serão encontrados em dois blocos funcionais responsáveis pela alocação de recursos de caminho virtual e canal virtual VPC(s)/VCC(s). Eles mantêm informações a respeito do *status* corrente de todos os VPC(s)/VCC(s) e dos recursos da rede.

Quando um VPC/VCC é estabelecido ou modificado, o banco de dados deve ser atualizado, assim como as informações para os módulos de IM, OM e SM.

### **3.9.2 Alocação de recursos**

O CAC se comunica com usuários e outras chaves para coordenar as ações que permitem a manutenção das conexões e o estabelecimento de novas conexões.

Uma função importante do CAC é a tomada de decisão a respeito da aceitação ou rejeição de uma nova conexão. No B-ISDN, a aceitação ou rejeição de uma conexão é muito complexo, diante da existência de uma grande variedade de características diferentes e requisitos variados de QoS. Ainda hoje os critérios para aceitação ou rejeição de uma conexão não estão totalmente definidos segundo Chen, Liu (1995).

Para não haver imparcialidades, foi proposta a divisão dos recursos de largura de banda em agrupamentos de largura de banda separados, para cada tipo de serviço, e uma nova conexão não pode usar a largura de banda reservada para outro tipo de serviço conforme Cooper, Park (1990).

A decisão de admissão depende do algoritmo de alocação de recursos que estiver sendo utilizado, pois, é ele que determina se os recursos disponíveis da

rede são suficientes ou não. Se os recursos suficientes não foram encontrados, a solicitação da nova conexão é imediatamente rejeitada.

A questão principal neste momento é: como os recursos são julgados como sendo suficientes ou insuficientes para a conexão requisitada? Esta questão embora objeto de intenso estudo, não tem alcançado resultados muito expressivos em termos práticos. De um modo geral, ao menos três pontos são levados em questão quando se trata de estabelecer uma nova conexão de acordo com Hughes, Wajda (1994), Saito et al. (1991) e Chen, Liu (1995): caminho virtual, seleção de rota e estimativa da QoS desejada.

### **3.10 Sistema de Gerenciamento**

É o responsável pelas funções do Plano de Gerenciamento do B-ISDN, no contexto da chave. De um modo geral, o SM é o responsável pela operação correta e eficiente da chave. Dentro da chave ATM, o SM envolve a coordenação de todos os outros blocos funcionais.

O nível das funções de gerenciamento dentro da chave pode variar, de chave para chave, sendo sua distribuição uma função do projeto da chave. Frequentemente o SM incorpora cinco funções: gerenciamento de configuração, gerenciamento de faltas, gerenciamento de desempenho, gerenciamento de segurança e gerenciamento de contabilidade. A figura 3.24 mostra um diagrama de blocos do SM. Essa representação não pode ser considerada como sendo a única, pois sendo o SM uma função do projeto da chave, outras implementações são possíveis.

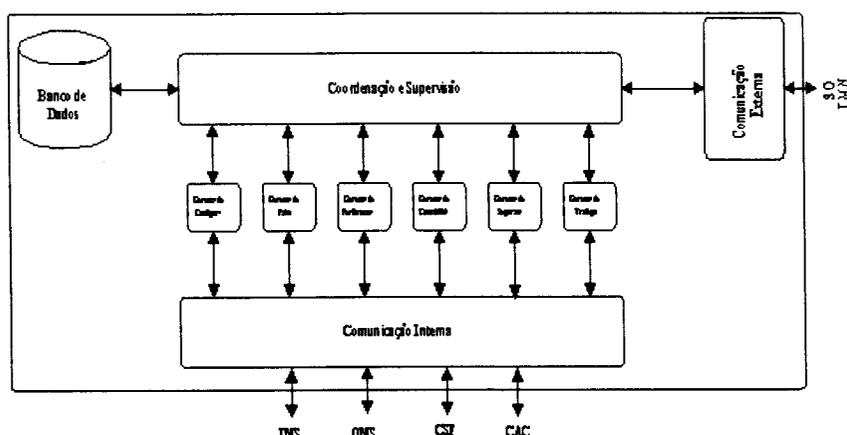


Figura 3. 24: Diagrama funcional do SM

O gerenciamento do tráfego é importante por causa da natureza assíncrona do ATM. O SM coleta informações de gerenciamento necessária para a tomada de decisão. Estas informações são também necessárias para os usuários e gerentes de rede para propósitos de controle e planejamento.

O gerenciamento de faltas é responsável pela detecção, isolamento, teste e recuperação dos problemas quando possível. O gerenciamento de desempenho está preocupado em avaliar e monitorar continuamente o desempenho, preocupando-se com: transmissão de informação pela camada física e processamento dos protocolos da camada ATM e AAL. As preocupações do gerenciamento de configuração são com os componentes da chave que estão ou não em serviço. As informações de configuração podem ser mantidas dentro de um banco de dados no SM.

O gerenciamento da contabilidade é o responsável por coletar informações do usuário e prover suporte para emissão de fatura de clientes, e também prover subsídios para o planejamento da rede. Gerenciamento de segurança é o responsável pela proteção da chave e seu banco de dados. São atributos do gerenciamento de segurança: identificação do usuário, autenticação e controle de sessão. A principal função do gerenciamento de tráfego é garantir a QoS aceita para uma determinada conexão, dentro de certos limites e reduzir a probabilidade de congestionamento conforme ITU-T Rec. I.432 (1993), ITU-T Rec. I.610 (1993), Chen, Liu (1995).

### 3.10.2 Informações de Gerenciamento

A figura 3.24 mostrou uma única base de dados, mas na prática, a informação pode residir em múltiplas localizações físicas.

Toda ou parte da informação pode estar acessível aos usuários ou gerentes da rede. A informação é armazenada em uma *Management Information Base* (MIB). Uma MIB é conceitualmente um depósito de informações de gerenciamento consistindo de uma coleção estruturada de objetos administrados e seus atributos.

O protocolo normalmente utilizado para acessar uma MIB é o SNMP. A figura 3.25 mostra o paradigma SNMP, onde uma *Network Management Station* (NMS) central troca mensagens com agentes residindo nos elementos da rede que são gerenciados. O agente em cada elemento de rede mantém uma MIB, que contém uma descrição dos objetos gerenciados.

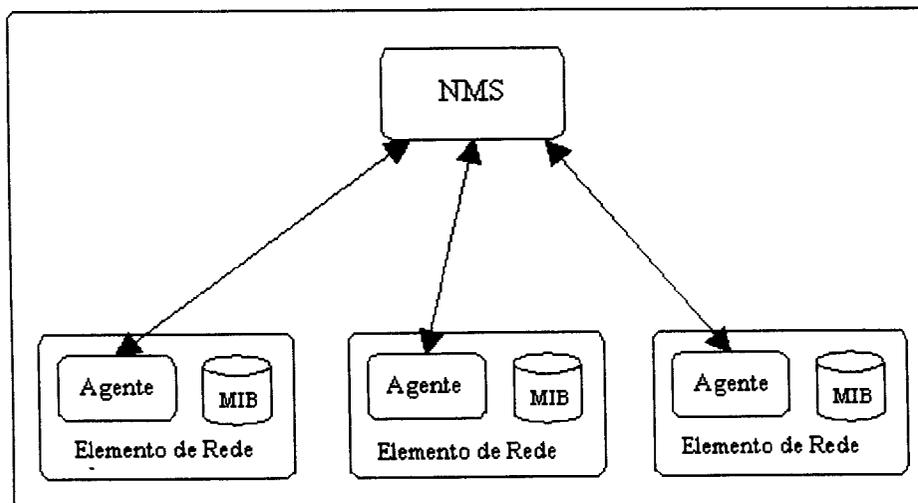


Figura 3. 25: Paradigma SNMP

O SM mantém informações de gerenciamento na MIB para comunicação com usuários ou com o gerente da rede. Nem toda informação na MIB é disponibilizada para os usuários.

O gerenciamento da rede necessita ser capaz de monitorar e controlar um certo número de chaves como um grupo. Isto significa que cada chave deve se comunicar com uma ou mais chaves da rede.

A TMN<sup>16</sup> é uma rede funcionalmente separada da rede, consistindo de uma rede de comunicação de dados SOs, interconectados a rede de telecomunicações em vários pontos. Na verdade a TMN pode compartilhar as instalações de transporte da rede de telecomunicações para transportar informações de administração conforme ITU-T Rec. M.3010 (1992) e Chen, Liu (1995).

Muito do CAC e SM, ainda estão em fase de padronização e recomendação pelo ATM Forum e ITU-T, ficando portanto muito de suas funções a critério do projeto da chave e de sua aplicação conforme Chen, Liu (1995).

---

<sup>16</sup> **TMN: *Telecommunications Management Network* - a separação das chaves (provendo transporte de informação) dos SOs (provendo funções de gerenciamento) através de uma rede de comunicações exemplifica o conceito de TMN.**

## **Capítulo 4**

*Este capítulo traz uma revisão dos principais trabalhos relacionados com o roteamento de células multicast em chaves ATM e também apresenta a arquitetura proposta por este trabalho*

### **Proposta para Simulação**

#### **4.1 Revisão dos trabalhos**

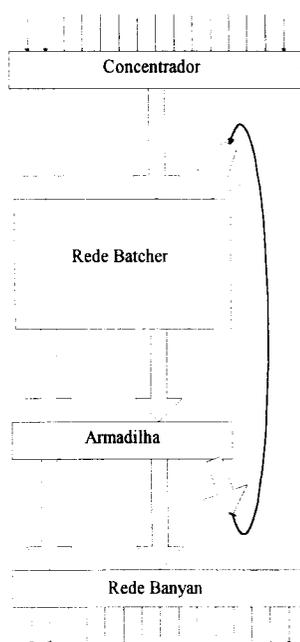
A motivação para este trabalho foi muito bem colocada por Ginsburg (1996) em seu livro *ATM solutions for enterprise internetworking*, “... *Embora uma melhora para o que existe atualmente, o primeiro salto em capacidade entregue pelo ATM não resolve o antigo problema das redes corporativas de escalar dados multicast eficazmente.*” Este capítulo apresenta, em linhas gerais, diversas propostas para tratar o *multicast* em chaves ATM, dando ênfase à estrutura de roteamento, item 3.8 do capítulo anterior e apresenta a proposta para a estrutura de roteamento que será simulada. Os trabalhos foram separados em ordem cronológica e apenas os que foram considerados relevantes na elaboração desta tese serão apresentados.

##### **4.1.1 A arquitetura de chave *Starlite***

Um dos trabalhos mais significativos e marco inicial para as pesquisas foi o *Starlite* de Huang, Knauer (1984), chave digital projetada para atender as necessidades de comunicação de dados, imagem, enfim tudo o que puder ser digitalizado. Com uma arquitetura flexível é a primeira chave descrita na literatura

que emprega o conceito de redes *Batcher-Banyan* como estrutura de roteamento. Foi construída adicionando-se às redes *Batcher-Banyan* um bloco concentrador e um dispositivo de captura (armadilha ou *trap*). Esta estrutura (*Batcher-Banyan*) garante que toda célula chegue a seu destino, desde que não existam múltiplas células destinadas a mesma porta de saída. Cada porta de saída pode no máximo ser endereçada por uma única célula e sua grande vantagem é a modularidade e a facilidade de expansão. A chave manipula células de tamanho fixo, nas quais um rótulo de roteamento é adicionado na entrada da chave.

A função do dispositivo de captura é detectar células com destinos idênticos e as fazer retornar para a entrada da rede *Batcher*, para que possa tentar novamente no próximo ciclo. As células recicladas podem ser recicladas novamente caso a disputa pela mesma porta de saída persista nos próximos ciclos. Para evitar o problema da perda de seqüência, as células que foram recicladas ganham prioridade de atendimento no próximo ciclo. O concentrador tem por função reduzir o tamanho da rede *Batcher* (classificação) baseado na probabilidade reduzida de se ter células ativas em todas as entradas da chave simultaneamente. A figura 4.1 a seguir exhibe, em diagrama de blocos, a organização da estrutura de roteamento da chave *Starlite*.



**Figura 4.1:** Diagrama de blocos da chave *Starlite*

Da forma como foi apresentado na figura 4.1 a chave *Starlite* não pode realizar *multicast*. Para resolver este problema, adicionam-se duas redes conforme a figura 4.2: uma rede de cópia e uma rede com o propósito de ordenar as células que chegam na entrada da estrutura para a cópia, denominada ordene-para-copiar (*sort-to-copy*), neste trabalho o termo em inglês será adotado para referir a rede ordene-para-copiar. A rede *sort-to-copy* recebe em sua entrada, células originais (células fontes), e cópias de células vazias que foram geradas por uma entidade especial. São estas células vazias que serão utilizadas para as cópias das células *multicast*. Na próxima fase, a rede de cópia pega a informação no campo de dados da célula fonte e o copia no campo de dados da célula vazia sendo o roteamento feito utilizando-se o bit de cópia no cabeçalho da célula.

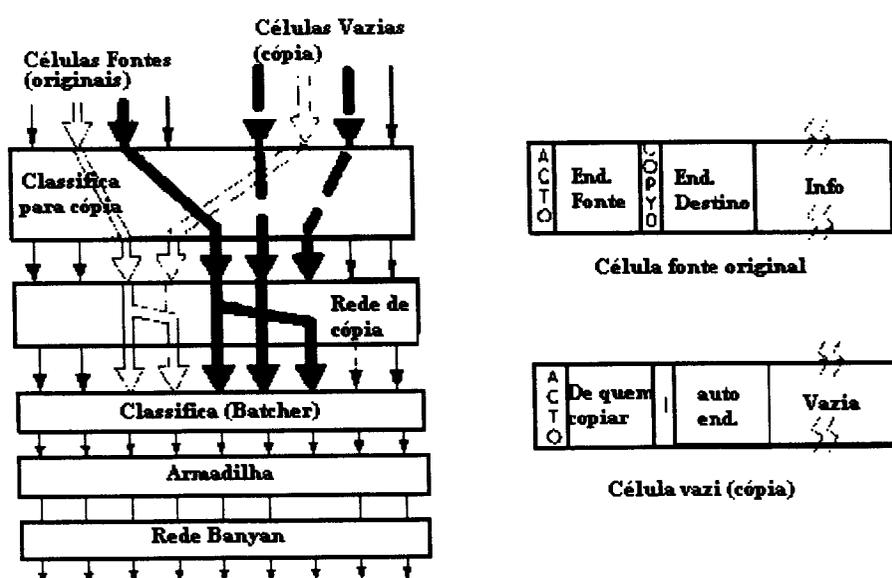
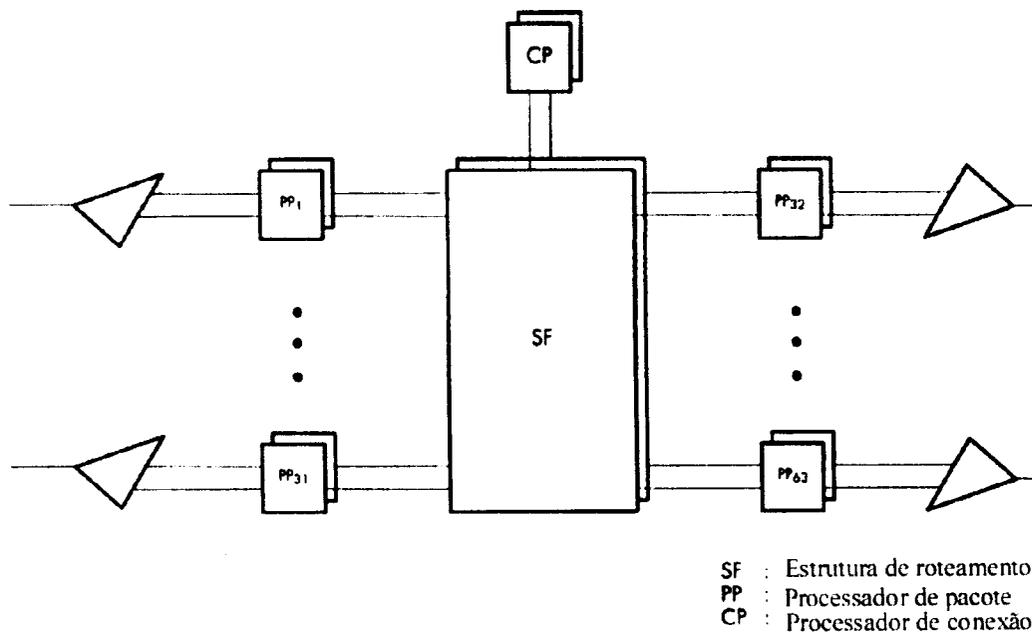


Figura 4.2: *Multicast na chave Starlite*

#### 4.1.2. A arquitetura de chave *St. Louis*

Esta proposta de chave foi feita por Turner (1986) e se mostrou adequada para o chaveamento de pacotes em grandes redes públicas de comunicações, provendo serviços de voz, dados, vídeo e teleconferência. O módulo da chave (SM) conforme a figura 4.3, é composto pelo processador de pacotes (PP) responsável pelo processamento do protocolo (finaliza o protocolo externo e adiciona

o rótulo para o roteamento interno). Em outras palavras, é o módulo que determina como cada pacote será roteado, e a estrutura de roteamento (SF) que é o coração da (SM). A estrutura de roteamento (SF) é organizada em paralelo, provendo múltiplos caminhos para os pacotes e assegurando um bom desempenho e confiabilidade. Se for necessário, pode realizar cópias de pacotes *multicast*, facilitando a realização das funções *multicast*. O processador de conexão (CP) é o responsável pelo estabelecimento da conexão, incluindo conexões *unicast* e *broadcast*.



**Figura 4.3:** Estrutura de roteamento da chave St. Louis

A estrutura de roteamento (SF) está representada em diagrama de blocos na figura 4.4 a seguir, onde se destacam seus quatro componentes principais: a rede de cópia (CN); a rede de distribuição (DN); a rede de roteamento (RN); e um tradutor de grupos *broadcast* (BGT). Quando um pacote *broadcast* com destino a  $k$  saídas diferentes passa através da rede de cópia (CN), são realizadas  $k$  cópias deste pacote. Já um pacote *unicast* passa através da (CN) de forma transparente (sem sofrer qualquer alteração). As redes de Cópia, Roteamento e Distribuição possuem a mesma estrutura interna.

O tradutor de grupo *broadcast* (BGT) realiza duas funções de tradução. Primeiramente, para cada pacote *broadcast* que chega, ele determina o

número do grupo de saída (GN) e o número do *link* (LN). Em seguida ele translada o (GN) para o (LN).

Os BGTs são os responsáveis pelas funções de tradução e preenchimento do rótulo de roteamento, especialmente para as conexões *multicast*. As funções dos BGTs são essenciais, pois cada cópia feita pelo (CN) possui um rótulo de roteamento idêntico.

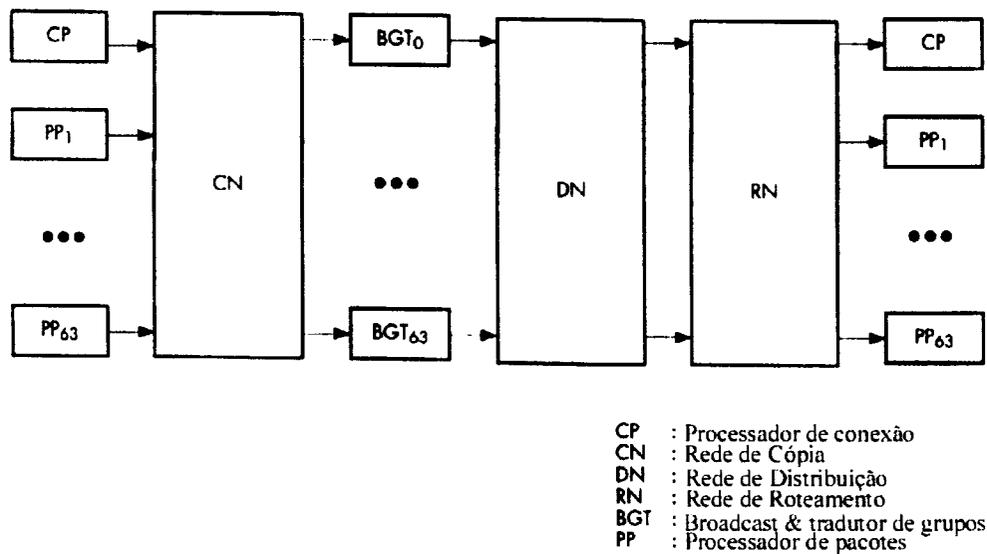


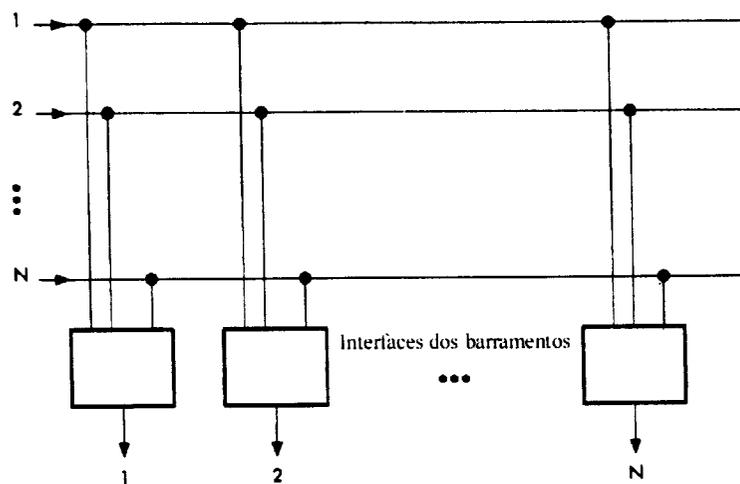
Figura 4.4: Diagrama de blocos da estrutura de roteamento da chave St. Louis

As redes de Distribuição e Roteamento (DN e RN) são as responsáveis pela movimentação dos pacotes para a saída adequada. A rede de Roteamento é na verdade uma rede *Banyan* com *buffer* na entrada de cada nó. A rede de Distribuição possui a mesma estrutura interna da rede de Roteamento, com a ressalva de que seus nós não levam em conta o endereço destino do pacote, sendo estes roteados alternadamente para cada uma de suas portas de saída.

#### 4.1.3. O elemento de roteamento *Knockout*

O elemento de roteamento *Knockout* foi introduzido pela primeira vez por Yeh, Hluchyj (1987). Apresenta facilidade para escalar grandes chaves a partir de pequenos elementos de chaveamento, conservando o mesmo princípio de comutação. Adota-se como estratégia de armazenamento o *buffer* na saída. Um

estágio concentrador é adicionado introduzindo uma certa probabilidade de perda de células, e as  $N$  entradas e  $N$  saídas operam todas à mesma velocidade. O meio de transferência é composto por  $N$  barramentos *broadcast*, um para cada entrada, e cada saída possui acesso a todas as entradas conforme a figura 4.5.



**Figura 4.5:** Estrutura organizacional de uma chave baseada no princípio Knockout

A grande vantagem deste tipo de barramento é a ausência do compartilhamento, permitindo que altas taxas de velocidade sejam alcançadas. No entanto, quando múltiplas células se destinam à mesma porta de saída (no pior caso  $N$  células destinadas a mesma porta de saída), obriga-se a existência de *buffer* em algum ponto da chave. Para garantir uma taxa de perdas igual a zero na fase de transferência de célula para o *buffer*, a memória deve operar (escrever) a uma velocidade  $N$  vezes maior que a velocidade de cada porta de entrada. Para reduzir esta velocidade, a chave *knockout*, acrescenta uma interface para o *buffer*, que atua como um concentrador, o qual resolve o problema da alta velocidade, porém, adicionando uma certa probabilidade de perda de células. A figura 4.6 mostra o estágio concentrador, os filtros de células que examinam o endereço associado a cada saída e o deslocador que distribui as células nos *buffers* de saída.

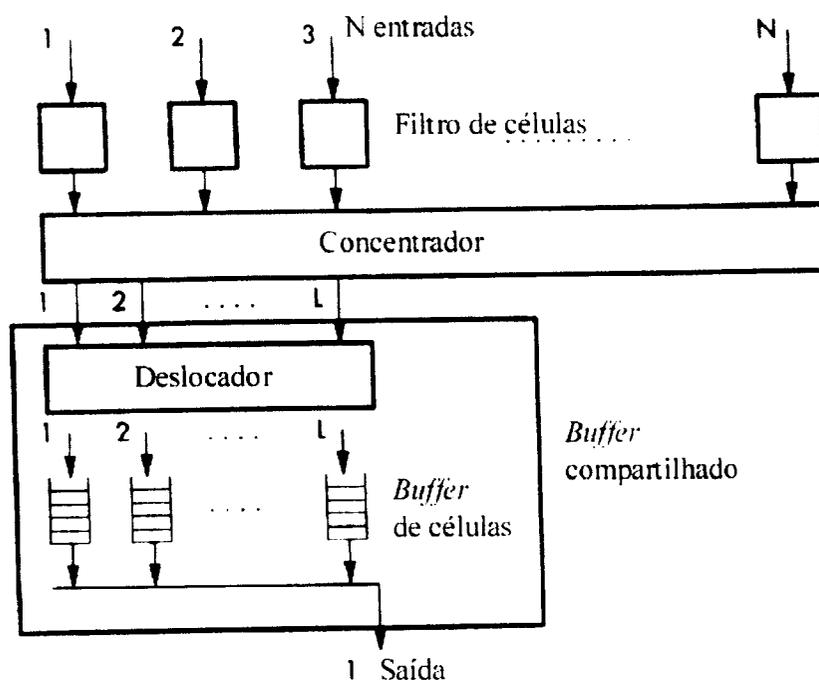


Figura 4.6: Interface do barramento Knockout

Para suportar as funções *multicast*, foi adicionado um módulo para realizar as funções *multicast* no sistema (figura 4.7). Cada módulo *multicast* é conectado a cada um dos N barramentos. As M saídas dos módulos *multicast* são adicionadas à interface de saída gerando um acréscimo de M barramentos, de tal forma, que a interface agora requer  $N + M$  entradas. A vantagem desta interface é que o número de módulos *multicast* pode ser alterado conforme a demanda por pedidos *multicast*. Por outro lado, dimensionar este sistema torna-se muito complexo. Para implementar o módulo *multicast*, um duplicador de células é adicionado, com a finalidade de gerar N cópias de células de cada uma das células que entram na estrutura com diferentes endereços destinos, os quais são gerados utilizando-se uma tabela de *multicast*.

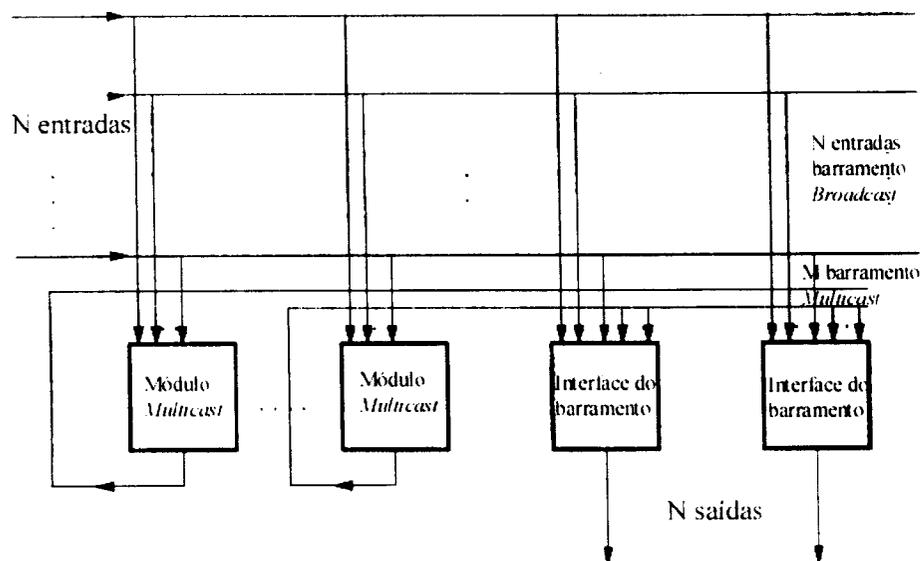


Figura 4.7: Módulos multicast adicionados a chave Knockout

#### 4.1.4 Técnica de agrupamento de canais de saída na estrutura de roteamento *Tandem Banyan Switching Fabric (TBSF)*

A utilização de grupos de saídas e a utilização de redes *Banyan* em cascata não são novidades, e no trabalho de Giacomazzi (1994), elas foram unidas para compor uma estrutura de auto roteamento, que emprega o conceito de fila de saída e a utilização de canais, que possibilitam múltiplas saídas na rede. O agrupamento de canais é adotado como uma estratégia para reduzir a complexidade da implementação da TBSF. A arquitetura é baseada em múltiplos estágios da rede *Banyan* em cascata.

A taxa de perda de células nesta estrutura pode ser controlada, selecionando de forma adequada o número de redes *Banyan* que deverão ser interligadas em cascata, e o tamanho do *buffer* de saída a ser utilizado. Como este número pode tornar-se relativamente grande para grandes chaves, o conceito de grupos de canais torna-se extremamente interessante, permitindo que um pacote

entre na estrutura de roteamento com destino a um grupo de saída e não mais para uma saída específica.

#### 4.1.4.1 A arquitetura TBSF

A estrutura de roteamento *Tandem Banyan Switch Fabric*, inclui 2 subsistemas básicos: a rede de interconexão e o conjunto de filas de saída. A rede de interconexão é um conjunto em cascata de redes *Banyan* conforme a figura 4.8.

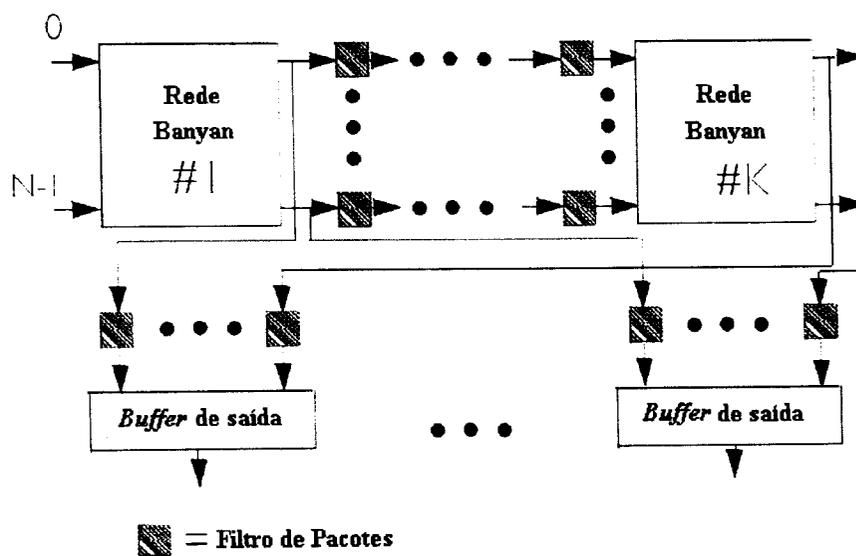
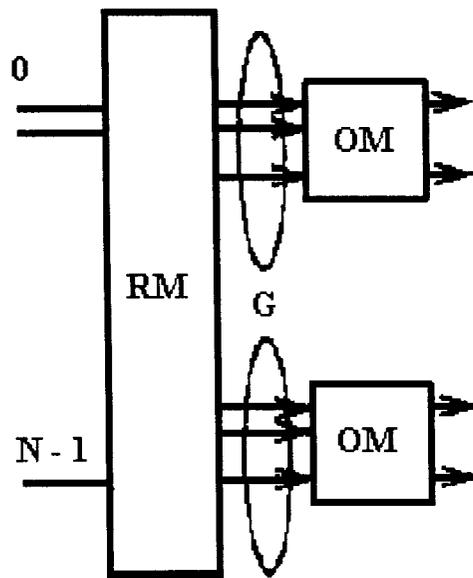


Figura 4.8: Arquitetura da TBSF

Quando ocorrer um conflito na saída de um elemento de roteamento, um pacote será roteado corretamente enquanto o outro será desviado. Os pacotes que alcançaram a saída da rede *Banyan* sem sofrer deflexão, foram corretamente roteados e atingiram seu objetivo. Já os pacotes que sofreram desvios são transmitidos para a próxima rede *Banyan* para uma nova tentativa. Os pacotes que sofrerem deflexão no último estágio da rede *Banyan*, não terão uma nova oportunidade e serão descartados.

#### 4.1.4.2 Algoritmo *bit mask*

Para definir um canal é necessário identificar um grupo de saída. Cada conjunto de saídas que compõem um grupo, formam um módulo de saída (OM), como mostra a figura 4.9. O conceito de grupo e *bit mask* é melhor visualizado na figura 4.10, onde os grupos foram definidos pelas portas de saída (0,1), (2,3), (4,5) e (6,7) (quatro grupos) e o *bit mask* utilizado para controle de transição recebe valor 1 no primeiro e no segundo estágio, permitindo que numa disputa pela mesma porta os pacotes possam ser desviados sem sofrer qualquer prejuízo. No terceiro estágio o *bit mask* recebe valor 0, impedindo que células ao disputarem uma saída possam ser desviadas. Sempre que ocorrer uma situação de disputa em um elemento de roteamento com *bit mask* igual a 0 (terceiro estágio da figura 4.10), nenhum dos pacotes poderão ser desviados. Considerando que os pacotes atingiram o grupo de saída com sucesso, isto não significa que tenham atingido a saída desejada. Apenas quando o *bit mask* estiver em 1 (primeiro e segundo estágio da figura 4.10) é que o pacote em situação de disputa poderá ser desviado.



**Figura 4.9:** Estrutura básica da TBSF

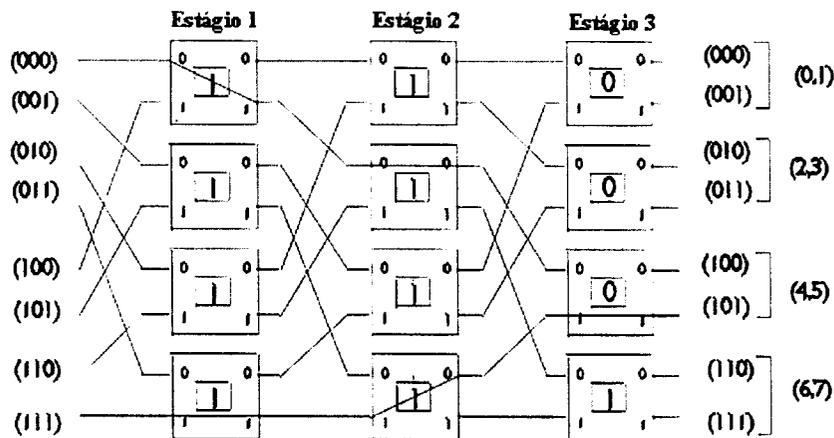


Figura 4.10: Exemplo do algoritmo bit mask

#### 4.1.5 Controle adaptativo de tráfego

Neste trabalho de Byun, Lee (1994), o controle adaptativo de tráfego é adicionado, como uma tentativa de resolver os antigos problemas de cópias de células nas redes de cópia não bloqueadas, propostas até então. A proposta é uma solução completa para os problemas inerentes a realização de *multicast*. O problema causado por *overflow* é resolvido com a CRAN *Cyclic Running Adder Network*, que calcula a soma das cópias de células solicitadas a partir de qualquer porta de entrada e não mais apenas da porta zero como era feito anteriormente. O CRAN também funciona como um controle adaptativo de tráfego. E para garantir uma melhor distribuição das células, é adicionado um estágio deslocador na saída da CRAN, garantindo uma distribuição uniforme das células nos *buffers* de saída. A figura 4.11 mostra uma rede RAN *Running Adder Network*, um conjunto DAE *Dummy Address Encoders*, um concentrador, uma rede BBN *Broadcast Banyan Network* e um grupo de TNT *Trunk Number Translators*.

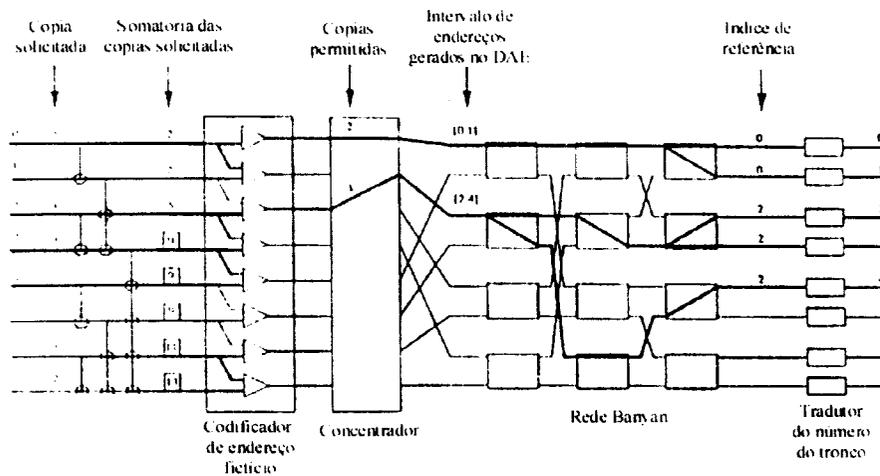


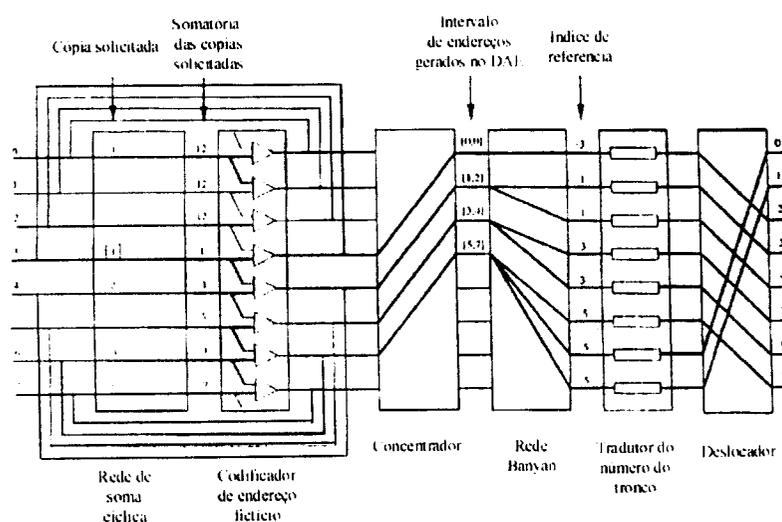
Figura 4.11: Exemplo de uma rede de cópia 8x8 não bloqueada

Na RAN é calculado o número de solicitações *multicast* e o DAE gera um pseudo *header* para cada pacote com solicitação *multicast*. O *header* contém um intervalo de endereço fictício, especificado por um endereço MIN e MAX de portas de saída, e um índice de referência IR. A BBN realiza a replicação dos pacotes com base nos intervalos MIN e MAX do *header*, e o TNT determina um endereço destino para cada pacote, tendo como referência o seu índice de cópia e o número do canal *broadcast*.

Na figura 4.11, um problema inerente a este tipo de configuração é exibido, quando o número de solicitações de cópias excede a capacidade da rede de cópia. O outro problema decorre do fato da soma iniciar sempre pela porta zero: este fato faz com que, na ocorrência de uma seqüência de *overflows*, as portas superiores sejam sempre prejudicadas no seu atendimento.

Para solucionar este problema, a figura 4.12 apresenta a CRAN *Cyclic Running Adder Network*, e adiciona também um deslocador. A estrutura da CRAN permite calcular a soma das cópias a partir de qualquer porta de entrada de forma cíclica. O ponto de partida é determinado pela condição de *overflow* do *slot* de tempo anterior. Esta decisão é tomada na saída da CRAN e retornada para as portas de entrada através de um *loop* de retorno. O deslocador se encarrega de que os pacotes

que foram copiados sejam uniformemente distribuídos nas entradas da rede de roteamento.



**Figura 4.12:** Rede de cópia 8x8 não bloqueada com controle adaptativo. CRAN

O ponto de partida da CRAN permanecerá o mesmo até que um novo *overflow* ocorra. Na saída da CRAN existe um bit de indicação de partida (SI) e os números que são utilizados para cópias (SCN) são determinados por pares adjacentes da RAN.

O SI indica qual a porta de entrada que será o próximo ponto de partida, e o SCN representa o número de cópias permitidos para o pacote correspondente no *slot* de tempo corrente. A figura 4.13 mostra a arquitetura de uma CRAN 8x8 e o formato do *header* associado ao pacote, o qual consiste de 3 campos: indicador de partida (SI), a *running sum* (RS) e a *routing address* (RA). Apenas uma porta do próximo ponto de partida recebe um SI diferente de zero como confirmação. O campo RS contém inicialmente o número de cópias solicitadas. Cada pacote será transportado para a saída após a realização da operação de soma realizada pela CRAN. O campo RA é inicialmente colocado em 1 se o pacote está ativo e 0 se inativo. Na saída do *running adder*, o campo RA será transladado para bit ativo, para que posteriormente possa ser usado como endereço de roteamento pelo concentrador.

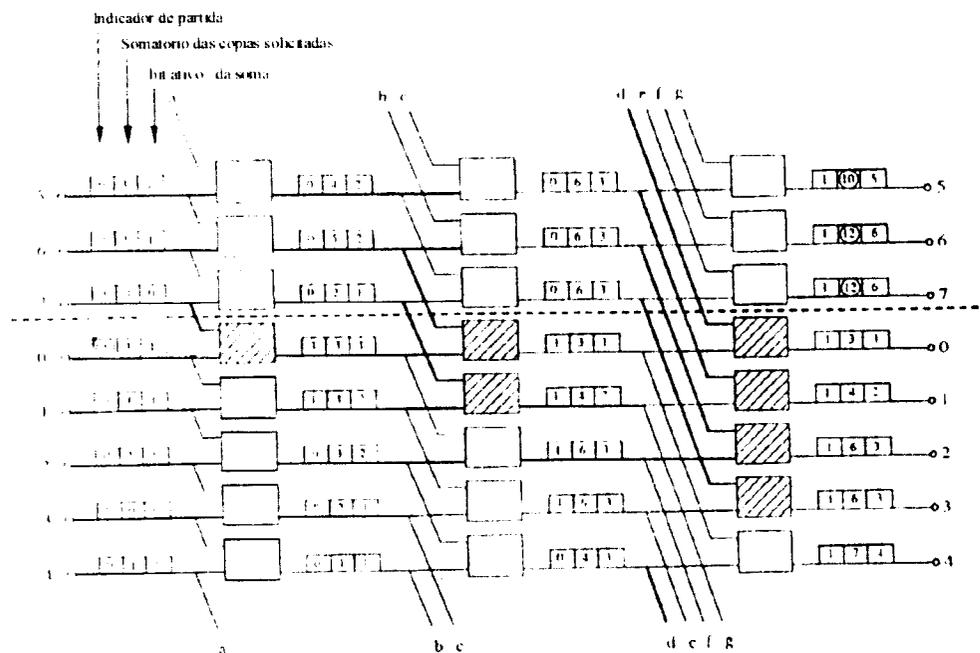


Figura 4.13: Estrutura lógica da CRAN

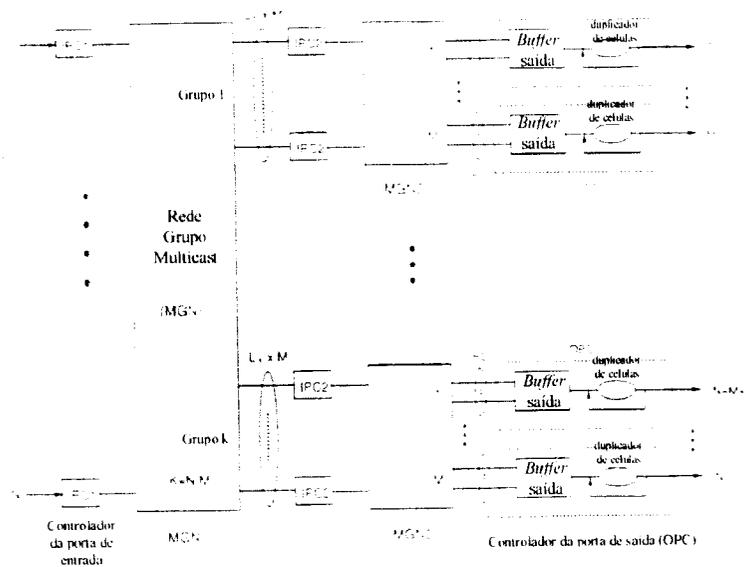
#### 4.1.6 Arquitetura modular e recursiva para implementar multicast em larga escala *MOBAS*

Proposta por Chao, Choe (1995), *MOBAS multicast output buffered ATM switch*, emprega uma extensão do princípio *knockout*, com o propósito de reduzir a complexidade do *hardware*.

A figura 4.14 mostra a arquitetura *MOBAS*, que consiste de controladores de portas de entrada (IPC1, IPC2), conjunto de redes *multicast* (MGN1, MGN2) e controladores de portas de saída (OPC). Os IPCs recebem as células, verificam as informações necessárias na tabela de translação e adicionam as informações de roteamento para que as células possam ser roteadas pelas MGNs. As MGNs copiam células *multicast* baseadas em suas informações de rota e enviam uma cópia para cada grupo de saída. Os OPCs armazenam temporariamente múltiplas cópias de células destinadas para uma determinada saída em um *buffer* de saída, gera múltiplas cópias para células *multicast* com o duplicador (CD), associa um novo VCI

obtido da tabela de translação para cada cópia e converte o formato da célula utilizado internamente para o formato padrão ATM, e envia a célula para seu destino final.

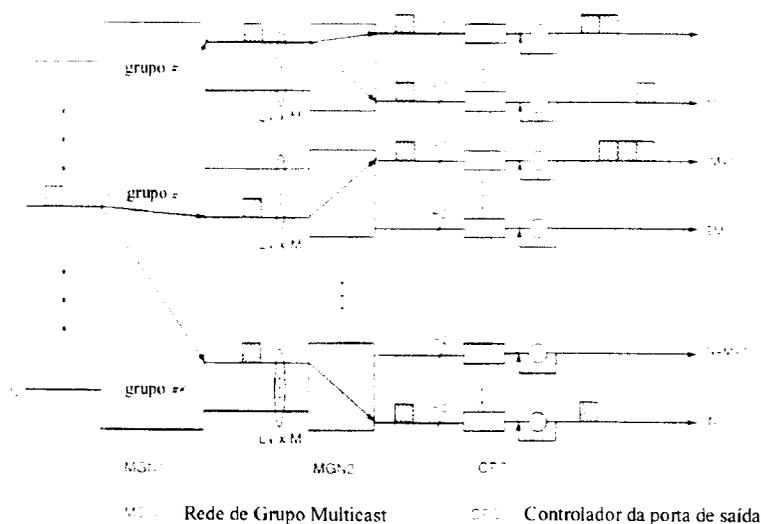
O conceito usado para implementar o *unicast* é uma combinação do princípio *knockout* e o compartilhamento de canais. Múltiplas portas de saída são agrupadas de tal forma que uma célula destinada para um determinado grupo de portas de saída pode compartilhar um mesmo *link* na estrutura de roteamento, o que reduz a complexidade das ligações internas. Sempre que o número de células destinadas a um determinado grupo exceder a capacidade do grupo, as células em excesso serão descartadas.



**Figura 4.14:** Arquitetura de chave ATM com buffer de saída MOBAS

No *multicast*, as células são replicadas nos módulos MGN1 e MGN2 e enviadas para múltiplas saídas. A figura 4.15 mostra um exemplo de como uma célula é copiada nos MGN's e duplicada no CD. Suponha que uma célula entre pela porta  $i$  com pedidos *multicast* para 4 portas de saídas: #1, #M, #(M + 1) e #N. Primeiramente a célula é enviada em *broadcast* para todos os  $k$  grupos de saída do MGN<sub>1</sub>, mas apenas os grupos #1, #2 e # $k$  aceitam a célula. Observe que apenas uma

cópia da célula apareceu em cada grupo e que as cópias podem aparecer em qualquer um dos  $L1 \times M$  links. A cópia da célula na saída do grupo #1 é agora replicada para 2 cópias no  $MGN_2$ . Na saída do  $MGN_2$ , existem 4 cópias da célula. Quando cada célula chega ao OPC, ela pode ser duplicada em múltiplas cópias pelo CD conforme a necessidade. Cada cópia duplicada no OPC é atualizada com um novo VCI obtido da tabela de translação do OPC antes de ser enviada para a rede. A razão para usar o CD é reduzir a quantidade de *buffer*, de tal forma a armazenar apenas uma cópia de cada célula *multicast* por porta de saída, ao invés de armazenar múltiplas cópias destinadas para múltiplos circuitos virtuais daquela mesma porta de saída.

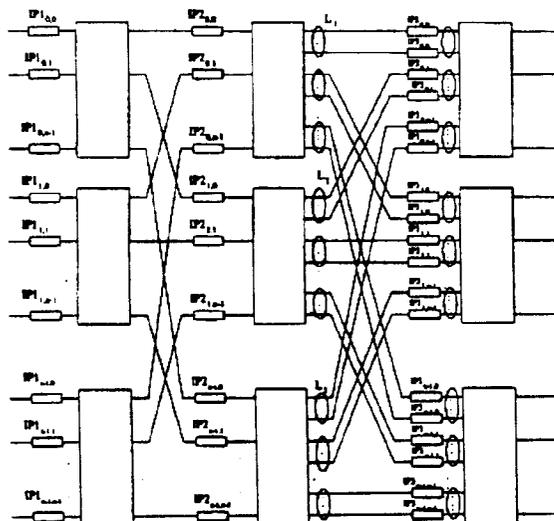


**Figura 4.15:** Exemplo de cópia de células multicast na MOBAS

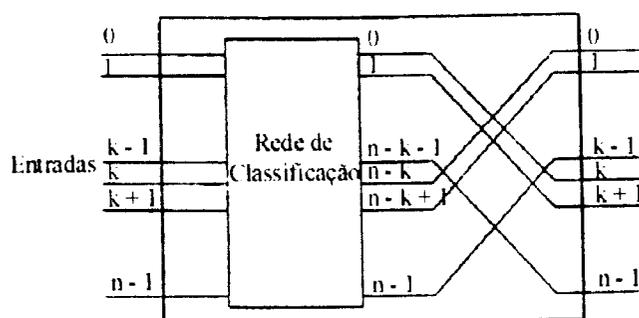
#### 4.1.7 A arquitetura Clos Knockout

A figura 4.16 mostra a arquitetura proposta por Chan et al. (1988), a qual consiste de 3 estágios diferentes, sendo o primeiro composto por  $N$  módulos de classificação contendo cada um  $N$  entradas e  $N$  saídas. O segundo estágio consiste de  $N$  elementos de chave *Knockout* sem *buffer* de saída e o terceiro estágio é composto por  $N$  elementos de chave *Knockout* modificados com *buffers* na saída.

Cada módulo do primeiro estágio é composto por uma rede *Batcher* com conexão cíclica conforme mostra a figura 4.17. O segundo estágio que é composto por elementos de comutação *Knockout* sem *buffer* de saída pode ser melhor visualizado na figura 4.18. A idéia básica é garantir uma expansão de  $L_1$  que assegure que ao menos  $L_1$  células possam ser destinadas simultaneamente para a mesma porta de saída sem haver perdas, tal que o tamanho do módulo de comutação seja  $N \times NL_1$ . Portanto, o segundo estágio é composto por  $N$  sub-módulos.



**Figura 4.16:** Arquitetura de uma chave multicast Clos-Knockout



**Figura 4.17:** Estrutura do modo de classificação

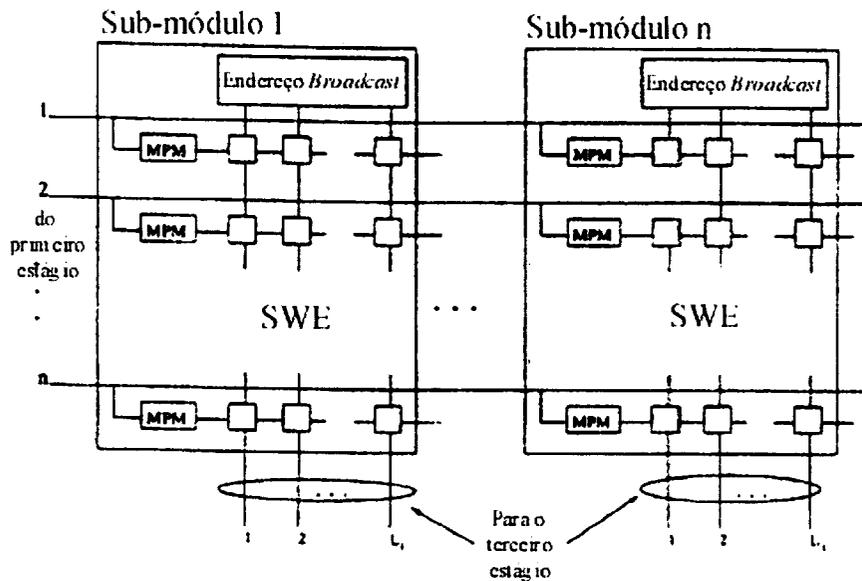


Figura 4.18: Módulo do segundo estágio da chave, elemento de comutação Knockout sem buffer

As células são primeiramente processadas pelo *multicast pattern masker* (MPMs) em cada sub-módulo. Sua função é atribuir todos os bits de endereçamento para zero, a menos do bit relativo ao módulo. Exemplo no sub-módulo  $i$  apenas o  $i$ -ésimo bit é preservado no seu valor original, sendo os demais mantidos em zero. Similarmente o *Address Broadcaster* (AB) do sub-módulo  $i$  irá criar células vazias com o *header* de roteamento do  $i$ -ésimo bit recebendo valor 1 e os outros bits recebendo 0. Usando estas células vazias, o *array of switch elements* (SWEs) em cada sub-módulo pode filtrar as células destinadas para as portas de saída dos outros sub-módulos. Ao mesmo tempo, ela irá reter ao menos  $L_i$  células destinadas a sua própria porta de saída. Após percorrermos o segundo estágio, as células seguem diretamente para o terceiro estágio em função da ausência de *buffer* na saída deste estágio. O terceiro estágio opera como o segundo, mas as células na saída são armazenadas em *buffers* e deixam a chave respeitando uma FIFO conforme mostra a figura 4.19.

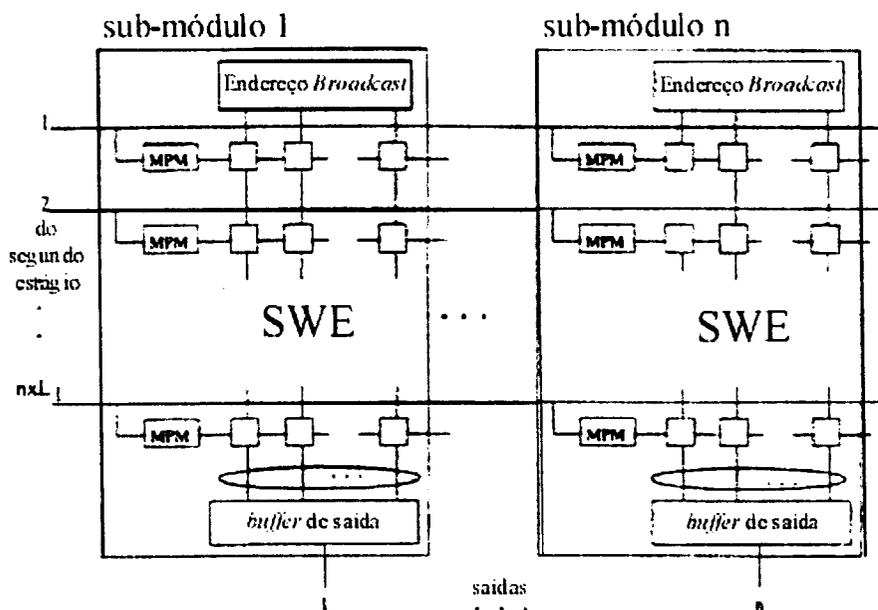


Figura 4.19: Estrutura modular do terceiro estágio da chave

#### 4.1.8 O compartilhamento de *buffer* empregado em chave ATM com o objetivo de facilitar a realização do *multicast*

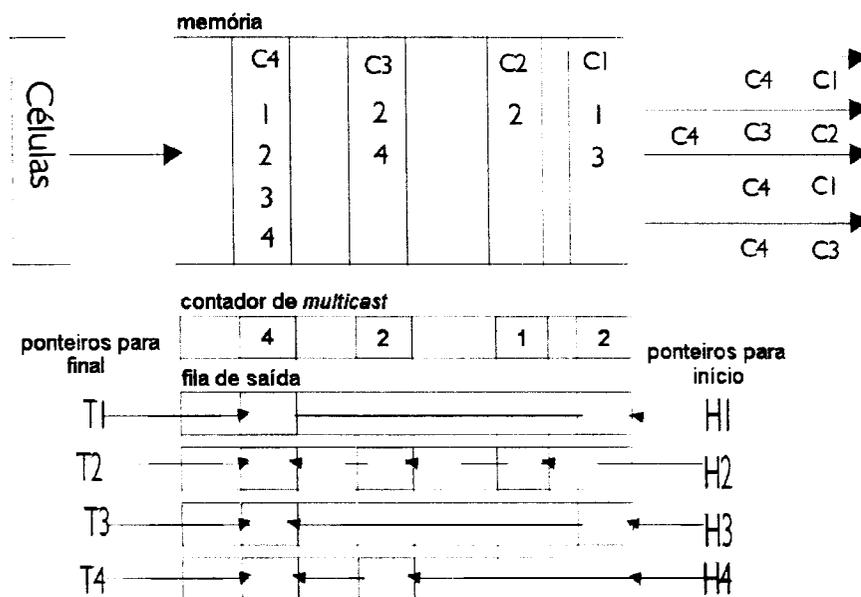
A chave ATM com compartilhamento de *buffer* proposta por Lin, Shung (1988) utiliza de forma eficiente os *buffers*. Isto facilita a implementação de *multicast*, visto que, uma célula armazenada em *buffer* pode ser lida por várias portas de saída, o que permite a realização do *multicast* sem a duplicação de células.

##### 4.1.8.1 *Multicast* em chaves ATM com *buffer* compartilhado

Em chaves ATM com *buffer* compartilhado, todas as células são misturadas no *buffer*, o que exige muito da estratégia de gerenciamento de filas adotada para o controle de células e a sua distribuição para múltiplos destinos.

A ilustração da figura 4.20 é uma adaptação direta da abordagem de duplicação de células para chaves ATM com *buffer* compartilhado. Quando células *multicast* entram em um *buffer* compartilhado, cópias de seu endereço ao invés de seu conteúdo são colocados em todas as filas de saída destino. Se uma célula

*multicast* é acessada por uma porta de saída destino, seu correspondente contador *multicast* tem seu valor diminuído de um. Quando o contador chega a zero, a célula é removida do *buffer* e sua posição de memória é liberada.



**Figura 4.20:** Duplicação de endereço multicast em chave ATM com buffer compartilhado

#### 4.1.8.2 Abordagem baseada em busca na fila

Na abordagem baseada em busca na fila, como ilustrado na figura 4.21, todas as células são armazenadas em *buffer* e seus *bitmaps* destinos são ajustados a direita através de seu cabeçalho FIFO. Cada porta de saída procura pelo *bitmap* armazenado no cabeçalho da FIFO para encontrar o primeiro bit ativo correspondente ao conjunto de células destinadas para aquela porta. O correspondente bit no *bitmap* da célula buferizada é resetado para zero após ter concluído todo processo de busca. Quando todos os *bitmaps* se tornam zero, a célula é removida da FIFO e todas as células armazenadas no lado esquerdo desta célula são deslocadas uma posição para a direita.

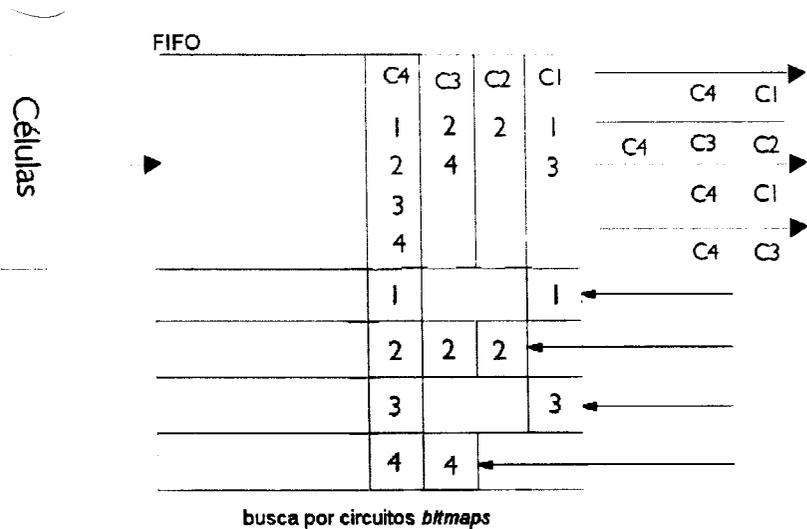


Figura 4.21: Pesquisa em filas de chaves ATM com buffer compartilhado

#### 4.1.8.3 Abordagem baseada no endereçamento de memória CAM

As chaves baseadas no CAM conforme mostra a figura 4.22, empregam um identificador CAM para localizar a seqüência numérica de um identificador de conexão para encontrar uma célula no *buffer*, e um *multicast* CAM (McCAM) para encontrar o *bitmap* destino para a conexão *multicast*. Isto segue uma disciplina *single release* e associa uma alta prioridade para células *multicast*. O McCAM é sempre o primeiro a ser checado, se ele mostrar que existe uma célula *multicast* armazenada no *buffer* para uma porta de saída, esta porta é bloqueada. Após todas as portas com *bitmap* destinos para células *multicast* serem bloqueadas, a célula é recuperada e entregue para todas as portas destino. Se uma porta vazia não encontrar um McCAM, a fila *unicast* será checada para ver se existe alguma célula armazenada no *buffer* que precise ser atendida.

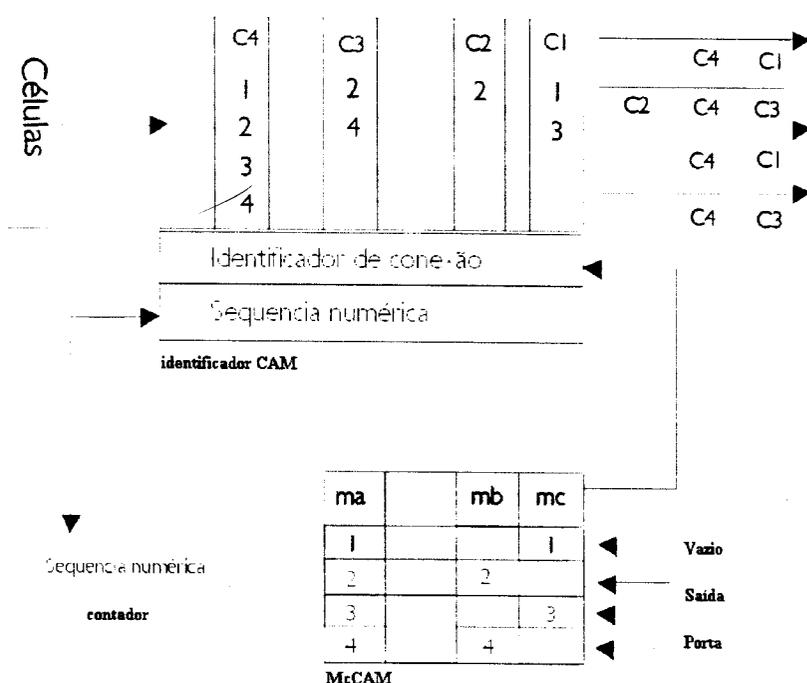


Figura 4.22: Multicast em chave ATM com buffer compartilhado baseada no CAM

#### 4.1.8.4 Abordagem de fila *multicast*

A abordagem de fila *multicast*, ao invés de usar um McCAM para monitorar todas as conexões *multicast*, a abordagem *multicast queue* organiza todas as células *multicast* armazenadas em *buffer* por ordem de chegada em uma lista chamada de fila *multicast* (*multicast queue*), como mostra a figura 4.23.

Esta abordagem é muito simples de ser implementada. Porém, devido a natureza sequencial da lista, o efeito conhecido como HOL (*head of line blocking*) irá limitar o desempenho para o tráfego *multicast*. Como mostra a figura 4.23 na abordagem *multicast queue*, apenas as células para as saídas 1 e 3 serão atendidas. A próxima célula *multicast* na fila *multicast* endereçada para as saídas 2 e 4 não serão atendidas no mesmo *slot* de tempo. Do contrário, nas três abordagens mencionadas anteriormente, a segunda célula *multicast* pode ser enviada para seus destinos no mesmo *slot* de tempo.

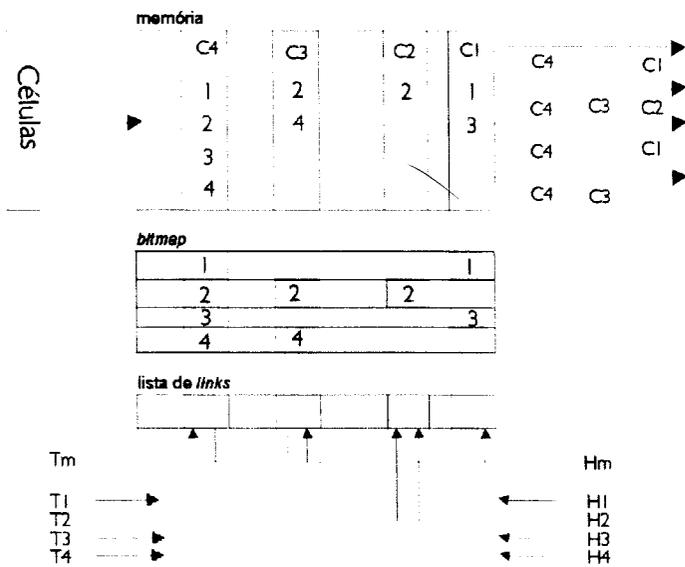


Figura 4.23: Multicast em chaves ATM com buffer compartilhado

#### 4.1.9 Multicast em chaves de alto desempenho

A arquitetura proposta por Urushidani et al. (1998) pode ser vista na figura 4.24. Ela consiste de uma rede de cópia capaz de realizar as cópias de acordo com um cabeçalho de cópia; um conjunto de *header translators* que substitui o cabeçalho de cópia das células pelo cabeçalho de roteamento; e uma rede de roteamento que realiza o roteamento das células *multicast* até seu destino final de acordo com o cabeçalho de roteamento.

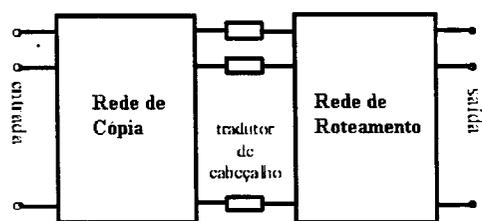


Figura 4.24: Arquitetura básica de uma chave multicast

A figura 4.25 mostra a arquitetura proposta chamada de rede de re\_roteamento. Ela é composta por elementos de chaveamento, *links* de roteamento e *links* que interconectam a mesma posição entre estágios adjacentes chamados de (*link by-pass*). Se a cópia de uma célula atinge a posição especificada (porta de saída desejada), a partir daquele estágio ela é roteada pelo *by pass links*. Células *multicast* que por um problema de contenção são desviadas de sua rota num estágio  $t$  reiniciam seu roteamento através da rede *Banyan* formada a partir do estágio  $t + 1$ .

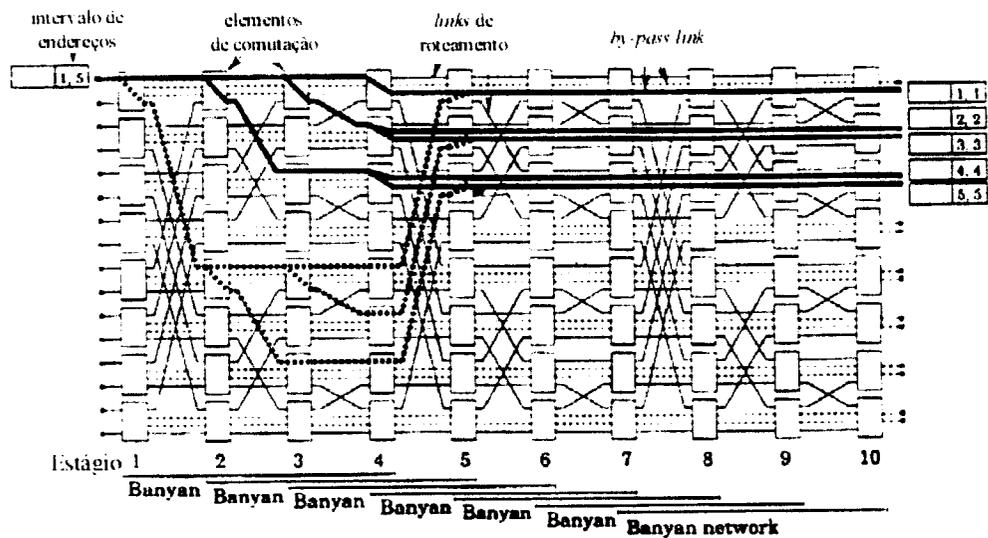


Figura 4.25: Arquitetura proposta

A figura 4.25 mostra um exemplo de cópia da porta de entrada 0000 para as portas de saída 0001 a 0101. A linha sólida mostra o circuito quando não ocorre contenção em nenhum estágio e a linha tracejada mostra o circuito quando a contenção ocorre no primeiro estágio.

Uma célula *multicast* pode ser replicada no elemento de chaveamento se o endereço final da cópia for contíguo. Para aliviar o tráfego as células *unicast* podem ser desviadas para o *link by-pass* a partir do primeiro estágio.

A figura 4.26, mostra a composição em diagramas de bloco do elemento de chaveamento: são dois controladores de entrada (IC0 e IC1), um controlador de contenção (CC), dois controladores de saída (OC0 e OC1) e dois controladores de *buffer* (BC0 e BC1). Uma célula que entra no *link* de roteamento é

replicada no controlador de entrada (IC) se necessário e então é roteada para os controladores de saída (OC) ou para o *link by-pass* através do OC e BC. Já a célula que entram pelo *by pass link* é roteada através do BC para o *link by-pass*.

O controlador de *buffer* (BC) é composto por 2 *buffers* A e B que armazenam múltiplas células e respeitam o princípio do, primeiro que entra é o primeiro que sai (FIFO). Existe também um seletor (Sel) que seleciona as células dos *buffers* A ou B e as roteia para os *links by-pass*.

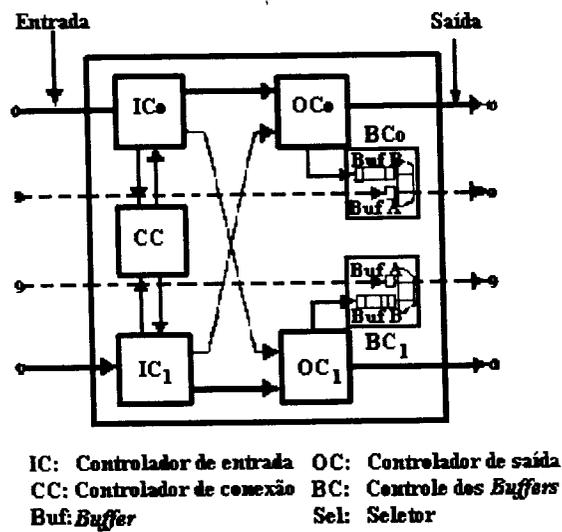


Figura 4.26: Estrutura interna do elemento de comutação

## 4.2 A arquitetura proposta

Com base nos métodos de replicação de pacotes utilizados, a arquitetura de chaves *multicast* pode ser dividida em dois grandes grupos: O primeiro grupo chamado de árvore de *multicast*, segundo Turner (1986) e Byun, Lee (1994) e o segundo grupo classificado como *broadcast* por Eng et al. (1988) e Chao, Choe (1995).

As chaves baseadas no princípio árvore de *multicast*, consistem de dois estágios. O primeiro estágio é uma rede de cópia com a função de replicar os

pacotes e o segundo estágio uma rede de roteamento, onde o objetivo é conduzir os pacotes para a porta de saída desejada. Para este tipo de chave, todo esforço dos pesquisadores tem se concentrado no projeto da rede de cópia.

É sabido que uma rede *Banyan* é não bloqueada conforme Prycker (1987), desde que, as células cheguem a sua entrada de forma ordenada segundo seu endereço destino. A solução para este tipo de problema tem sido a utilização de um estágio de classificação antes da rede *Banyan*. Emprega-se normalmente uma rede *Batcher*, para garantir a classificação antes de entregar a célula para a rede *Banyan*, compondo assim a estrutura *Batcher/Banyan*. A não classificação de células na entrada da rede *Banyan* levará a uma disputa desnecessária por um mesmo *link* de saída, podendo incorrer em perdas desnecessárias de células. Min et al. (1995) propôs uma chave *multi-channel* como uma rede de cópia não bloqueada. Redes de cópias recursivas conforme a proposta por Byun, Lee (1994) também foram introduzidas na tentativa de se reduzir a complexidade e aumentar a eficiência da rede de cópia.

A cópia das células para a realização do *multicast* é um obstáculo que tem despertado a atenção de muitos pesquisadores nos últimos anos. As soluções que se tem adotado para resolver este problema são: a utilização de uma rede de cópia na entrada da estrutura de roteamento; ou a utilização de alguma estratégia de *buffer* compartilhado na saída da estrutura de roteamento.

A rede cópia se encarrega de realizar tantas cópias das células quantas forem necessárias para que o pedido *multicast* possa ser atendido. Seu inconveniente é o fato de que ela irá adicionar ao tráfego de células *unicast* presentes na estrutura de roteamento, as cópias geradas para atender os pedidos *multicast*, aumentando assim a probabilidade de colisão e a conseqüente necessidade de se utilizar alguma estratégia de *buffer* ao longo da estrutura de roteamento.

Quando a opção é feita pela utilização de *buffer* compartilhado na saída da estrutura de roteamento, as cópias das células *multicast* são realizadas no momento em que a saída se torna disponível para recebê-la. Embora a utilização deste tipo de *buffer* seja extremamente interessante por eliminar a rede de cópia, o seu gerenciamento e custo tornam-se o grande gargalo.

Outro ponto importante a considerar é a restrição ao número de células com destino a mesma porta de saída. Quando mais de uma célula tem como objetivo a mesma porta de saída, com certeza uma situação de disputa ocorrerá em algum ponto da rede *Banyan*. Para solucionar este problema foram feitas várias propostas, sendo que, as mais relevantes foram mencionadas anteriormente neste capítulo.

Nas chaves *multicast* tipo *broadcast*, cada célula que entra é enviada em *broadcast* para todas as portas de saída. Nenhuma rede de cópia dedicada é requerida para a replicação das células. Um projeto conhecido e popular de chave deste tipo é a MOBAS de Chao, Choe (1995), que nada mais é que uma chave *Knockout* modificada. Por causa da natureza do compartilhamento do barramento da chave *Knockout*, o *broadcast* de células pode ser facilmente realizado. Porém, isto cria uma implicação na implementação do *hardware* e muita dificuldade de se implementar a chave *knockout* em termos práticos.

A arquitetura proposta neste trabalho, é totalmente modular e escalar, sendo composta por redes *Banyan* em cascata formando uma TBSF (*Tandem Banyan Switching Fabric*), empregando os conceitos de expansão do caminho e duplicidade de rota.

A expansão do caminho, consiste em multiplicar os *links* que interligam os elementos de chaveamento no interior da estrutura de roteamento, constituindo as vias pelas quais as células deverão trafegar de um ponto a outro da estrutura de roteamento. Para as simulações a expansão utilizada foi de 2, significando que, entre uma porta de entrada e uma dada saída, duas células poderão percorrer paralelamente sua rota sem qualquer conflito, sendo que, uma deverá percorrer sua rota pelo caminho padrão e a outra pelo caminho auxiliar.

A duplicidade de rota permite que em situações de disputa, uma célula possa ser desviada de sua rota original para uma rota alternativa que esteja disponível e ainda assim atingir seu objetivo. Entre uma rede *Banyan* e outra existe um estágio intermediário destinado a dar vazão às células que já atingiram a porta de saída desejada, e aliviar o tráfego no interior da estrutura.

A grande vantagem de se utilizar este estágio intermediário é que ele emprega elementos ativos, como o módulo de chaveamento. Isto permite que, na

ausência de células *unicast* destinadas àquela porta de saída, cópias antecipadas de células *multicast* sejam realizadas. Caso a saída alcançada pertença ao conjunto dos endereços *multicast*, células *multicast* poderão estar deixando a estrutura de roteamento já neste ponto, aliviando ainda mais o estágio seguinte da estrutura de roteamento. Em uma estrutura TBSF os estágios intermediários são constituídos por filtros que apenas retiram da estrutura de roteamento, as células que já atingiram a saída desejada.

Uma rede *Banyan* com  $N$  entradas e  $N$  saídas ( $N \times N$ ), é definida como um conjunto em cascata de  $n = \log_2 N$  estágios de elementos de chaveamento (SEs), sendo que, cada SE possui  $d$  entradas e  $d$  saídas ( $d \times d$ ), com  $N/d$  SEs em cada estágio.

A figura 4.27 a seguir, mostra a arquitetura proposta com  $N = 8$ ,  $d = 2$  e  $\varepsilon = 2$  (expansão). Note que como  $\varepsilon = 2$ , o estágio intermediário é composto por duas colunas de elementos de chaveamento. O número de colunas empregadas para o estágio intermediário, será uma função direta do valor de  $\varepsilon$ . Isto ocorre devido a possibilidade de células com o mesmo destino terem percorrido caminhos paralelos ( $\varepsilon = 2$ ) e precisarem ser chaveadas para a saída auxiliar a fim de atingir seu objetivo. Quando isto ocorre, a célula que percorreu sua trajetória pelo caminho padrão, deixa a estrutura de roteamento na primeira coluna, enquanto que, a célula que percorreu sua trajetória pelo caminho auxiliar, deixa a estrutura de roteamento na coluna seguinte. Isto acontece porque nos blocos intermediários apenas as saídas auxiliares permitem o acesso à porta de saída.

Na rede de cópia, além do estágio adicionado para células com o mesmo destino (função de  $\varepsilon$ ), acrescenta-se um segundo estágio (coluna) com o propósito de garantir a vazão de uma possível cópia de célula gerada na rede de cópia, para atender a um pedido *multicast* e que possa coincidentemente estar endereçada para o mesmo destino das anteriores.

Desta maneira, os estágios da arquitetura proposta podem ser calculados da seguinte forma: rede de roteamento  $n_r = \log_2 N$  ( $n_r$  número de estágios na rede de roteamento); saídas intermediárias  $n_s = \varepsilon - 2$  ( $n_s$  saídas intermediárias); e na rede de cópia o número de estágios é dado por  $n_c = \log_2 N + \varepsilon$  ( $n_c$  número de estágios na rede de cópia); o total de estágios será dado por  $n = n_r + n_s + n_c$ .

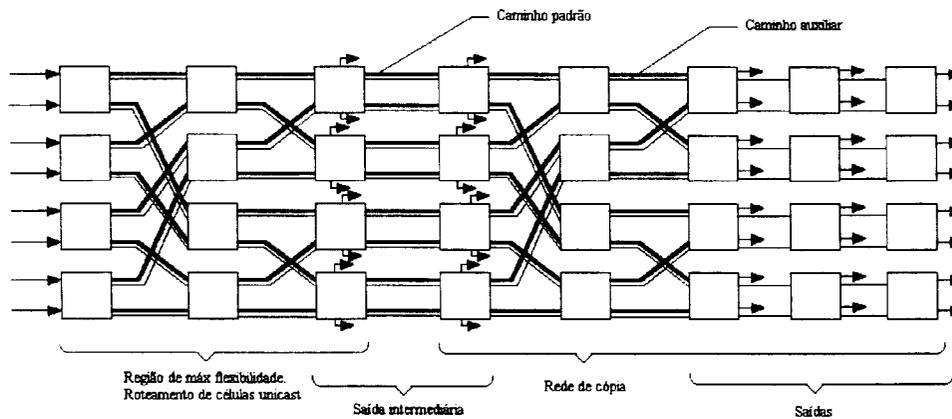


Figura 4.27: Diagrama em blocos da arquitetura proposta

No primeiro bloco as células são roteadas através da rede Banyan como se fossem células *unicast*, sem que as cópias das células *multicast* acrescentem tráfego à rede. Neste bloco, as células *unicast* disputam em cada *link* sua porta destino. Quando uma situação de disputa ocorre, a primeira tentativa é seguir pelo caminho auxiliar (paralelo a rota que seria ideal). Pode acontecer casos em que nem mesmo o caminho auxiliar esteja disponível; neste caso, a célula é desviada para uma rota alternativa e só voltara a disputar seu *link* de saída no segundo bloco da TBSF. O ponto que é considerado mais relevante na região de roteamento de células *unicast*, é a flexibilidade na disputa por um *link* de saída, reduzindo a taxa de perdas de células nesta região a níveis próximos de zero. Outro ponto relevante é a ausência de cópias de células *multicast* congestionando a estrutura. O roteamento de células *unicast* no primeiro bloco, alivia o tráfego de células *multicast* na rede de cópia.

A rede de cópia possui algumas características diferentes da rede de roteamento. Nela são feitas as cópias das células *multicast*, agora com o tráfego reduzido, devido a entrega de grande parte das células *unicast* terem ocorrido no primeiro estágio. As colisões recebem um tratamento diferenciado. Quando ocorre uma disputa por um determinado *link* de saída, a única alternativa é o caminho auxiliar, não sendo mais permitido o desvio para uma rota alternativa. A rede de cópia pode também ser chamada de rede de convergência, pois as células necessariamente devem convergir para seus *links* de saída, sendo portanto, um bloco

bem mais rígido quanto ao roteamento das células. Na impossibilidade de ser atendida neste estágio, a célula será descartada.

A rede de cópia termina com um estágio de entrega, que segue a mesma lógica do estágio de entrega intermediário acrescido de um estágio (coluna), para dar vazão a uma possível cópia de células, que embora tenham atingido seu destino o fizeram pelo caminho auxiliar, e precisam ser roteadas para o caminho padrão para que possam ser entregues a seu destino.

Esta estrutura é totalmente modular e não utiliza *buffer* em seus *links* internos, permitindo um ganho considerável em desempenho e na redução dos custos. Neste trabalho não foi levado em conta o módulo de saída da chave. Este módulo é importante pois, como as células deixam a estrutura de roteamento em pontos diferentes, a arquitetura tem como inconveniente a possibilidade da perda de seqüência, a qual precisa ser tratada no módulo de saída.

A figura 4.28 (a) e (b) a seguir traz uma representação em diagrama de blocos do módulo padrão e módulo de entrega, utilizados no roteamento das células.

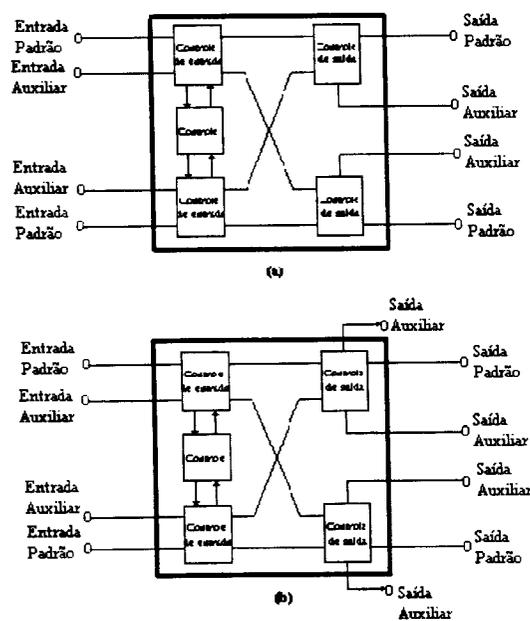


Figura 4.28: Representação esquemática do módulo de roteamento padrão (a) e módulo de roteamento com saídas auxiliares (b)

No capítulo 5, esta arquitetura será simulada o que permitirá analisar o comportamento desta estrutura quando submetido ao tráfego de células ATM.

## Capítulo 5

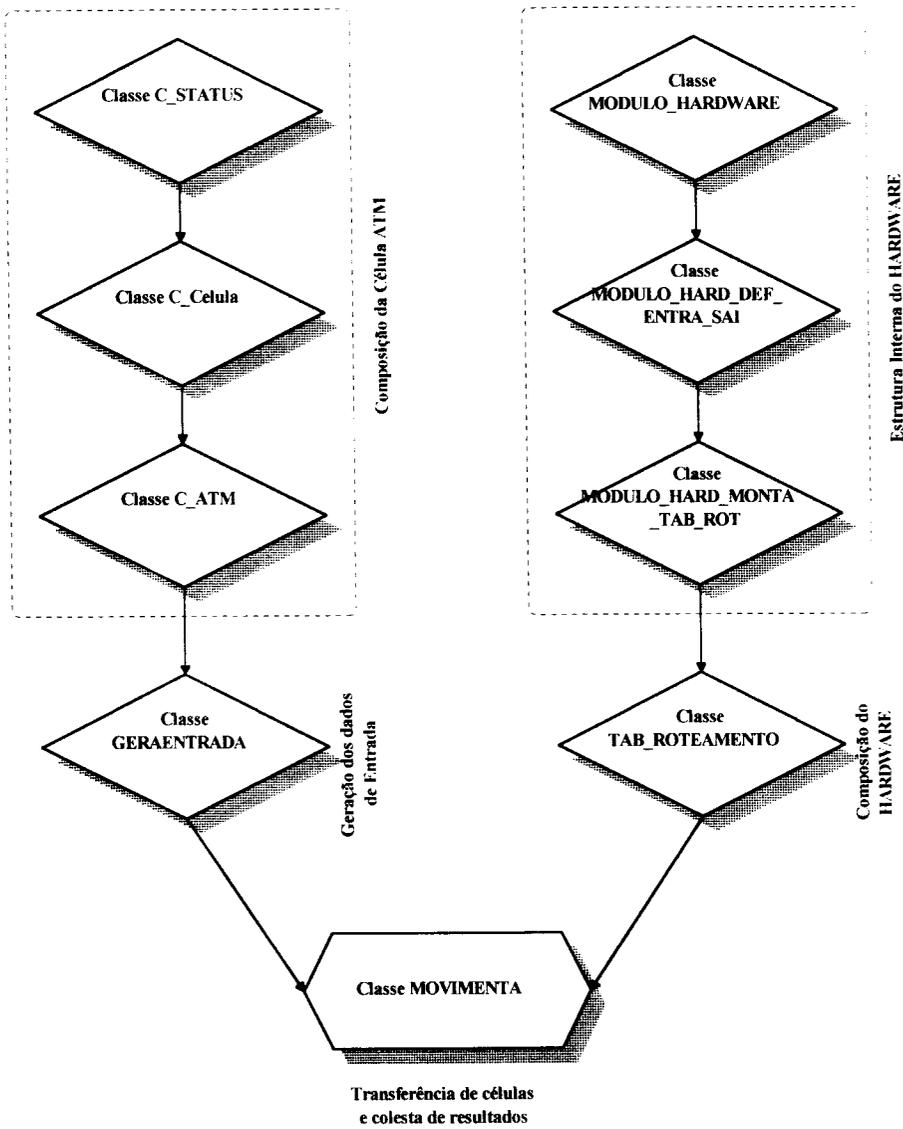
*Este capítulo descreve o simulador e apresenta os resultados obtidos com a simulação.*

### **Simulador e resultados da simulação**

#### **5.1 O Simulador**

O simulador escrito em C++ apresenta três hierarquias de classes principais: a primeira responsável pela célula ATM; a segunda responsável pelo *hardware* da estrutura de roteamento, modelado conforme descrito no capítulo 4; e a terceira responsável pelo tráfego de células ATM no interior da estrutura de roteamento, e também pela coleta dos resultados da simulação.

A figura 5.1 apresenta em diagramas de blocos as principais classes empregadas na composição do simulador, e a forma como elas se relacionam. As classes C\_STATUS, C\_CELULA e C\_ATM tratam a célula ATM; a classe GERAENTRADA é a responsável pela geração dos dados de entrada; e as classes MODULO\_HARDWARE, MODULO\_HARD\_DEF\_ENTRA\_SAI e MODULO\_HARD\_MONTA\_TAB\_ROT compõem a estrutura interna do HARDWARE; e a classe TAB\_ROTAMENTO reúne as três classes anteriores para formar a estrutura de roteamento proposta.



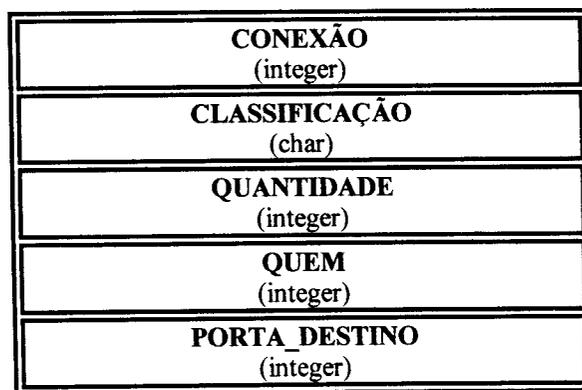
**Figura 5.1:** Diagrama de blocos da relação existente entre as diversas classes que compõem o simulador

### 5.1.1 A célula ATM

A célula ATM é representada pelas classes C\_STATUS, C\_CELULA e C\_ATM. A classe GERAENTRADA tem como membro, um objeto do tipo

CELULA\_ATM e será a classe responsável pela geração dos dados de entrada. A forma como estas classes se relacionam será descrita a seguir:

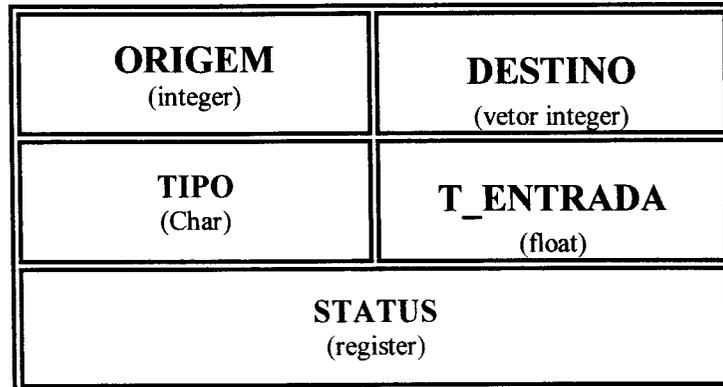
- a) Classe C\_STATUS: é a responsável pela composição do cabeçalho da célula ATM, e pelo controle da conexão, figura 5.2. É composta pelos seguintes membros: CONEXÃO que controla o número da conexão a qual a célula pertence; QUANTIDADE controle da quantidade de cópias de células que devem ser geradas em uma conexão *multicast*; QUEM especifica qual o número da cópia, só é válido para conexões *multicast*; PORTA\_DESTINO especifica a porta de saída desejada; e CLASSIFICAÇÃO que classifica a célula como sendo uma fonte (mãe) ou uma cópia (filha), também se aplica apenas a conexões *multicast*. Os métodos utilizados nesta classe realizam operações de leitura e escrita nos membros privados da classe.



**Figura 5.2:** STATUS de uma célula ATM  
(referência para o simulador)

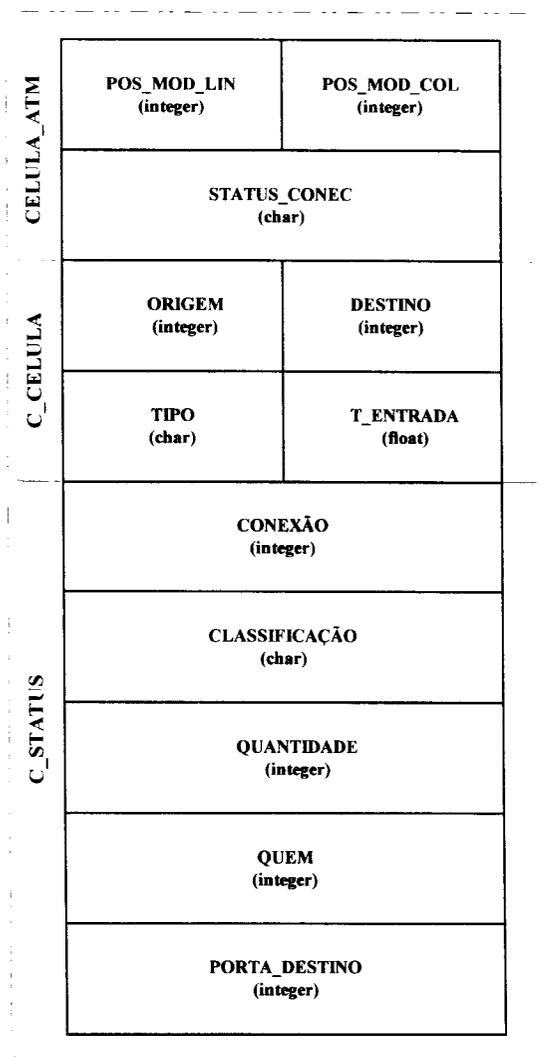
- b) Classe C\_CELULA: juntamente com a classe C\_STATUS compõem o cabeçalho da célula ATM proposto, figura 5.3. É composta pelos seguintes membros: ORIGEM que especifica a porta na qual a célula deve ser inserida na estrutura de roteamento; DESTINO[] que é um vetor de controle dos destinos da conexão aplicado às cópias de células *multicast*. Neste tipo de conexão (*multicast*), a célula mãe é destinada para o endereço especificado em PORTA\_DESTINO, e as cópias (células filhas) são encaminhadas para os endereços assinalados no vetor

DESTINO[]; e o membro TIPO especifica a natureza da conexão solicitada (*unicast* ou *multicast*). Os métodos pertencentes a esta classe, realizam operações de leitura e escrita nos membros privados da classe.



**Figura 5.3:** Cabeçalho de uma célula ATM (referência para o simulador)

- c) Classe CELULA\_ATM: é derivada das classes C\_STATUS e C\_CELULA, sendo a responsável pela composição final da célula ATM e dos dispositivos de controle utilizados na simulação. É composta pelos membros: POS\_MOD\_LIN e POS\_MOD\_COL que registram, passo a passo, a posição da célula (linha x coluna) no seu trajeto pela estrutura de roteamento; O membro STATUS\_CONEC monitora o *status* da conexão, assinalando com “C” as conexões que foram aceitas, “R” as que foram recusadas, “A” as conexões que estão ativas e “X” para sinalizar que houve colisão. A figura 5.4 representa uma célula ATM utilizada na simulação.



**Figura 5.4:** Representação de uma célula ATM utilizada para testes de simulação

- d) Classe GERAENTRADA: reúne todas as anteriores através de seu membro SOLENT[], vetor de 16 posições do tipo CÉLULA\_ATM, representando as 16 portas de entrada da estrutura de roteamento (consideradas para simulação). Um dos métodos que merece destaque é o GERATIPO(), encarregado de gerar os tipos das células: “V” para células vazias, representando a ausência de célula em uma determinada porta de entrada; “U” para células *unicast*; e “M” para células *multicast*.

Os tipos são gerados por meio de um gerador randômico, que retorna um índice que será utilizado para apontar para uma posição do vetor GTP[]. Este vetor, conforme a figura 5.5, é um vetor de caracteres que tem suas posições preenchidas pelos tipos válidos descritos anteriormente. A célula gerada terá como tipo o valor de retorno desta operação.

V	V	U	V	U	V	V	M	V	V	U	V	V	U	V	V
V	V	U	V	U	V	V	U	V	V	U	V	V	U	V	V

**Figura 5.5:** Representação esquemática do vetor GTP utilizado como referência para a geração dos tipos de células pelo simulador

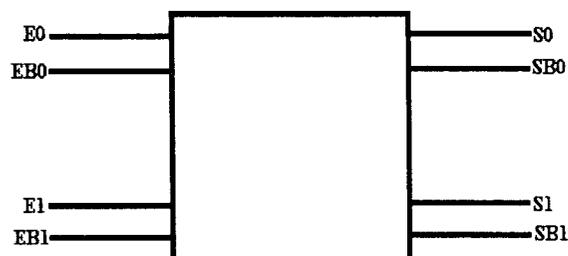
O método GERANCONEC() é responsável por gerar de forma aleatória um número entre 0 e 999 (para efeito de simulação), que irá representar o número da conexão a qual célula pertence. GERAPENT() é o método que associa um número entre 0 e 15 (representando as 16 portas propostas para a estrutura de roteamento), forma aleatória às células geradas. Este número irá definir a qual porta de entrada a célula deve ser dirigida. Como a geração das portas é aleatória, este método utiliza um vetor auxiliar GPE[] de 16 posições, como uma referência para as portas de entrada já assinaladas (atribuídas a uma determinada célula), evitando que uma porta de entrada já utilizada seja atribuída a outra célula. A idéia por trás da utilização do GPE[] é bastante simples. Como a cada ciclo 16 células são geradas, sempre que uma célula é gerada, o sistema varre o vetor e verifica se a porta já foi atribuída, caso tenha sido utilizada o processo se repete até que uma porta livre seja encontrada. GERAPDEST() é o método responsável por gerar de forma aleatória as portas de saída para as quais as células devem ser encaminhadas. Para as portas de saída não existe restrição quanto a repetição de portas destinos, o que possibilita que múltiplas células sejam endereçadas para a mesma porta de saída. GERAMDEST() gera de forma aleatória os destinos para as cópias das células *multicast*. Neste caso, como nas portas de entrada, um cuidado especial é tomado para não haver repetição de endereço no vetor DESTINO[], pois

não faria sentido duas cópias de uma mesma célula serem endereçadas para a mesma porta de saída. O método GERATENT() é responsável por capturar do sistema o instante em que a célula entra na estrutura de roteamento. MONT\_ARQ\_GERA() armazena as entradas em um arquivo binário chamado GERA\_ENT, que será utilizado na simulação como massa de dados de entrada.

### 5.1.2 O Hardware

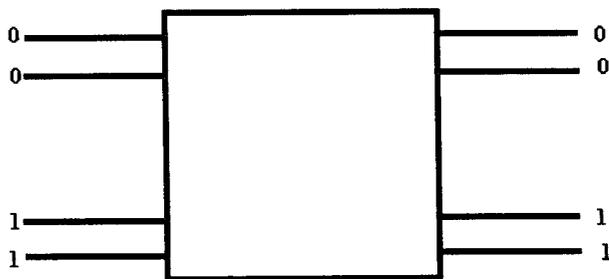
O *hardware* que compõe a estrutura de roteamento é formado pelas classes MODULO\_HARDWARE, MODULO\_HARD\_DEF\_ENTRA\_SAI e MODULO\_HARD\_MONTA\_TAB\_ROT. A classe TAB\_ROTAMENTO é a responsável pela composição da estrutura física do *hardware*, reunindo as classes anteriores. Estas classes se relacionam conforme o descrito a seguir:

- a) Classe MODULO\_HARDWARE: define o aspecto externo de cada módulo no interior da estrutura de roteamento conforme a figura 5.6 a seguir. O número de portas de entrada/saída é função da posição do módulo de roteamento no interior da estrutura de roteamento. É composta pelos seguintes membros: ENTRADA/SAIDA que controla a quantidade de portas de entrada/saída e LINHA/COLUNA que controla a posição do módulo no interior da estrutura de roteamento. Os métodos utilizados nesta classe realizam operações de leitura e escrita nos seus membros privados;



**Figura 5.6:** Representação esquemática do módulo hardware visto por fora como uma caixa preta

b) Classe `MODULO_HARD_DEF_ENTRA_SAI`: a partir da posição e do número de portas entrada/saída estabelecidos na classe `MODULO_HARDWARE`, define-se quem são estas portas (um valor associado a cada porta que se relaciona com o ambiente externo), conforme a figura 5.7 a seguir. Seus membros são: `ENTRADA_0[]`, `ENTRADA_1[]` (consideradas como as entradas principais), `ENTRADA_B0[]`, `ENTRADA_B1[]` (entradas alternativas), `SAIDA_0[]`, `SAIDA_1[]` (saídas principais) e `SAIDA_B0[]`, `SAIDA_B1[]` (saídas alternativas). Na posição 0 destes vetores está especificado o número da porta (0 a 15). Na posição 1, o *status* da porta (0 para porta inativa e 1 para porta ativa). Na posição 2, a porta destino para a qual a célula deve ser encaminhada (0 a 15). Os métodos desta classe realizam operações de leitura e escrita em seus membros privados;



**Figura 5.7:** Representação das portas de entrada e saída em um objeto do tipo `MODULO_HARD`

c) Classe `MODULO_HARD_MONTA_TAB_ROT`: esta classe monta uma tabela 4x5 (tabela de roteamento), na qual as possibilidades de rotas disponíveis para uma célula no interior do módulo de roteamento estão mapeadas. Nesta tabela, as linhas representam as possibilidades de entrada (`ENTRADA_0`, `ENTRADA_B0`, `ENTRADA_1` e `ENTRADA_B1`) e as colunas as possibilidades de saída (`SAIDA_0`, `SAIDA_B0`, `SAIDA_1`, `SAIDA_B1` e `DESCARTE` (quando não for possível atender o destino solicitado)). Os membros mais importantes desta classe são: `TAB_ROT_INTERNO[][]`, tabela de inteiros na qual estão mapeadas as possibilidades de rotas conforme figura 5.8 a seguir: `BSAI_0`, `BSAI_B0`, `BSAI_1` e `BSAI_B1` são os membros destinados ao controle lógico das portas de saída.

Métodos que se destacam: SETA\_T\_RTINNT() leva em conta o número de portas de entrada/saída definidas na classe MODULO\_HARDWARE e sinaliza estas portas na tabela TAB\_ROT\_INTERNO, atribuindo valor 1 às portas disponíveis (válidas naquela posição da estrutura de roteamento), e 8 às portas não disponíveis (não válidas naquela posição da estrutura de roteamento). O método ASSIM\_TAB() é responsável por assinalar na TAB\_ROT\_INTERNO a porta de entrada ativa. Estas portas são assinaladas com 0 (portas que receberam uma solicitação de entrada/saída). O método VER\_PRIOR() varre a tabela de roteamento, coluna a coluna, atribuindo prioridades para o atendimento das células conforme figura 5.9 a seguir. O método ARQ\_PERDAS() registra em um arquivo binário as células que foram descartadas.

	S J SAIDA_0	SB J SAIDA_B0	S J+1 SAIDA_1	SB J+1 SAIDA_B1	D DESCARTE
E J ENTRADA_0					
EB J ENTRADA_B0					
E J+1 ENTRADA_1					
EB J+1 ENTRADA_B1					

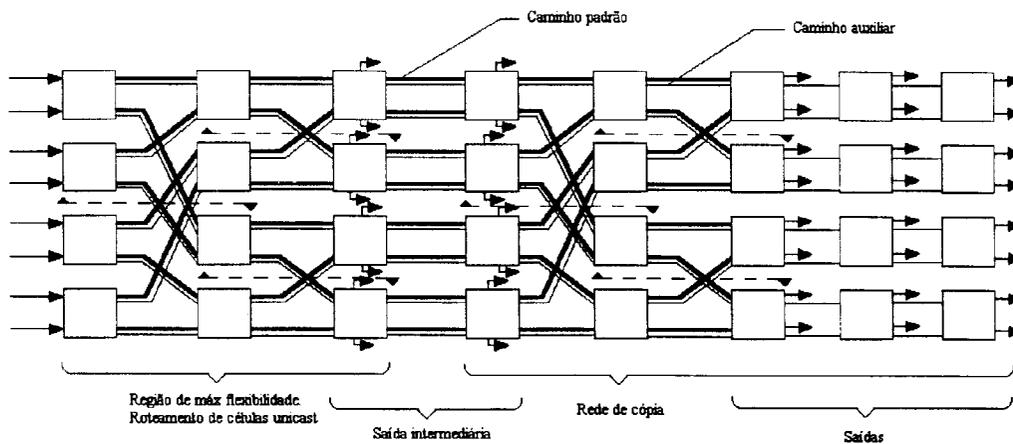
**Figura 5.8:** Tabela de roteamento interno a cada objeto do tipo MODULO\_HARD\_DEF\_ENTRA\_SAI, referência para o roteamento das células entre as portas de entrada e saída de um mesmo objeto. J representa a coluna na qual o objeto está inserido na TABELA DE ROTEAMENTO

	S J SAIDA_0	SB J SAIDA_B0	S J+1 SAIDA_1	SB J+1 SAIDA_B1	D DESCARTE
E J ENTRADA_0					
EB J ENTRADA_B0					
E J+1 ENTRADA_1					
EB J+1 ENTRADA_B1					

**Figura 5.9:** Tabela de roteamento interno destacando as regiões com prioridade no atendimento a suas solicitações

- d) Classe `TAB_ROTAMENTO`: é a classe que reúne todas as anteriores através de seu único membro `ROTEAMENTO[[]]`, que é uma matriz do tipo `MODULO_HARD_MONTA_TAB_ROT`, representando a estrutura de roteamento conforme figura 5.10 (representação esquemática das regiões da estrutura de roteamento). As linhas/colunas da matriz `ROTEAMENTO[[]]` são compostas por elementos do tipo `MODULO_HARD_MONTA_TAB_ROT` num total de 8 linhas, o que perfaz as 16 portas de entradas por 9 colunas. Os métodos desta classe que merecem destaque são: `INIC_TAB_ROT()`, inicializa a tabela de roteamento definindo quem são as portas de entrada, saída e descarte para cada módulo da estrutura; `SET_EATV()` e `SET_SATV()`, o seu objetivo é atribuir um valor de controle as entradas/saídas que serão ativadas (preparadas para receber uma célula); `COP_MULT()`, realiza as cópias das células *multicast*; `SET_PDEST()`, assinala as portas destino da estrutura de roteamento; `LE_TRI_SS()`, este método é interessante pois é o responsável por executar uma

varredura na tabela de roteamento interno, módulo a módulo, e fazer a associação com as portas de entrada/saída da classe `MODULO_HARD_DEF_ENTRA_SAI`; `LI_SAI_ENT()` é o método encarregado de varrer todas as saídas de uma coluna da estrutura de roteamento para associá-las com as entradas da coluna seguinte, em outras palavras, movimenta as células de uma coluna a outra; `ARQ_PDEST()`, responsável por assinalar as células que obtiveram sucesso na travessia da estrutura de roteamento (células que atingiram seu destino); `HEMISFERIO()`, este método leva em conta a simetria da estrutura de roteamento, retornando para cada célula o hemisfério (a estrutura de roteamento está dividida em 2 hemisférios, o superior e o inferior que variam com a posição da célula no interior da estrutura de roteamento) para o qual ela deve ser direcionada, para que chegue ao seu destino; `PEGA_PDEST()`, `PEGA_SPDEST()` e `PEGA_MPDEST()` são métodos que retornam o valor da porta destino nas portas de entrada, porta de saída e em células `MULTICAST`; `LE_LIN_ENT()`, este método é importante por ser o encarregado de executar a conversão do número da porta entrada/saída que variam de 0 a 15 para as linhas da matriz `ROTEAMENTO` que variam de 0 a 8, pois cada objeto que compõe esta matriz possui duas entradas e duas saídas principais, mais duas entradas e duas saídas auxiliares sinalizadas pela mesma referência da entrada/saída principal (mesmo número de porta). Este método leva em conta também a coluna sob análise pois, a quantidade de portas entrada/saída dos objetos da matriz variam em função de sua posição na matriz `ROTEAMENTO`; `VER_ENT_ATV()`, `VER_SAI_ATV()` e `VER_MSAI_ATV()` são responsáveis por varrer as colunas da matriz `ROTEAMENTO` em busca de portas entrada/saída ativas e também saídas com células *multicast*.



**Figura 5.10:** Representação das regiões da estrutura de roteamento proposta: As linhas tracejadas dividem a estrutura em hemisférios superior e inferior

### 5.1.3 A movimentação das células na estrutura de roteamento

A movimentação das células e coleta de resultados na estrutura de roteamento é de responsabilidade da classe MOVIMENTA, que através de seus membros privados C\_ATM e EST\_ROT integram as duas hierarquias de classes descritas anteriormente, compondo as partes do simulador. A composição é feita da seguinte forma:

- a) Classe Movimenta: composta pelos membros C\_ATM do tipo GERAENTRADA e EST\_ROT do tipo TAB\_ROTAMENTO, possui os seguintes métodos principais: LE\_RARGERA() método encarregado de ler o arquivo GERA\_ENT e de assinalar as portas de entrada da estrutura de roteamento conforme solicitado em GERA\_ENT; CONTA\_GERA(), CONTA\_UNI\_MULT(), CONTA\_SUCS(), CONTA\_PERDIDAS e CONTA\_MPERDIDAS() são os métodos encarregados pela contabilidade do sistema onde: CONTA\_GERA() contabiliza o total de células úteis (células do tipo “U” e “M”), contidas no arquivo GERA\_ENT e que deverão percorrer a estrutura de roteamento; CONTA\_UNI\_MULT() faz a

separação da quantidade de células úteis definindo quantas são *unicast* e quantas são *multicast*; `CONTA_SUCS()` contabiliza a quantidade de células que obteve sucesso (atingiram o destino correto) ao percorrer a estrutura de roteamento; `CONTA_PERDIDAS()` contabiliza as células tipo *unicast* que foram descartas ao longo da estrutura de roteamento; e `CONTA_MPERDIDAS()` contabiliza as células tipo *multicast* que foram descartadas ao longo da estrutura de roteamento; `VE_ENT_SETA_C()` é o método encarregado pela leitura dos dados de entrada e por assinalar as variáveis de entrada da classe `MODULO_HARD_DEF_ENTRA_SAI`, sinalizando as portas ativas. O método `LI_ENT_SAI()`, tem por função ligar em um mesmo objeto da estrutura de roteamento as entradas ativas com as saídas correspondentes; `PASSO_PASSO()` é o método encarregado de chamar os anteriores em uma seqüência tal que permita que uma célula possa percorrer toda estrutura de roteamento coluna a coluna até atingir seu estágio final. Para que isto ocorra, `PASSO_PASSO()` inicializa seu processo de chamada dos métodos anteriores pela última coluna (coluna 9), descendo a cada ciclo do simulador em direção da coluna 0 ou porta de entrada da estrutura de roteamento.

## 5.2 A Simulação

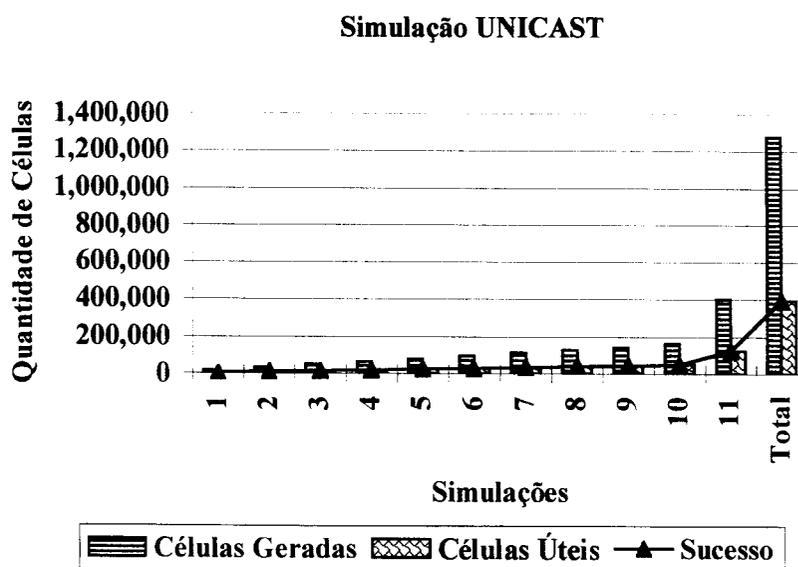
A simulação foi dividida em duas partes sendo a primeira destinada a atender apenas pedidos *unicast* e a segunda, atendendo pedidos *unicast* e *multicast*. Cada unidade de simulação é composta de 11 arquivos `GERA_ENT` totalizando 1.280.000 células geradas, distribuídas em ordem crescente. Os resultados destas simulações foram agrupados em tabelas e gráficos conforme os itens a seguir:

### 5.2 Simulação de células *UNICAST*

Nesta simulação, 100% das células úteis que percorrem a estrutura de roteamento, são do tipo *unicast*. Não houve um controle prévio dos dados gerados, a menos da conversão dos pedidos *multicast* para pedidos *unicast*.

**Tabela 5.1:** Simulação unicast: nesta simulação, todos os pedidos multicast foram substituídos por células unicast, trafegando pela estrutura de roteamento somente células unicast

Simulação	Células Geradas	Células Úteis	Sucesso	Perdas
1	16.000	4.979	4.979	0
2	32.000	9.865	9.865	0
3	48.000	14.809	14.809	0
4	64.000	19.771	19.770	1
5	80.000	24.749	24.749	0
6	96.000	29.624	29.624	0
7	112.000	34.648	34.648	0
8	128.000	39.470	39.469	1
9	144.000	44.456	44.456	0
10	160.000	49.452	49.452	0
11	400.000	123.561	123.561	0
<b>Total</b>	<b>1.280.000</b>	<b>395.384</b>	<b>395.382</b>	<b>2</b>



**Figura 5.11:** Simulação unicast livre de controle dos dados gerados

### Simulação UNICAST

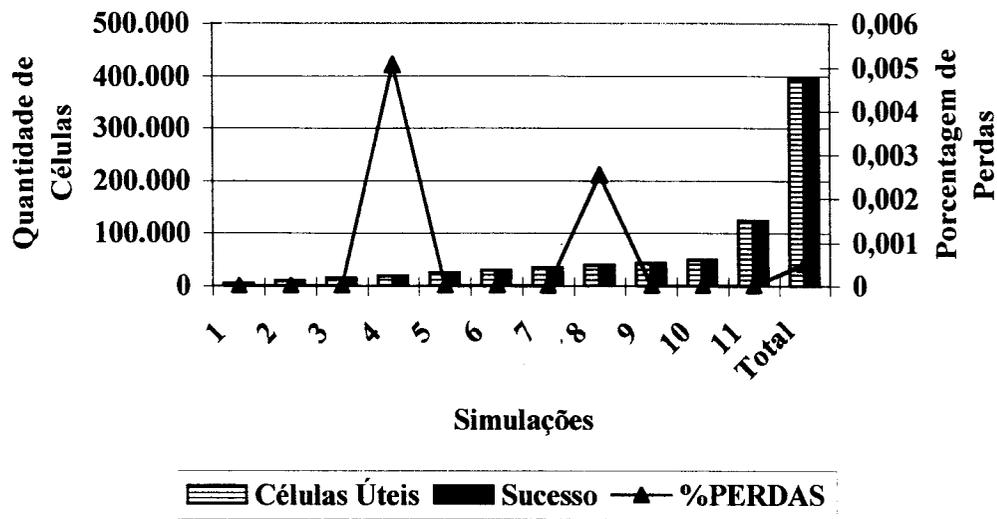


Figura 5.12: Representação da percentagem de perdas das células em relação às células úteis geradas

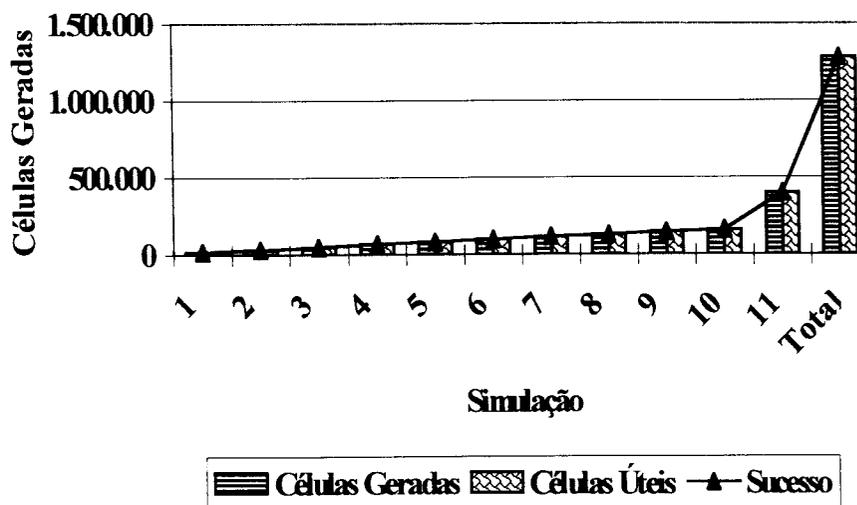
As células perdidas nesta simulação, foram decorrentes de uma demanda excessiva de células com destino à mesma porta de saída, em um mesmo ciclo.

No segundo grupo de simulação *unicast*, foi imposta a condição de que todas as 16 entradas da estrutura de roteamento (corresponde a um registro do arquivo GERA\_ENT), fossem submetidas a uma solicitação *unicast*. O objetivo desta simulação é submeter a estrutura de roteamento a um caso crítico, onde todas as suas portas, a todo instante estarão recebendo células. Por outro lado, não houve restrição com respeito as portas de saída. Isto permitiu que em um mesmo registro (16 solicitações *unicast*) houvesse um número elevado de células com destino a mesma porta de saída. É interessante observar o aumento no número de colisões, ocasionados pela disputa de um mesmo *link* no interior da estrutura de roteamento.

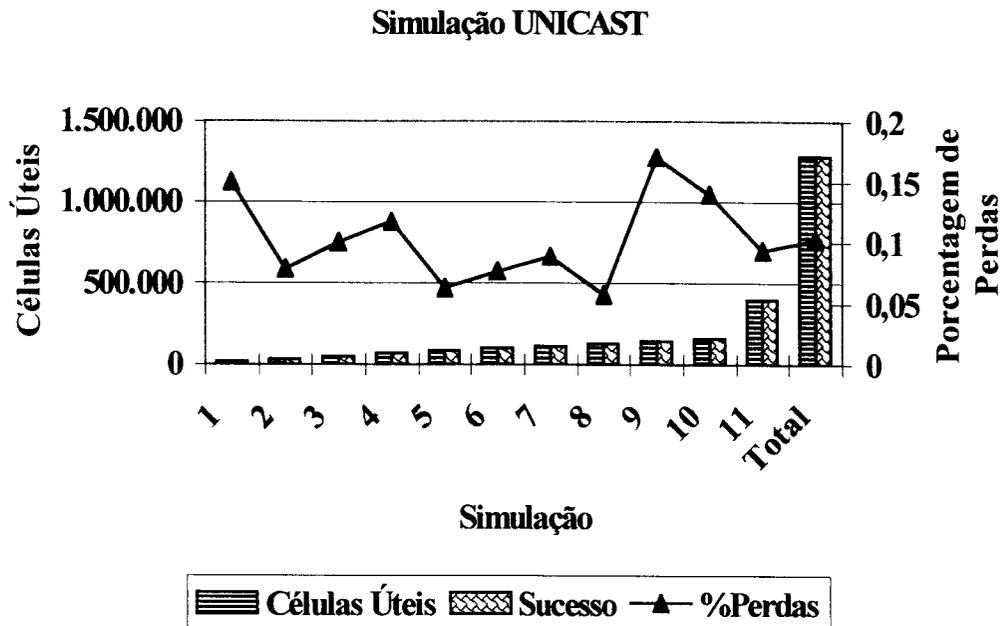
**Tabela 5.1:** Simulação unicast. Nesta simulação todos os pedidos multicast e células vazias foram substituídos por solicitações unicast

Simulação	Células Geradas	Células Úteis	Sucesso	Perdas
1	16.000	16.000	15.976	24
2	32.000	32.000	31.975	25
3	48.000	48.000	47.952	48
4	64.000	64.000	63.925	75
5	80.000	80.000	79.950	50
6	96.000	96.000	95.927	73
7	112.000	112.000	111.901	99
8	128.000	128.000	127.927	73
9	144.000	144.000	143.755	245
10	160.000	160.000	159.777	223
11	400.000	400.000	399.628	372
<b>Total</b>	<b>1.280.000</b>	<b>1.280.000</b>	<b>1.278.693</b>	<b>1.307</b>

### Simulação UNICAST



**Figura 5.13:** Simulação unicast com a imposição de que todas as portas de entradas sejam solicitadas por células úteis



**Figura 5.14:** Representação da porcentagem de perdas de células em relação às células úteis que foram geradas

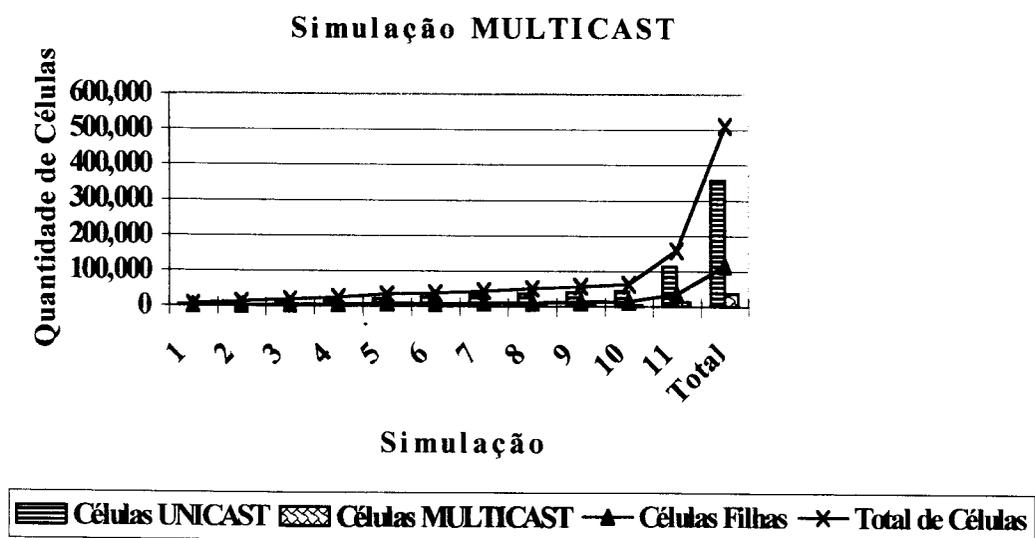
Embora a porcentagem de perdas de células tenha subido de 0,0005% para 0,1021%, o motivo detectado foi o mesmo, ou seja, uma quantidade excessiva de células com destino a mesma porta de saída no mesmo ciclo.

### 5.2.2 Simulação de células *multicast*

Nesta simulação, as células que percorrem a estrutura de roteamento atendem a solicitações tanto *unicast* quanto *multicast*, não havendo controle prévio dos dados gerados. Isto permitiu que múltiplos pedidos *multicast* fossem atendidos em um mesmo ciclo de simulação. É interessante observar, a grande quantidade de perdas de células ocasionado pelo aumento excessivo de células percorrendo a segunda parte da estrutura de roteamento (região de conversão), em função da grande quantidade de cópias de células *multicast* geradas nesta região.

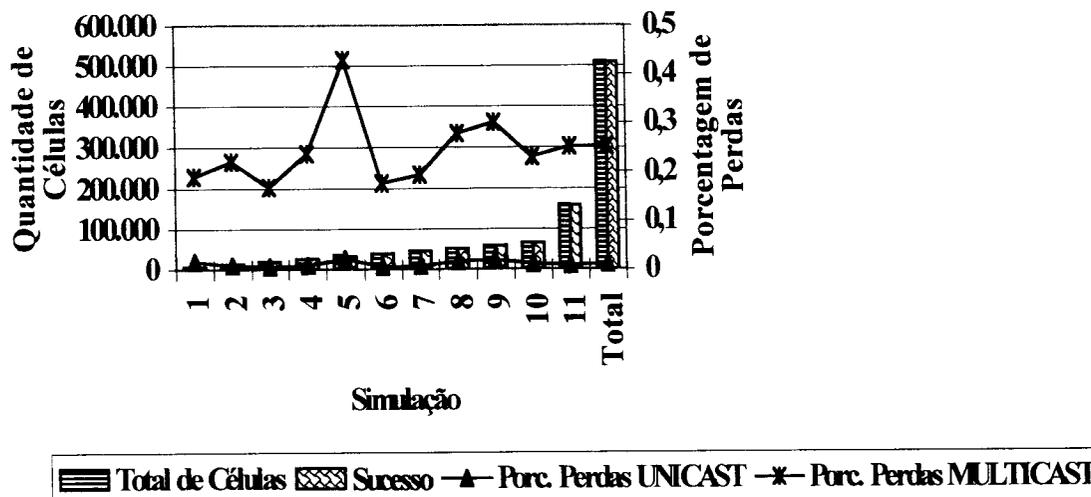
**Tabela 5.2:** Simulação multicast livre (sem controle do número de solicitações multicast por ciclo)

Simulação	Células Geradas	Células UNICAST	Células MULTICAST	Células Filhas	Total de Células	Sucesso	Perda UNICAST	Perda MULTICAST
1	16.000	4.438	465	1.429	6.332	6.319	1	12
2	32.000	8.883	955	2.903	12.741	12.712	1	28
3	48.000	13.452	1.395	4.231	19.078	19.045	1	32
4	64.000	17.829	1.885	5.652	25.365	25.303	2	60
5	80.000	21.628	2.989	8.995	33.612	33.461	7	144
6	96.000	26.812	2.831	8.477	38.120	38.051	2	67
7	112.000	31.309	3.259	9.794	44.362	44.273	3	86
8	128.000	35.768	3.760	11.344	50.872	50.723	8	141
9	144.000	40.331	4.205	12.702	57.238	57.055	11	172
10	160.000	44.707	4.682	14.058	63.447	63.294	7	146
11	400.000	111.778	11.707	34.960	158.445	158.034	13	398
Total	1.280.000	356.934	38.133	114.545	509.612	508.270	56	1.286



**Figura 5.15:** Simulação multicast livre de restrições quanto a entrada de dados

### Simulação MULTICAST



**Figura 5.16:** Porcentagem de perdas de células unicast e multicast em relação a quantidade total de células que deveriam atravessar a estrutura de roteamento

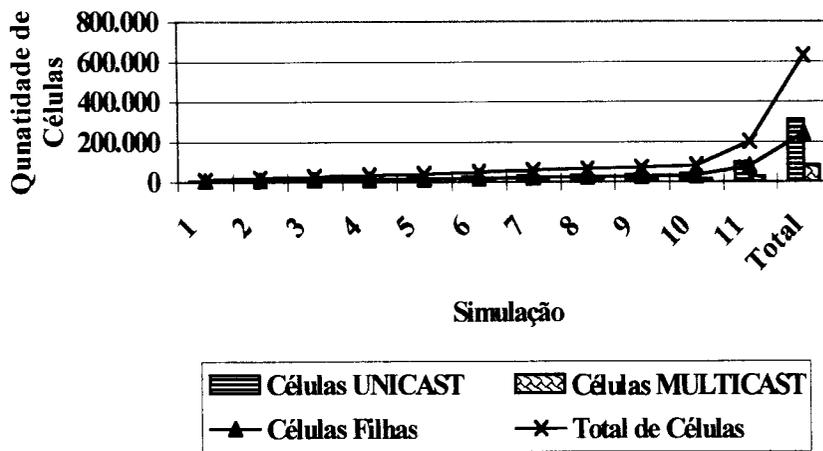
Apesar da quantidade de células perdidas ter crescido, esta não foi decorrente dos pedidos *multicast* de uma forma direta, e sim da quantidade excessiva de pedidos *multicast* em um mesmo ciclo. Para se ter uma idéia, no pior caso detectado nesta simulação, de 12 entradas ativas na estrutura de roteamento, 8 estavam destinadas a atender células *multicast*.

No segundo grupo de simulação *multicast*, foi imposta a condição de que apenas um pedido *multicast* fosse aceito a cada ciclo de simulação. Outro ponto importante foi a imposição de que todo ciclo de simulação válido (composto por ao menos uma célula com tipo diferente de "V"), necessariamente deveria conter uma solicitação *multicast*. Nenhuma restrição com respeito as portas de saída foi imposta. É interessante observar o aumento no número de células percorrendo a estrutura de roteamento e a redução de colisões tanto *unicast* quanto *multicast*.

**Tabela 5.4:** Simulação multicast limitando a uma solicitação multicast por ciclo

Simulação	Células Geradas	Células UNICAST	Células MULTICAST	Células Filhas	Total de Células	Sucesso	Perda UNICAST	Perda MULTICAST
1	16.000	3.867	998	3.021	7.886	7.886	0	0
2	32.000	7.728	1.994	6.014	15.736	15.736	0	0
3	48.000	11.456	2.988	8.985	23.429	23.429	0	0
4	64.000	15.599	3.993	11.998	31.590	31.590	0	0
5	80.000	19.455	4.981	14.976	39.412	39.412	0	0
6	96.000	23.281	5.986	17.952	47.219	47.219	0	0
7	112.000	27.170	6.974	21.042	55.186	55.186	0	0
8	128.000	30.958	7.979	23.920	62.857	62.857	0	0
9	144.000	34.542	8.973	26.874	70.389	70.389	0	0
10	160.000	38.732	9.974	29.972	78.678	78.678	0	0
11	400.000	96.793	24.921	74.923	196.637	196.635	2	0
Total	1.280.000	309.581	79.761	239.677	629.019	629.017	2	0

**Simulação MULTICAST**



**Figura 5.17:** Simulações multicast com a restrição de que toda entrada de células úteis deve conter um e apenas um pedido multicast.

### Simulação MULTICAST

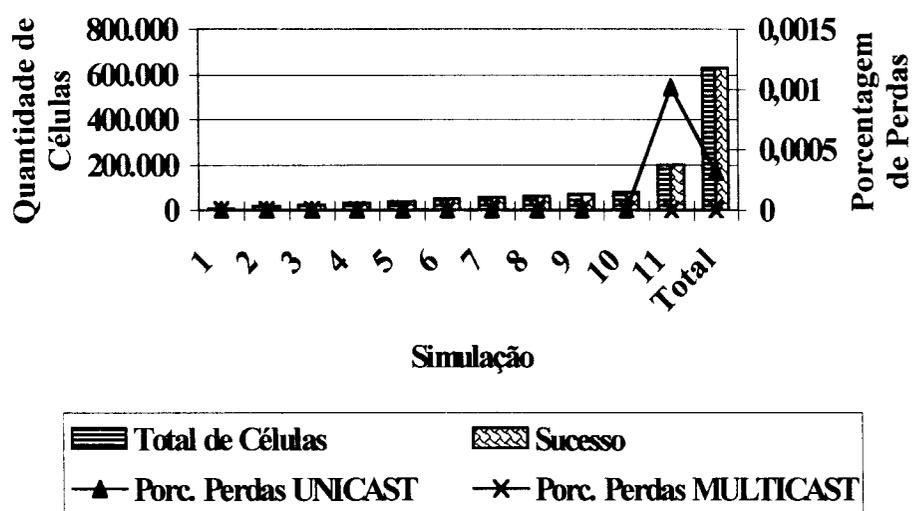


Figura 5.18: Representação da porcentagem de perdas unicast e multicast em função do total de células que deveriam percorrer a estrutura de roteamento

Esta simulação vem colaborar com as observações feitas na simulação anterior. Embora o número de células que percorrem a estrutura tenha crescido de 509.612 para 629.019, o total de células perdidas baixou de 1.342 para 2 células. E a outra constatação importante, é que as células *unicast* em função de sua vazão no primeiro estágio da estrutura de roteamento não interferem no roteamento das células *multicast*.

No capítulo 6 serão apresentadas as conclusões finais e também as propostas para trabalhos futuros.

## Capítulo 6

*Este capítulo apresenta as conclusões do trabalho e as propostas para trabalhos futuros.*

### **Conclusão e propostas de trabalhos futuros**

#### **6.1 Conclusão**

Os resultados da simulação indicam que a estrutura de roteamento, embora flexível, é ainda fortemente dependente dos dados de entrada.

As saídas de fuga nas colunas 3 e 4 permitem um intenso escoamento das células *unicast*. A cópia das células *multicast*, no segundo estágio da estrutura de roteamento, garante uma reduzida taxa de perdas, pouco influenciando no tráfego das células *unicast*.

O gerador randômico utilizado na geração dos dados de entrada, não condiz com a realidade do tráfego na qual uma chave ATM será submetida. Além disso, se mostrou “muito viciado” com relação a geração das portas de saída e o tipo de célula, o que de uma certa forma acabou sendo muito interessante pois, submeteu a estrutura de roteamento a testes muito mais rigorosos do que aqueles que se conseguiriam com os valores esperados na prática (distribuição geométrica segundo Urushidani et al 1998).

Um ponto interessante a ser considerado na estrutura de roteamento é o fato de ser ela toda construída com elementos simples. A ausência de *buffer*, além

de reduzir os custos, reduz também a complexidade dos mecanismos de gerenciamento e controle, aumentando o *throughput*.

É importante frisar, que a estrutura de roteamento é apenas um dos componentes de uma chave ATM. Os algoritmos de policiamento, mecanismos de controle de acesso a conexão (CAC), e os protocolos de gerenciamento, não foram levados em conta. A estrutura de roteamento proposta, foi submetida a tráfegos de células geradas, com um rigor muito acima daquele esperado na prática. Sendo assim, os resultados obtidos no capítulo 5 apontam para uma estrutura robusta e que, com muito pouco controle pode levar a taxa de perdas de células a valores muito próximos de zero, senão zero (que se pode constatar nas tabelas 5.1 e 5.4). Neste ponto, cabe ressaltar que as células descartadas são decorrentes do excesso de células solicitando a mesma porta de saída (acima de 4), e o excessivo número de pedidos *multicast* em um mesmo ciclo, casos estes, que seriam facilmente controlados com um mecanismo de policiamento simples, filtrando estes tipos de pedidos.

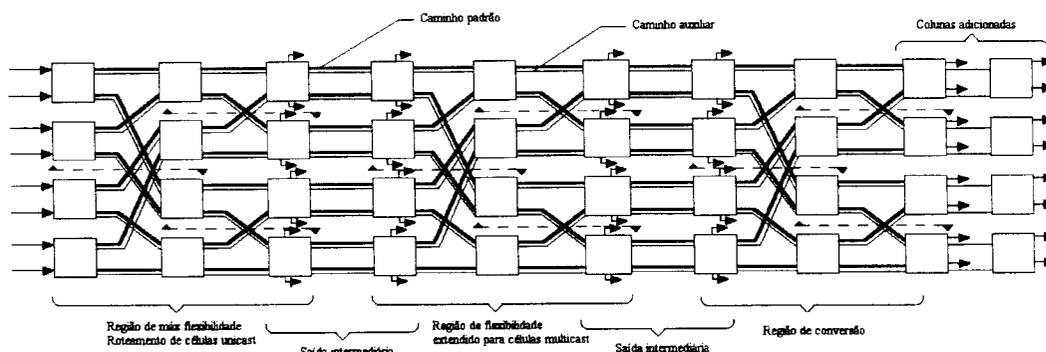
## 6.2 Propostas de trabalhos futuros

Um projeto interessante, seria o de um gerador de dados de entrada que atenda com mais precisão o comportamento do tráfego, tanto *unicast* como *multicast*, em uma rede ATM. Para isto, um ponto de partida seria utilizar um gerador que ao invés de uma distribuição randômica, respeitasse uma distribuição geométrica conforme a proposta de Urushidani et al (1998).

Outro trabalho interessante seria a implementação de um algoritmo de policiamento para fiscalizar o tráfego de células que chegam na estrutura de roteamento a cada ciclo.

Outra possibilidade seria a de modificar a estrutura de roteamento da seguinte forma: colunas 1 a 3 continuam como sendo a região de flexibilidade para o atender as células *unicast*, colunas de 4 a 6 deixariam de ser uma região de convergência (sem flexibilidade) e passariam a ser um segundo bloco de flexibilidade, agora aplicado a células *multicast*. As colunas de 7 a 9 passariam a ser a região de convergência. Para isto, será necessário adicionar duas colunas em

relação a proposta da figura 4.30, sendo uma para finalizar a rede de convergência e a outra para retirar da estrutura de roteamento, as células que chegaram a seu destino pelo caminho auxiliar. Esta proposta adicionaria apenas dois estágios de *hardware* e estenderia a flexibilidade obtida para células *unicast* às células *multicast*, permitindo que mais solicitações *multicast* sejam atendidas em um mesmo ciclo. As saídas de fuga continuariam a existir, só que agora distribuídas nas colunas (3 e 4) e (6 e 7) atendendo tanto a células *unicast* quanto *multicast*. Para uma melhor visualização desta proposta, observe a figura 6.1 a seguir.



**Figura 6.1:** Proposta de modificação na estrutura de roteamento

Como o ATM se presta ao tráfego de voz, dados e imagem, outro ponto interessante seria analisar os limites para os quais as taxas de perda de células começam a se tornar significativas em função do tipo de tráfego a que a chave será submetida. Tal análise contribuiria para a construção de chaves dedicadas a um custo muito menor e a consequente utilização, mais inteligente, dos recursos da rede.

## Referências Bibliográficas

- Awater, G. A., Schoute, F. C. *Optimal Queueing Policies for Fast Packet Switching of Mixed Traffic* IEEE Journal on Selected Areas in Communications, vol. 9, no. 3, April 1991.
- Bajaj, S. B. et al. *High-Speed ATM/SONET Infrastructure Research in ATDNet* IEEE Network, July/August 1996.
- Becker, D., Savary, H. F., Verhille, H. *Prospective Views on the Alcatel Broadband Architecture* Electrical Communication vol. 64, no. 2/3 – 1990.
- Boettle, D., Preiasach, H., Woelk, J. *A 16 x 16 switch element for 1.2 Gbit/s in 0.8  $\mu$ m BICMOS technology* – Ann. Télécommun., 48, no. 3-4, 1993.
- Boudec, Y., J. *The Asynchronous Transfer Mode: a tutorial* Computer Network and ISDN Systems 24 1992.
- Byun, J. W., Lee, T. T. *The Design and Analysis of an ATM Multicast Switch with Adaptive Traffic Controller* IEEE/ACM Transactions on Networking, vol. 2, no. 3, June 1994.
- Casale, S. et al. *Service Integration Issues on an ATM DQDB MAN* Computer Communications, vol. 17, no. 6, June 1994.
- Chan, K. S. et al. *Clos-Knockout: A Large-Scale Modular Multicast ATM Switch* IEICE Transactions Communications, vol. E81-B, no. 2, February 1998.
- Chang, C. Y., Paulraj, A. J., Kailath, T. *A Broadband Packet Switch Architecture with Input and Output Queueing* IEEE, 1994.
- Chao, H. J., Choe, B. S. *Design and Analysis of a Large-Scale Multicast Output Buffered ATM Switch* IEEE/ACM Transactions on Networking, vol. 3 no. 2 April 1995.
- Chao, H. J., Ghosal, D., Saha, D., Tripathi, S, K. *IP on ATM Local Area Networks* IEEE Communications Magazine, August 1994.
- Chen, T. M., Liu, S. S. *Management and Control Functions in ATM Switching Systems* IEEE Network, July/August 1994.
- Chen, T. M., Liu, S. S. *ATM Switching Systems* Artech House Boston London, 1995.

- Choi, J. S. *Design and Analysis of a Growable ATM Switch with Multicasting Capability* IEICE Transactions Communications, vol. E81-B, no. 2, February 1998.
- Choi, J. S., et al. *Design of a New Multicast Addressing Scheme for Self-Routing ATM Tree Networks* IEICE Transactions Communications, vol. E81-B, no. 2, February 1998.
- Comer, D. E. - *Internetworking with TCP/IP volume I* Prentice Hall Third Edition, 1995.
- Coppo, P. et al. *Optimal Cost/Performance Design of ATM Switches* IEEE/ACM Transactions on Networking, vol. 1, no. 5, October 1993.
- Deitel, H. M., Deitel, P. J. *C++ How to Program* Prentice Hall, Inc. A Simon & Schuster Company englewood cliffs, New Jersey, 1994.
- Dixon, R. C. *Cells-In-Frames: A System Overview* IEEE Network, July/August 1996.
- Eng, K. et al. *A Knockout switch for variable-length packets* IEEE Journal on Selected Areas in Communications, vol. 5, no. 9, December 1987.
- Eng, K. et al. *A Growable Packet (ATM) Switch Architecture: Design Principles and Applications* IEEE Transactions on Communications, vol. 40, no. 2, February 1992.
- Eng, K. et al. *Multicast and broadcast services in a Knockout packet switch* Infocom '88, New Orleans, March 1988.
- Eng., K. et al. *A Knockout switch for variable-length packets* ICC '87, Seattle, June 1987.
- Fischer, W. et al. *Data Communications Using ATM: Architectures, Protocols, and Resource Management* IEEE Communications Magazine, August 1994.
- Forcina, A. et al. *A Multicast Broadband Switching Module in a Hybrid ATM environment* ICC '89, Boston, June 1989.
- Friesen, V. J. et al. *Resource Management with Virtual Paths in ATM Networks* IEEE Network, September/October 1996.
- Giacomazzi, P. *Channel Grouping Techniques in the Tandem Banyan Switching Fabric* IEEE. 1994.

- Ginsburg, D. *ATM Solutions for Enterprise Internetworking* Addison-Wesley New York, 1996.
- Huang, A. *The relationship between STARLITE, a wideband digital switch and optics* ICC '86, Toronto, June 1986.
- Huang, A., Knauer, S. *Starlite: a wideband digital switch* Globecom '84, Atlanta, November 1984.
- Huang, N. C. *An Integrated Voice/Data Network Architecture Using Virtual Circuit Switching* IEEE Global Telecommunication Conference Globecom '85, New Orleans, Dec. 1985.
- Huang, N. C., Lea, C. T. *Architecture of a Time Multiplexed Switch* Globecom '86, Houston, November 1986.
- Huang, N. F. et al. *Some routing problems on Broadband ISDN* Computer Networks and ISDN Systems, 1994.
- Hui, J. *A broadband packet switch for multi-rate services* ICC '87, Seattle, June 1987.
- ITU International Telecommunication Union *Recommendation I.321* Geneva, 1991.
- Iwata, A. et al. *ATM Routing Algorithms with multiple QoS requirements for multimedia Internetworking* IEICE Transactions Communication, vol. E79B, no. 8, August 1996.
- Jung, Y. C., Un, C. K., Ryu, S. M., Lee, S. C. *Analysis of out-of-sequence problem and preventative schemes in parallel switch architecture for high-speed ATM network* IEE Proc, Commun., vol. 141, no. 1, February 1994.
- Karol, M. J., Yeh, Y. S. *A Growable Packet (ATM) Switch Architecture: Design Principles and Applications* IEEE Transactions on Communications, vol. 40, no. 2, February 1992.
- Katvenis, M. *buffer Requirements of Credit-Based Flow Control when a Minimum Draining Rate is Guaranteed* 4<sup>th</sup> IEEE Workshop on Architecture & Implementation of High Performance Communication Subsystems, HPC '97, Greece, June 1997.
- Kim, J. B., Suda, T., Yoshimura, M. *International Standardization of B-ISDN* Computer Networks, May 1994.

- Kinoshita, K., Saito, T. *ATM LAN Technologies for Integrated Business Communication Services* NTT Review, vol. 6, no. 1, January 1994.
- Lagerstedt, B., Nyman, H. *ATM in Public Telecommunications Network* Ericsson Review N° 3, 1993.
- Lee, D. S., Sengupta, B. *Queueing Analysis of a Threshold Based Priority scheme For ATM Networks* IEEE/ACM Transactions on Networking, vol. 1, no. 6, December 1993.
- Liew, S. C. *Performance of Various Input-buffered and Output-buffered ATM Switch Design Principles under Bursty Traffic: Simulation Study* IEEE Transactions on Communications, vol. 42, no. 2/3/4, February/March/April 1994.
- Lin, A. Y. *On the Performance of an ATM Switch with Multichannel Transmission Groups* IEEE Transactions on Communications, vol. 41, no. 5, May 1993.
- Lin, H. T., Hughes, H. D. *Performance evaluation of priority buffer management in ATM switches* Computer Systems Science & Engineering, November 1997.
- Lin, Y. S., Shung, B. *An Efficient Architecture for Multicasting in Shared Buffer ATM Switches* IEICE Transactions Communications, vol. E81-B, no. 2, February 1998.
- McDysan, D. E., Spohn D. L. *ATM Theory and Application* McGraw-Hill Series on Computer Communications, 1994.
- Meyer, J. F., Montagna, S., Paglio, R. *Dimensioning of an ATM switch with shared buffer and threshold priority* Computer Networks and ISDN Systems vol. 26, 1993.
- Miyaho, N., Itoh, A., Shiimoto, K. *ATM Node System Technology for Effective Maintainability* IEICE Trans. Communications, vol. E79-B, no. 12, December 1996.
- Miyaho, N., Miura, A. *Integrated Switching Architecture and Its Traffic Handling Capacity in Data Communication Networks* IEICE Trans Commu., vol. E79-B, no., 12 December 1996
- Monteiro, J. A. S., Frazão, J. A. J. *On the Efficiency of Policing Mechanisms for ATM Networks* IEEE Proceedings, 1994.

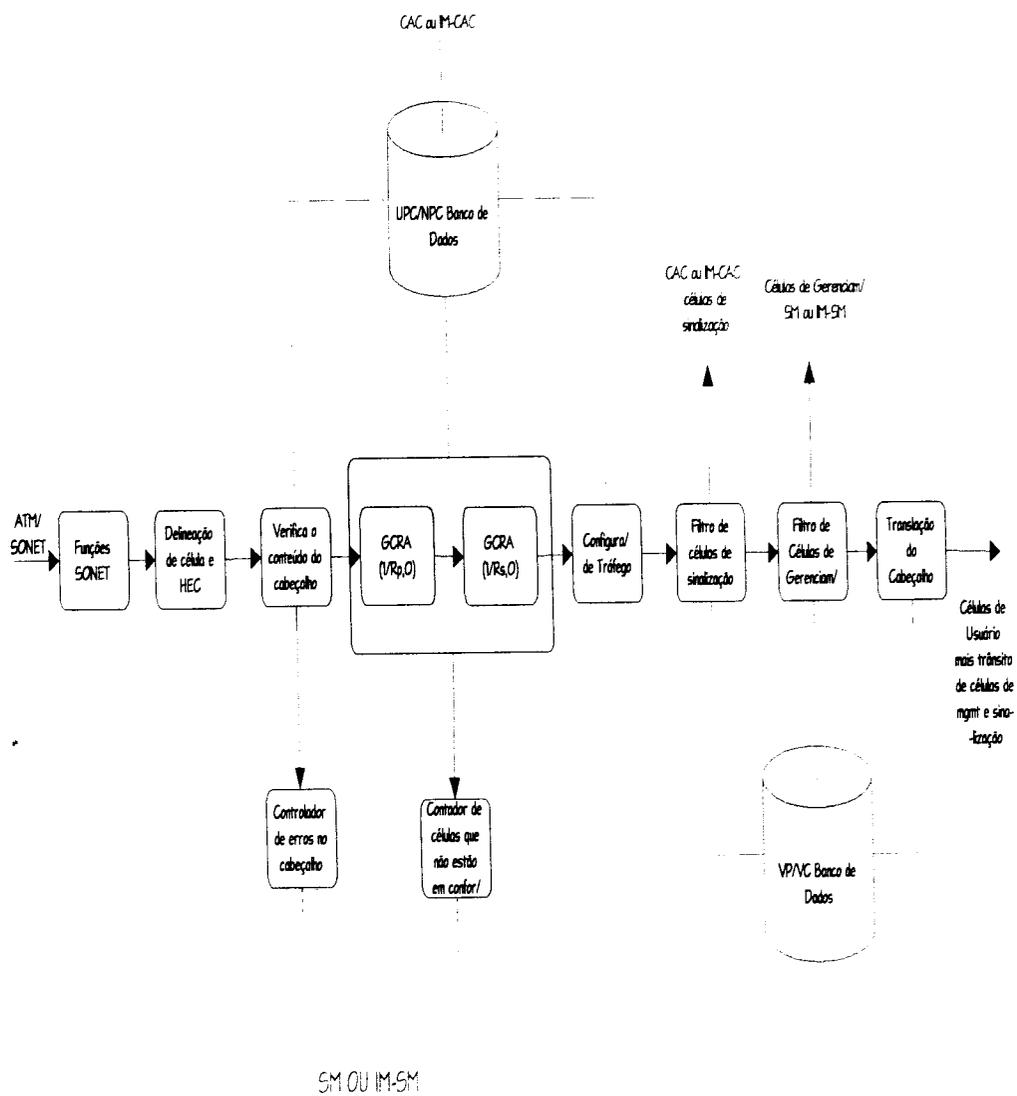
- Obara, H. *Design of a Multistage Self-Routing Switch with a Distributed Cell Sequence Control* *Electronic and Communication in Japan*, part 1, vol. 73, no. 10, 1990.
- Park, H. S., Kwak, D. Y., Rhee, W. S., Jeon, M. Y., Kim, J. K. *Global Traffic Control in ATM Networks* *IEICE Trans Commu.*, vol. E78-B, no. 4, April 1995.
- Pattaviana, A., Bruzzi G. *Analysis of Input and Output Queuing for Nonblocking ATM Switches* *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, June 1993.
- Pitts, J. M., Schormans, J. A. *Introduction to ATM Design and Performance* John Wiley & Sons, 1996.
- Prycker, M. de, Bauwens, J. *A Broadband central Office for ATM* *International Journal of Digital and Analog Cabled System*, May 1987.
- Prycker, M. de, Bauwens, J. *A Switching Exchange for an Asynchronous time Division based Network* *ICC '87*, Seattle, June 1987.
- Pryckerm M. D. *Asynchronous Transfer Mode* - Ellis Horwood, 1993.
- Shimokoshi, K., Shiraishi, Y. *A Simulation Study on LAN/MAN Interconnection with an ATM Network* *IEICE Transaction Communication*, vol. E78-B, no. 3, March 1995.
- Skelly, P. et al. *A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer* *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, August 1993.
- Staalhagen L. *A Comparison Between the OSI Reference Model and the B-ISDN Protocol Reference Model* *IEEE Network*, January/February 1996.
- Stroustrup, B., Ellis, M. A. *C<sup>++</sup> Reference Manual* Addison Wesley, ANSI Base Document, 1990.
- Suh, J., Jun, C. H. *Performance analysis of Knockout switch with input buffers* *IEE Proc. Communication*, vol. 141, no. 3, June 1994.
- Tagle, P. U., Sharma, N. K. *Multicast Packet Switch Based on Dilated Network* *IEICE Transactions Communications*, vol. E81-B, no. 2, February 1998.
- Tanenbaum, A. S. *Redes de Computadores* Editora Campus, 1994.
- The International Telegraph and Telephone Consultative Committee *Recomendation I.321* Geneva, 1991.

- Tierney, B.L., Johnston, W. E., Lee, J. R., Hoo, G. *Performance Analysis in High-speed Wide Area IP-over-ATM Networks: Top-to-Bottom End-to-End Monitoring* IEEE Network, May/June 1996.
- Travieso, G. *Introdução a C++* Universidade de São Pulo, Instituto de Física de São Carlos, 1999.
- Turner, J. *Design of a Broadcast Packet Switching Network* IEEE Transactions on Communications, vol. 36, no. 6, June 1988.
- Turner, J. S. *Design of a Broadcast Packet Network* IEEE Annual Joint Conference in Computer Communications 1986.
- Turner, J., Yamanaka, N. *Architerural Choices in Large Scale ATM Switchs* IEICE Transactions Communications, vol. E81-B, no. 2, February 1998.
- Urushidani, S. et al. *A High-Performance Multicast Switch and Its Feasibility Study* IEICE Transactions Communications, vol. E81-B, no. 2, February 1998.
- Wright, D. J. *Voice over ATM: An Evaluation of Network Architecture Alternatives* IEEE Network, September/October 1996.
- Yeh, Y. S. et al. *The Knockout Switch: a simple, modular architecture for high-performance packet switching* IEEE Journal on Selected Areas in Communications, vol. 5, no. 8, October 1987.
- Zhong, W., Shimazu, Y., Tsukuda, M., Yusimatsu, K. *A Modular Tbit/s TDM-WDM Photonic ATM Switch Using Opitcal Output Buffers* IEICE Trans. Communications, vol. E77-B, n° 2 February 1994.

# Apêndice A

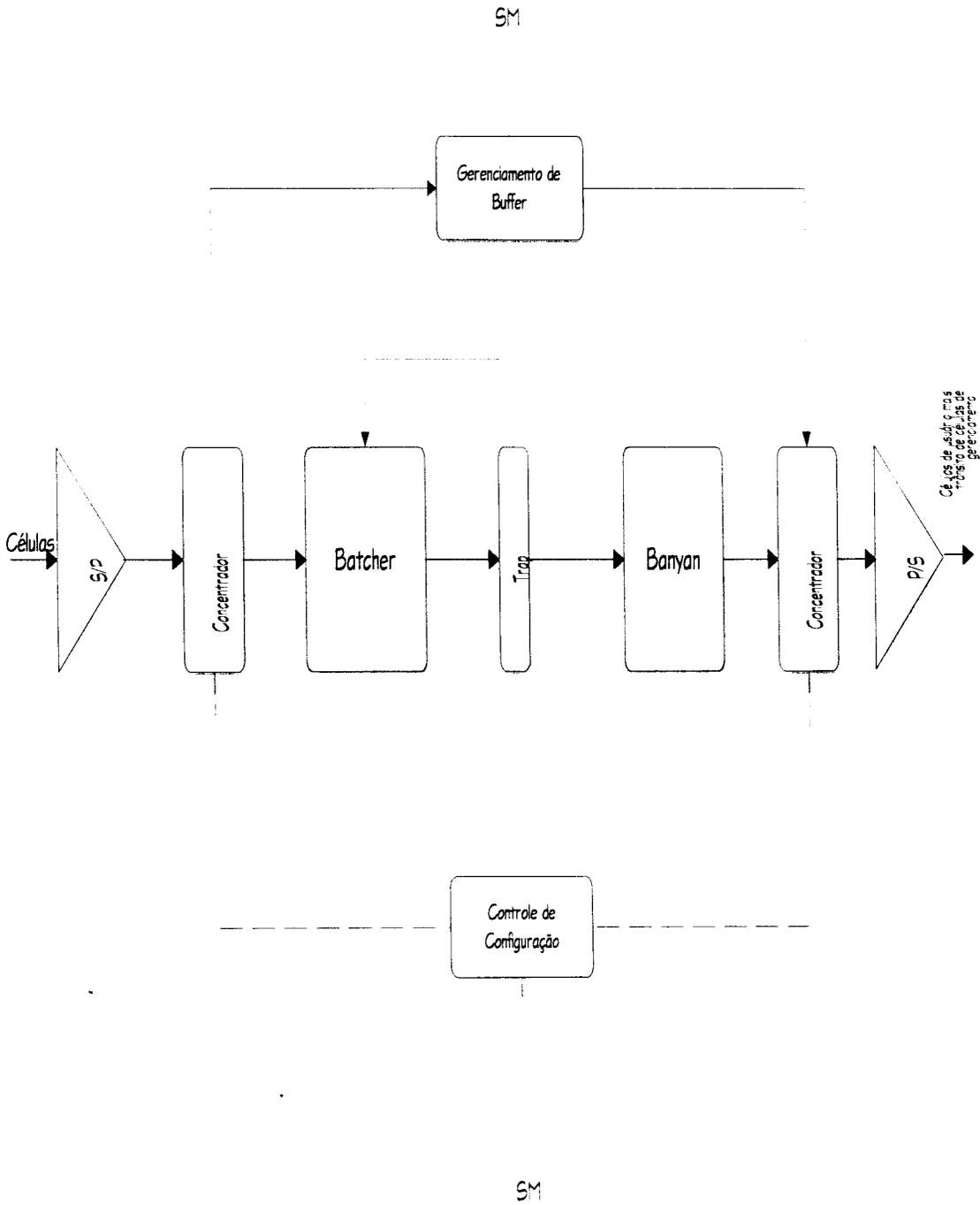
## Arquitetura padrão de uma chave ATM

### Módulo de Entrada:

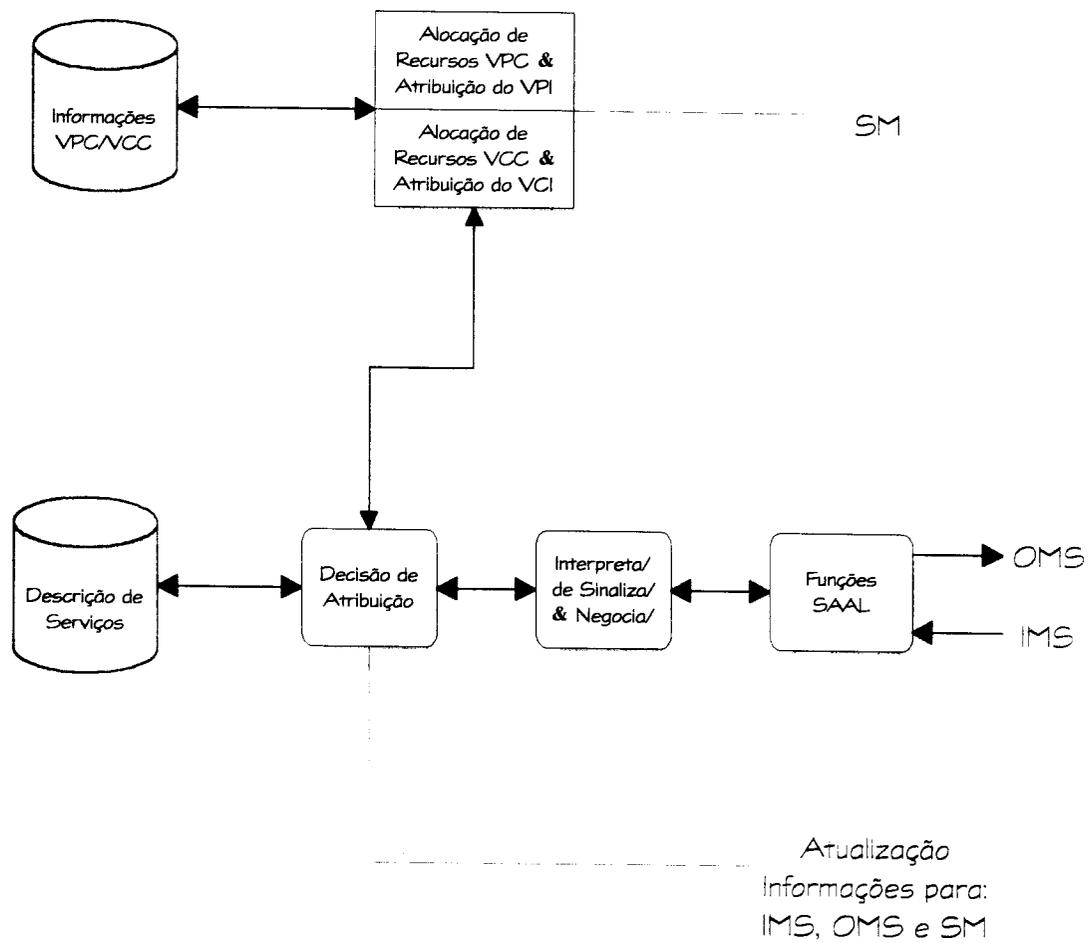




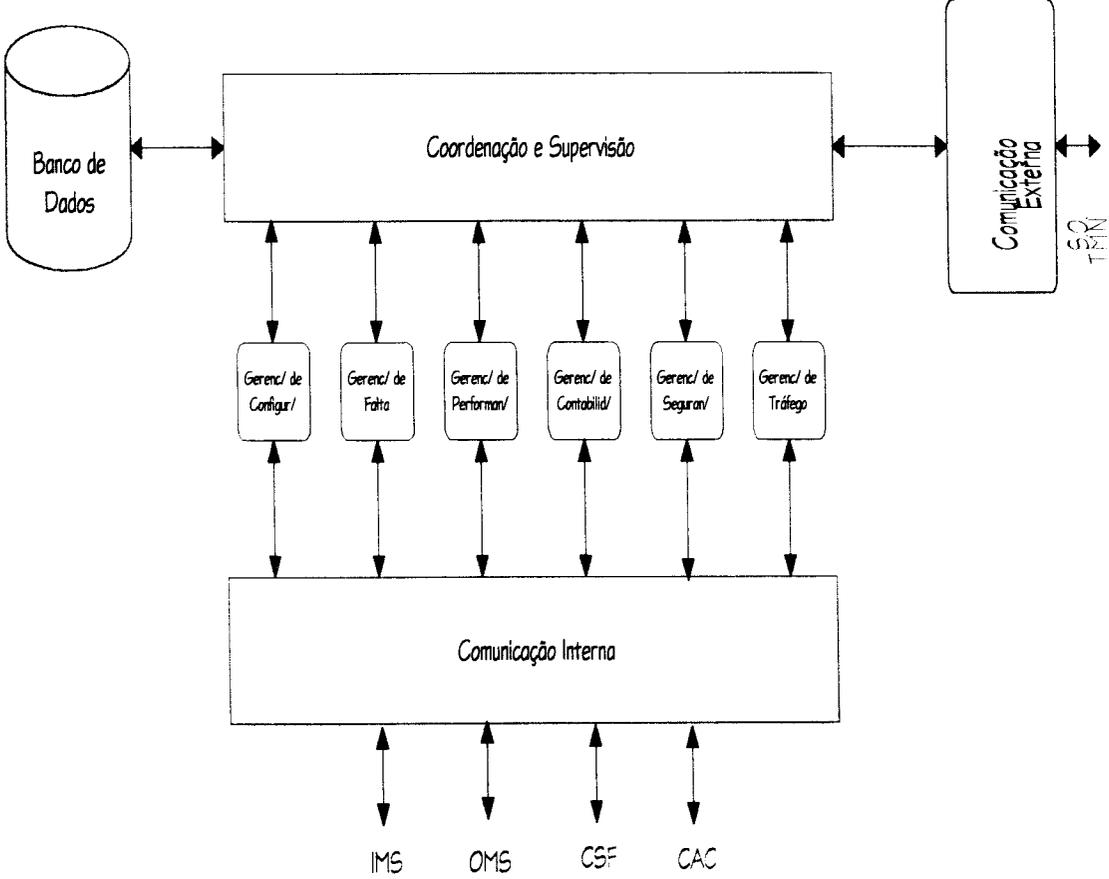
**Módulo de Roteamento:**



## Módulo de Controle e Admissão da Conexão:



**Módulo de Gerenciamento:**



## Glossário

**Nota:** Para um glossário completo sobre ATM consultar em:

[www.atmforum.com/atmforum/library/glossary/gloss-print.html](http://www.atmforum.com/atmforum/library/glossary/gloss-print.html)

<b>AAL</b>	- <i>ATM Adaptation Layer</i> – Camada de Adaptação ao ATM
<b>ABR</b>	- <i>Available Bitrate</i> Taxa de Bits Disponíveis
<b>ANSI</b>	- <i>American National Standards Institute</i> – Instituto Nacional Americano de Padrões
<b>API</b>	- <i>Application Program Interface</i> – Interface de Programa Aplicativo
<b>ARP</b>	- <i>Address Resolution Protocol</i> – Protocolo de Resolução de Endereços
<b>ASE</b>	- <i>Application Service Element</i> - Elemento de Serviço de Aplicação
<b>ASO</b>	- <i>Application Service Object</i> - Objeto de Serviço de Aplicação
<b>ATM</b>	- <i>Asynchronous Transfer Mode</i> - Modo de Transferência Assíncrono
<b>ATMARP</b>	- <i>ATM Address Resolution Protocol</i> – Protocolo de resolução de endereços definido pela RFC 1577, baseado no ARP.
<b>B-ISDN</b>	- <i>Broadband - ISDN</i> - ISDN de Banda Larga
<b>BISUP</b>	- <i>Broadband ISDN User's Part</i> – Um protocolo SS7, que define as mensagens de sinalização para serviços de controle da conexão
<b>Blocking</b>	- Diz-se da <i>switch</i> que oferece recursos em quantidade inferior ao mínimo necessário para permitir conexões com todas as entradas suportadas e as saídas disponíveis
<b>Bridge</b>	- Dispositivo para conexão de redes físicas que opera na camada de enlace de dados no modelo OSI
<b>Buffer</b>	- Armazenamento temporário de informações
<b>CAC</b>	- <i>Connection Admission Control</i> – Controle de Admissão de Conexão, é definido como um conjunto de ações realizadas pela rede durante a fase de <i>set-up</i> para determinar se a conexão solicitada pode ser aceita ou se ela deve ser recusada

<b>CBR</b>	- <i>Constant Bitrate</i> - Taxa de Bits Constantes
<b>CCITT</b>	- <i>International Consultative Committee for Telephone and Telegraph</i> - Comitê Consultivo Internacional de Telegrafia e Telefonia
<b>CCR</b>	- <i>Commitment, Concurrency and Recovery</i> - Segurança, Concorrência e Recuperação
<b>CE</b>	- <i>Connection Endpoint</i> - Ponto final da Conexão
<b>Cell Relay</b>	- Tipo de tecnologia de transmissão de dados que, como o ATM, se baseia na transmissão e comutação de unidades de informação de tamanho fixo
<b>Células idle</b>	- Células transmitidas, na ausência de informação útil a transmitir, para manutenção do sincronismo
<b>CF</b>	- <i>Coordination Function</i> - Função de Coordenação
<b>CLP</b>	- <i>Cell Loss Priority</i> - Prioridade de Perda de Células
<b>Comutação</b>	- Operação realizada pelas chaves, onde a informação recebida em uma de suas entradas é dirigida à saída correta
<b>CP</b>	- <i>Common Part</i> - Parte Comum
<b>CPCS</b>	- <i>Common Part Sublayer Convergence</i> - Parte Comum da Subcamada de Convergência
<b>CS</b>	- <i>Convergence Sublayer</i> - Sub-camada de Convergência
<b>DS-n (0,1,2,3)</b>	- Formato de transmissão de sinais digitais – padrão americano – operando à taxa básica de 44,736 Mb/s, com capacidade para suportar até 672 canais de 64 kb/s
<b>E.164</b>	- Formato de endereços públicos especificado pela norma E.164 da ITU-T
<b>ELAN</b>	- <i>Emulated LAN</i>
<b>ETSI</b>	- <i>European Telecommunications Standards Institute</i> – Instituto Europeu de Padrões em Telecomunicações
<b>Fabric</b>	- Malha de interconexão entre as portas de uma chave. Módulo central da estrutura de roteamento
<b>FDDI</b>	- <i>Fiber Distributed Data Interface</i> – Interface de Dados Distribuídos por Fibra. Padrão de comunicação de dados, adotado

pelo ANSI que especifica uma rede em anel baseada em fibra operando a uma taxa de 100 Mb/s

<b>Frame</b>	- Normalmente associado a camada de enlace de dados do modelo de referência OSI. Segundo padrões pré definidos, representa um conjunto de octetos
<b>FTTC</b>	- <i>Fiber to The Curb</i> – Fibra até a Calçada. Tecnologia de infraestrutura para distribuição de sinais de banda larga até residências utilizando fibra e cobre
<b>Gateway</b>	- Dispositivo que promove a integração de redes operando sob protocolos distintos
<b>GFC</b>	- <i>Generic Flow Control</i> - Controle de Fluxo Genérico
<b>HDTV</b>	- <i>High Definition Television</i> – Televisão de Alta Definição. Padrão de transmissão de TV caracterizado pelo maior número de pontos usados na composição da imagem
<b>HEC</b>	- <i>Header Error Correction</i> - Correção de Erros de Cabeçalho
<b>HFC</b>	- <i>Hybrid Fiber Coax</i> – Híbrido Fibra Coaxial. Tecnologia de infraestrutura de distribuição de sinais até residências que utiliza cabos óticos e coaxiais
<b>ILMI</b>	- <i>Interim Local Management Interface</i> – Interface Interinapara Gerenciamento Local. Especificação proposta pelo ATM Forum visando às atividades de gerenciamento em redes ATM
<b>InARP</b>	- <i>Inverse Address Resolution Protocol</i> – Protocolo de resolução de endereços da família TCP/IP, definido pelo RFC 1293
<b>InATMARP</b>	- <i>Inverse ATM Address Resolution Protocol</i> – Protocolo de resolução de endereços definido pelo RFC 1577, baseado no InARP
<b>Interworking</b>	- Interoperação – Capacidade de comunicação e interação cooperativa entre redes de tecnologias distintas
<b>IP</b>	- <i>Internet Protocol</i> – Protocolo da Internet
<b>ISDN</b>	- <i>Integrated Service Digital Network</i> - Rede Digital de Serviços Integrados

<b>ITU-T</b>	- <i>International Telecommunication Union - Technology</i> - União Internacional de Telecomunicações - Tecnologia
<b>LIS</b>	- <i>Logical IP Subnetwork</i> – Sub-rede Lógica IP. Conjunto de estações IP que compartilham um mesmo endereço de rede ou sub-rede
<b>LLC/SNAP</b>	- <i>Logical Link Control/SubNetwork Access Protocol</i> – Método de encapsulamento de pacotes em que um cabeçalho padrão é adicionado a cada pacote encapsulado (cabeçalho LLC/SNAP)
<b>MAC</b>	- <i>Medium Access Control Sublayer</i> – Subcamada de Controle de Acesso ao Meio. Corresponde a parte inferior da camada de enlace do modelo de referência OSI, com função de controle de acesso ao meio compartilhado nas LANs
<b>MIB</b>	- <i>Management Information Base</i> – Base de Dados de Gerenciamento, usada por protocolos de gerenciamento
<b>MPOA</b>	- <i>Multiprotocol over ATM</i> – Multiprotocolo sobre ATM. Ainda em fase de padronização pelo ATM Forum, para operação de protocolos em modo nativo
<b>Multicast</b>	- Conexão onde a informação gerada em um elemento integrante do grupo previamente definido é transmitida a todos os outros elementos desse grupo
<b>N-ISDN</b>	- <i>Narrowband - ISDN</i> - ISDN de Banda Estreita
<b>NNI</b>	- <i>Network Node Interface</i> - Interface dos Nós da Rede
<b>OSI</b>	- <i>Open Systems Interconnection</i> - Interconexão de Sistemas Abertos
<b>Payload</b>	- Conjunto de octetos que contém informações úteis ao usuário final
<b>PCI</b>	- <i>Protocol Control Information</i> - Informação de Controle do Protocolo
<b>PDU</b>	- <i>Protocol Data Unit</i> - Unidade de Dados de Protocolo
<b>PM</b>	- <i>Physical Media</i> - Meio Físico
<b>PNNI Fase 1</b>	- Primeira fase do padrão NNI para redes privadas, elaborado pelo ATM Forum
<b>PT</b>	- <i>Payload Type</i> - Tipo de Dados

<b>OAM</b>	- <i>Operation Administration and Maintenance</i> – um conjunto de funções de gerenciamento de rede, que provê indicações de falta, informações de desempenho, e diagnóstico dos dados
<b>PVC</b>	- <i>Permanent Virtual Circuit</i> – link com uma rota estática definida com antecedência, normalmente através de ligação manual
<b>QoS</b>	- <i>Quality of Service</i> - Qualidade de Serviço
<b>Queuing</b>	- Armazenamento temporário de informações em rigorosa ordem de chegada
<b>RFC</b>	- <i>Request for Comments</i> – Designação comum de um conjunto de documentos desenvolvidos sob o controle do IETF que contém padrões, técnicas, observações, procedimentos e especificações relativos à família de protocolos TCP/IP
<b>ROSE</b>	- <i>Remote Operations Service Element</i> - Elemento de Serviço de Operação Remota
<b>Roteador</b>	- Dispositivo que opera na camada de rede do modelo OSI. Responsável pela tomada de decisões sobre os caminhos a serem seguidos pelo tráfego na rede
<b>RTSE</b>	- <i>Reliable Transfer Service Element</i> - Elemento de Serviço de Transferência Confiável
<b>SAP</b>	- <i>Service Acces Point</i> - Ponto de Acesso a Serviço
<b>SAR</b>	- <i>Segmentation and Reassembly</i> - Segmentação e Remontagem
<b>SDH</b>	- <i>Synchronous Digital Hierarchy</i> - Hierarquia Digital Síncrona
<b>SDU</b>	- <i>Service Data Units</i> - Unidades de Dados de Serviço
<b>SNAP</b>	- <i>Subnetwork Attachment Point</i> - Ponto de Acesso a Sub-rede
<b>SONET</b>	- <i>Synchronous Optical Network</i> - Rede Óptica Síncrona
<b>SS</b>	- <i>Service Specific</i> - Serviço Específico
<b>SSCS</b>	- <i>Service Specif Convergence Sublayer</i> - Serviço Especifico da Sub-camada de convergência
<b>Switches</b>	- Dispositivos em uma rede de comunicações destinados à comutação
<b>TAXI</b>	- <i>Transparent Asynchronous Transmitter Receiver Interface</i> – Especificação para transmissão de sinais digitais em fibras óticas,

adotado originalmente no âmbito do padrão FDDI e posteriormente assumido pelo ATMForum como um dos possíveis padrões para a camada física no modelo ATM

<b>TC</b>	- <i>Transmission Convergence</i> - Convergência de Transmissão
<b>TE</b>	- Terminal Equipment – Equipamento terminal do usuário
<b>UNI</b>	- <i>User Network Interface</i> - Interface Rede Usuário
<b>VBR</b>	- <i>Variable Bitrate</i> - Taxa de Bits Variáveis
<b>VC</b>	- <i>Virtual Connection</i> - Conexão Virtual
<b>VCI</b>	- <i>Virtual Channel Identifier</i> - Identificador de Conexão Virtual
<b>VOD</b>	- <i>Video on Demand</i> – Vídeo sob Demanda. Serviço de difusão de filmes em vídeo no qual o usuário pode determinar quando e que filmes assistir
<b>VP</b>	- <i>Virtual Path</i> - Caminho Virtual
<b>VPI</b>	- <i>Virtual Path Identifier</i> - Identificador de Caminho Virtual
<b>Workgroup</b>	- Conceito relacionado a organização do trabalho, em que um grupo de pessoas geograficamente dispersas atuam cooperativamente para a execução de determinada tarefa