

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Simulação  
do Protocolo de Adaptação  
ATM Tipo 2 (AAL2)**

por

DORIAN AMORIM

Dissertação submetida à avaliação, como requisito parcial para  
a obtenção do grau Mestre em  
Ciência da Computação

Prof. Juergen Rochol  
Orientador

Porto Alegre, março de 2001.

**CIP – CATALOGAÇÃO NA PUBLICAÇÃO**

Amorim, Dorian

Simulação do Protocolo de Adaptação ATM Tipo 2 (AAL2) / por Dorian Amorim. - Porto Alegre: PPGC da UFRGS, 2001.

127f. : il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2001. Orientador: Rochol, Juergen.

1. Protocolo AAL2. 2. Simulação de redes. 3. Modelagem de tráfego de voz. 4. Redes ATM. I. Rochol, Juergen. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Henneman

Pró-Reitor Adjunto de Pós-Graduação: Prof. Philippe Olivier Alexandre Navaux

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## *Agradecimentos*

Gostaria de deixar meus sinceros agradecimentos a todos que de alguma forma contribuíram para o êxito deste trabalho, e em especial:

- a Deus, por sua força e sabedoria, que ilumina a todos durante nossas vidas;
- à minha querida esposa Renata, pela estímulo e confiança, mas acima de tudo pela compreensão e paciência durante esta longa jornada.
- ao meu orientador, Prof. Juergen Rochol, por ser prestativo e compreensivo, e pela orientação e oportunidade de desenvolvimento desta dissertação de mestrado;
- a todos os professores do PPGC da UFRGS com quem tive contato, pelo conhecimento que me passaram;
- à UDESC, na pessoa do magnífico reitor Raimundo Zumblick, pelo apoio e por proporcionar essa oportunidade de qualificação profissional;
- a meus familiares, pelo apoio durante este período;
- aos meus colegas de trabalho, Cinara, Luciano e Julibio, pelo incentivo, força e orientação no desenvolvimento deste trabalho;
- aos meus colegas de mestrado, pelo prazer de suas amizades, conversas e trocas de conhecimentos.

## *Sumário*

<b>Lista de Abreviaturas.....</b>	<b>6</b>
<b>Lista de Figuras .....</b>	<b>9</b>
<b>Lista de Tabelas.....</b>	<b>10</b>
<b>Resumo.....</b>	<b>11</b>
<b>Abstract.....</b>	<b>12</b>
<b>1 Introdução.....</b>	<b>13</b>
<b>2 A Tecnologia ATM.....</b>	<b>16</b>
2.1 O que é ATM.....	16
2.2 Origem do ATM .....	16
2.3 Descrição do ATM.....	17
2.4 Funcionamento do ATM.....	21
2.5 As Camadas de Adaptação ATM.....	24
2.5.1 Descrição do AAL.....	24
2.6 O Protocolo AAL 2 .....	26
2.6.1 O Histórico da Padronização.....	27
2.6.2 Primitivas de Comunicação .....	27
2.6.3 Descrição Geral .....	29
2.6.4 Formatação e Codificação da Subcamada CPS .....	31
2.6.5 Processamento da Subcamada CPS.....	33
2.6.6 Transmissor CPS.....	34
2.6.7 Receptor CPS.....	37
2.6.8 A Subcamada de Convergência Específica do Serviço (SSCS).....	39
<b>3 QoS e o Tipo das Aplicações.....</b>	<b>42</b>
3.1 Qualidade de Serviço (QoS).....	42
3.2 As Categorias de Serviço.....	44
3.3 Plano de Gerência do ATM .....	46
3.4 Conexões a Redes Locais.....	49
<b>4 Modelos de Fontes de Tráfegos VBR .....</b>	<b>51</b>
4.1 A Representação Matemática dos Modelos.....	51
4.2 Tipos de Modelos de Tráfego.....	53
4.2.1 Processo de Poisson .....	53
4.2.2 Processo de Bernoulli.....	53
4.2.3 Processo de Markov .....	54
4.2.4 Modelo ON/OFF .....	54
4.2.5 Processo de Markov Modulado.....	55
4.2.6 Processo de Poisson Modulado por Markov - MMPP.....	55
4.2.7 Processo de Poisson Interrompido - IPP .....	56

4.2.8	<i>Processo de Bernoulli Interrompido - IBP</i> .....	56
4.2.9	<i>Outros Modelos e Estudos</i> .....	57
<b>4.3</b>	<b>Simulação da Fonte de Tráfego</b> .....	<b>58</b>
4.3.1	<i>Criação de Fontes de Tráfego</i> .....	59
<b>5</b>	<b>Simulação do AAL2</b> .....	<b>60</b>
<b>5.1</b>	<b>Simulação de Redes</b> .....	<b>60</b>
5.1.1	<i>Comentários</i> .....	60
5.1.2	<i>Definições e Conceitos</i> .....	60
5.1.3	<i>Comparação com outras Soluções</i> .....	61
5.1.4	<i>Problemas e Desafios</i> .....	62
<b>5.2</b>	<b>Aquisição e Análise dos Resultados</b> .....	<b>63</b>
<b>5.3</b>	<b>A Escolha do Simulador</b> .....	<b>67</b>
<b>5.4</b>	<b>Descrição do simulador</b> .....	<b>68</b>
5.4.1	<i>O Tempo e o Gerenciador de Eventos</i> .....	69
5.4.2	<i>Rotinas de Tratamento de Eventos</i> .....	71
5.4.3	<i>Conexão entre os Componentes e as Rotas</i> .....	72
5.4.4	<i>O Registro das Saídas</i> .....	72
<b>5.5</b>	<b>Criação do Componente AAL2</b> .....	<b>73</b>
5.5.1	<i>Modificando as Aplicações</i> .....	75
5.5.2	<i>Mudando os Pacotes e as Rotas</i> .....	75
5.5.3	<i>O Algoritmo do AAL2</i> .....	76
5.5.4	<i>Verificação do Módulo</i> .....	77
<b>6</b>	<b>Teste e Validação</b> .....	<b>80</b>
<b>6.1</b>	<b>Técnicas de Verificação e Validação</b> .....	<b>80</b>
<b>6.2</b>	<b>A Verificação do Simulador</b> .....	<b>83</b>
<b>6.3</b>	<b>Artigos sobre o AAL2</b> .....	<b>85</b>
6.3.1	<i>Comentários sobre a Telefonía Celular</i> .....	85
6.3.2	<i>A Eficiência do Protocolo AAL2</i> .....	87
6.3.3	<i>Comparação entre o AAL1 e o AAL2</i> .....	88
6.3.4	<i>O Relógio de Uso Combinado (Timer_CU)</i> .....	90
<b>6.4</b>	<b>A Comparação para Validação</b> .....	<b>92</b>
<b>7</b>	<b>Conclusão</b> .....	<b>99</b>
<b>Anexo 1</b>	<b>Exemplos da Recomendação I.363.2 para a Segmentação e Multiplexação do AAL2</b> .....	<b>101</b>
<b>Anexo 2</b>	<b>Diagramas SDL do Módulo AAL2 criado para o Simulador NIST</b> .....	<b>104</b>
<b>Anexo 3</b>	<b>Exemplos de Arquivos do Simulador</b> .....	<b>115</b>
<b>Anexo 4</b>	<b>Modelo do AAL2 com Buffer</b> .....	<b>119</b>
<b>Anexo 5</b>	<b>Módulo de Inferência Estatística</b> .....	<b>123</b>
	<b>Bibliografia</b> .....	<b>126</b>

## *Lista de Abreviaturas*

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ABT	ATM Block Transfer
ACR	Available Cell Rate
ATM	Asynchronous Transfer Mode
ATC	ATM Transport Capability
ATM-PDU	ATM Protocol Data Unit
AS	ATM Switch
AUU	ATM-User-to-ATM-User Indication
B-ISDN	Broadband-Integrated Service Data Network
BER	Bit Error Rate
BT	Burst Tolerance
BTE	Broadband Terminal Equipment
CAC	Connection Admission Control
CBR	Constant Bit Rate
CBS	Cell Burst Size
CDV	Cell Delay Variation
CDVT	Cell Delay Variation Tolerance
CER	Cell Error Rate
CI	Congestion Indication
CID	Channel Identifier
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CPCS	Common Part Convergence Sublayer
CPS	Common Part Sublayer
CPS-INFO	CPS Interface Data
CPS-PDU	CPS Protocol Data Unit
CPS-PH	CPS Packet Header
CPS-PP	CPS Packet Payload
CPS-SDU	CPS Service Data Unit
CPS-UUI	CPS User-to-User Indication
CRC	Cyclic Redundancy Check
CS	Convergence Sublayer

CTD	Cell Transfer Delay
DSU	Data Service Unit
DTE	Data Terminal Equipment
FDDI	Fiber Distributed Data Interface
FEC	Forward Error Correction
GFC	Generic Flow Control
HEC	Head Error Control
IETF	Internet Engineering Task Force
ILMI	Interim Local Management Interface
ISDN	Integrated Service Data Network
ITU	International Telecommunication Union
ITU-T	ITU- Telecommunication Standardization Sector
LAN	Local Area Network
LI	Length Indication
LM	Layer Management
LSB	Least Significant Bit
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MMPP	Markovian Modulated Poisson Process
NNI	Network Network Interface
OAM	Operating Administration and Maintenance
OSF	Offset Field
OSI	Open System Interconnection
PAD	Padding
P-NNI	Private -NNI
PCR	Peak Cell Rate
PDH	Plesiochronous Digital Hierarchy
PDU	Protocol Data Unit
PMD	Physical Medium Dependent
PRM	Protocol Reference Model
PT	Payload Type
PTI	Payload Type Indicator
PVC	Permanent Virtual Connection
QoS	Quality of Service
RLP	Received (Cell) Loss Priority

RM	Resource Management
SAP	Service Access Point
SAR	Segmentation And Reassembly sublayer
SDH	Synchronous Digital Hierarchy
SDL	Specification and Description Language
SDU	Service Data Unit
SLP	Submitted (Cell) Loss Priority
SN	Sequence Number
SONET	Synchronous Optical Network
SSCS	Service Specific Convergence Sublayer
SSCS-PDU	SSCS Protocol Data Unit
SVC	Switched Virtual Connection
STF	Start Field
STM	Synchronous Transfer Mode
TC	Transmission Convergence ou Traffic Control
TCP/IP	Transmission Control Protocol / Internet Protocol
TD	Traffic Descriptor
UBR	Unspecified Bit Rate
UNI	User Network Interface
UPC	Usage Parameter Control
UTP	Unshielded Twisted Pair
UUI	User-to-User Indication
VBR	Variable Bit Rate
VC	Virtual Channel
VCC	Virtual Channel Connection
VCI	Virtual Channel Identifier
VP	Virtual Path
VPC	Virtual Path Connection
VPI	Virtual Path Identifier
WAN	Wide Area Network

## *Lista de Figuras*

FIGURA 2.1 - Modelo de Referência de Protocolos da B-ISDN .....	17
FIGURA 2.2 - Quadros do sistema SDH para frequência básica 155,52 Mb/s (STM-1) .....	19
FIGURA 2.3 - Interface padronizadas. ....	19
FIGURA 2.4 - Estruturada célula ATM - Interface UNI e NNI .....	20
FIGURA 2.5 - Mensagens de controle de conexão ponto-a-ponto .....	22
FIGURA 2.6 - Exemplo de um VCC formado por 4 VCLs com VCIs diferentes .....	23
FIGURA 2.7 - Formatos de endereços ATM para redes privadas .....	23
FIGURA 2.8 - Relação entre as subcamadas definidas nas normas.....	27
FIGURA 2.9 - Estrutura do Protocolo AAL tipo 2 .....	30
FIGURA 2.10 - Relação entre o AAL-SAP e o ATM-SAP .....	30
FIGURA 2.11 - O pacote da Subcamada da Parte Comum .....	31
FIGURA 2.12 - Exemplo de CPS-PDU com dois pacotes CPS e preenchimento de PAD .....	32
FIGURA 2.13 - Exemplo de multiplexação na camada AAL2.....	33
FIGURA 2.14 - Máquina de estado da transmissão .....	34
FIGURA 2.15 - Estrutura da SSSC dentro do AAL2.....	40
FIGURA 2.16 - Formato do SSSAR-PDU.....	40
FIGURA 2.17 - Formato do SSTED-PDU.....	41
FIGURA 3.1 - Diagrama do uso do ATM em diversos tipos de tráfego.....	44
FIGURA 3.2 - Exemplo de divisão da largura de banda.....	46
FIGURA 4.1 - Esquema representando as chegadas no tempo.....	52
FIGURA 4.2 - Representação simbólica dos estados no modelo de Markov .....	54
FIGURA 4.3 - Modelo ON/OFF .....	55
FIGURA 5.1 - Exemplo de rede ATM criada no simulador .....	69
FIGURA 5.2 - Esquema representando o escalonamento de eventos no simulador .....	70
FIGURA 5.3 - Duas idéias propostas de modificação do simulador .....	74
FIGURA 5.4 - Adição de rota entre as aplicações .....	75
FIGURA 5.5 - Exemplo de saída do depurador LOGAAL2.....	79
FIGURA 6.1 - Comparação da eficiência de transmissão.....	86
FIGURA 6.2 - Número de canais em um tronco DS1 variando o atraso de empacotamento .....	90
FIGURA 6.3 - Crescimento da densidade do pacote com o aumento do Timer_CU.....	91
FIGURA 6.4 - Valores de Timer_CU para alcançar 90% e 95% da máx. densidade do pacote. 92	
FIGURA 6.5 - Comparação do número de fontes obtidas pelo simulador com a Figura -6.1 ....	95
FIGURA 6.6 - Comparação da densidade do pacote obtida pelo simulador.....	96

## *Lista de Tabelas*

TABELA 2.1 - Estrutura da hierarquia em uma rede ATM.....	18
TABELA 2.2 - Classes de serviço para o ATM.....	25
TABELA 2.3 - Primitivas usadas na interação entre as camadas .....	29
TABELA 2.4 - Códigos do campo CID.....	31
TABELA 3.1 - Parâmetros de Qualidade de Serviço (QoS).....	43
TABELA 3.2 - Categorias de serviços do ATM.....	45
TABELA 3.3 - Funções por nível de gerência.....	47
TABELA 6.4 - Teste para verificar a integridade do simulador original.....	84
TABELA 6.1 - Modelo de tráfico para a simulação .....	86
TABELA 6.2 - Eficiência do protocolo AAL2 .....	87
TABELA 6.3 - Largura de Banda requerida usando AAL2 para 6 canais de voz .....	87
TABELA 6.4 - Número de canais de voz em um tronco DS1 usando AAL2.....	88
TABELA 6.5 - Modelo de tráfico para a simulação do Timer_CU .....	91
TABELA 6.6 - Tipos de testes executados para validação .....	94
TABELA 6.7 - Uso do arquivo de configuração “teste5” para validação do simulador.....	94
TABELA 6.8 - Uso do arquivo de configuração “teste6” para validação do simulador.....	96
TABELA 6.9 - Comparação do numeros de canais utilizando um tronco DS1.....	98

## *Resumo*

Este trabalho descreve o protocolo de adaptação ATM do tipo 2 (AAL2), e expõe as mudanças que foram realizadas no simulador de rede ATM com a finalidade de estudá-lo. É proposto que se conheça melhor este novo padrão, planejado para baixa taxa de transferência e multiplexação de usuários e pode contribuir nos serviços de tráfego de voz, quando ocorre a integração de diversas aplicações que é o objetivo da B-ISDN.

Após algumas explicações sobre o ATM em geral, é descrita de forma detalhada a recomendação I.362 que padroniza o novo AAL. É explicado o comportamento da máquina de estados de transmissão, e como ocorre a temporização para carregar diversos pacotes de voz numa mesma célula. Como foi idealizado para a classe de serviços do tipo VBR em tempo real, comentam-se alguns mecanismos próprios do AAL2 para controle de tráfego e sua influência. Descreve-se a subcamada SSCS proposta para pacotes longos, recentemente padronizada. São apresentados alguns artigos e estudos realizados sobre o AAL2 que descreve quantitativamente o ganho esperado pelo protocolo e levanta algumas questões comparando-o ao AAL1. Por ter sido inicialmente uma necessidade da telefonia celular, realiza-se um breve comentário de estudos dirigidos para a área.

Com o intuito de realizar a simulação do protocolo, foi utilizado o simulador de redes ATM, desenvolvido pelo NIST, que possui certas qualidades desejadas para este trabalho, porém, foram necessárias realizar modificações para implementar a camada AAL que não estava prevista na arquitetura original dele. Para se criar um ambiente de testes propício, e com a preocupação de não corromper o funcionamento padrão, foram criadas estruturas semelhantes às existentes e realizadas mudanças sutis, reconhecidas normalmente pelo simulador original.

A partir destas simulações, pretende-se deixar uma ferramenta para fazer análises deste protocolo; utilizando modelos de tráfego de voz, a fim de obter informações do seu comportamento. No entanto, este estudo limitou-se somente a demonstrar a verificação e validação do simulador a partir dos resultados gerados. Para verificar a integridade do código fonte original foram utilizados os exemplos do próprio NIST e assim garantir que nada foi alterado. Além disso, o novo módulo foi comparado a norma através de um "debug". Na validação, devido ao fato de não existir uma rede real montada e disponível para testes, foram empregados artigos para comparar os resultados e demonstrar a boa aproximação obtida. Acredita-se que desta forma obteve-se o resultado desejado de um ambiente para estudo e compreensão do AAL2 e que, futuramente, pode ser usado para todos os protocolos da camada AAL.

**Palavras-Chave:** Protocolo AAL2, Simulação de redes, Modelos de tráfego de voz, Redes ATM.

**TITLE:** “SIMULATION OF THE ATM TYPE 2 ADAPTATION PROTOCOL “

## *Abstract*

This paper describes the ATM type-2 adaptation protocol (AAL2), laying out the changes made to the ATM network simulator so that it could be studied. A better understanding of this new standard, which was planned for low delay and user multiplexing and can contribute to voice traffic services, when the integration of various applications occurs, which is the B-ISDN goal, is proposed.

After some general explanations about ATMs, the I.362 recommendation, which standardize the new AAL, is portrayed in detail. The behavior of the transmission state machine, and how the temporization occurs when loading several voice packages on the same cell, is explained. As it has been idealized for the VBR type-2 service class in real time, some mechanisms characteristic of the AAL2 for traffic control and its influence are discussed. The recent standardized SSCS sub-layer proposed for long packages is described. Some articles and studies about the AAL2, outlining quantitatively the expected gain by the protocol and raising some questions, comparing it to AAL1, are presented. Because it has initially been a necessity of the cell-phone telephony, a brief commentary of the studies directed toward this area is made.

With the intention of performing the protocol simulation, the ATM network simulator developed by NIST, which possesses certain desired qualities for this work, is utilized. However, some modifications for implementing the AAL layer not present in its original architecture had to be made. For the purpose of creating an appropriate testing environment, and also given the need for not corrupting the standard operation, structures similar to the existing were produced and slight changes, which were recognized without any difficulty by simulator, were performed.

From these simulations, we intend to leave a tool for analyzing this protocol using voice traffic models, so that information about its behavior can be obtained. Nevertheless, the present paper limited itself to only demonstrate the verification and validation of the simulator from the generated results. For verification, examples of the original source were utilized to make sure that nothing had been altered. Moreover, the new module was compared to the standard through a “debugging” process. For validation, due to the fact that there was no actual network assembled available for the testing, articles for comparing results and demonstrating its close approximation were used. It is believed that through this way the desired result of an environment for AAL2 studying and comprehension has been reached, and that it can be used, in the future, for all AAL layer protocols.

**Keywords:** ALL2 Protocol, Network Simulation, Voice Traffic Models, ATM Networks.

## ***1 Introdução***

Desde seu surgimento, o computador passou por várias fases e em cada uma delas teve um uso que lhe conferiu grande difusão pelo mundo. No início, foi utilizado como uma moderna máquina de calcular com os militares; adquiriu também a categoria de um eficiente editor de texto enquanto difundia-se pelas grandes empresas, e foi evoluindo para aplicações cada vez mais complexas na medida em que se tornava popular. Hoje, com a evolução das redes, sem discussão, o computador é uma ferramenta poderosa de comunicação.

A queda de custo do hardware retirou o computador do uso restrito de departamentos e bancos; intensificando sua utilização nas residências. Isto, aliado ao desenvolvimento da multimídia, que popularizou e facilitou o acesso às informações, foram as duas grandes forças responsáveis pelo desenvolvimento da INTERNET e das páginas da WWW.

Atualmente, também pode-se observar a explosão de outros meios de comunicação tais como a telefonia, principalmente, no caso da móvel, que se tornou acessível e cresce em toda parte do mundo, indiferente aos problemas econômicos. Com esse crescimento de demanda, não só na área de dados, mas também em todos os meios de comunicação, torna-se prioritária uma melhor utilização e desenvolvimento dos recursos de rede existentes. Há muito tempo discute-se sobre a integração de diversos tipos de aplicações em um único tipo de rede de comunicação, de forma a minimizar custos e aproveitar melhor os recursos.

O grande paradigma proposto pelos organismos de padronização foi a tecnologia ATM, que veio com o objetivo de encabeçar a Rede Digital de Serviços Integrados de Faixa Larga (B-ISDN). A partir desta proposta, os fabricantes iniciaram suas pesquisas, influenciando inclusive, as padronizações que se seguiram. Assim, desde a década de 80 o ATM vem sendo divulgado como a solução para a integração das redes, além de ser considerado como o protocolo de comunicação que iria garantir a qualidade dos serviços (QoS) de todas as aplicações até então desenvolvidas. Houve uma grande evolução neste ponto, mas ainda existem diversas dificuldades a serem vencidas.

A noção de QoS, que veio junto da tecnologia ATM, trouxe consigo os primeiros grandes desafios ainda não resolvidos na proposta do B-ISDN. Faltam mais estudos das aplicações e dos diferentes ambientes operacionais para se definirem parâmetros mais precisos, e assim, obter uma situação de perfeita garantia do QoS, bem como de uma utilização eficiente dos escassos recursos da rede que, infelizmente, ainda são bastante onerosos no caso do ATM.

É importante observar que realmente o ATM possui vantagens, e estas possibilitam um melhor controle sobre o tráfego de rede, inclusive sobre a variação no atraso das informações. Isto representa um grande diferencial, principalmente para as aplicações de tempo real tais como vídeo e som, e este aspecto tem sido crucial para sua implementação.

Para fazer uso da tecnologia ATM, as aplicações utilizam uma camada de adaptação (AAL) em que são executadas funções específicas para cada tipo de serviço de acordo com a natureza da aplicação; servindo de interface para tráfegos tão diversos

tais como os de pacotes ou os isócronos (T1 ou E1). Desta forma, a rede na camada ATM pode manter sua simplicidade e velocidade de processamento, pois não precisa conhecer a aplicação; tratando todas as células da mesma maneira. Porém, existem diferentes padrões para a camada AAL e cada um deles é responsável pelo atendimento diversificado dos diferentes serviços do ATM, além de manter o controle de segmentação e remontagem das células, multiplexação, detecção e correção de erros ou de células perdidas, entre outros.

Em 1997, foi padronizado outro protocolo de adaptação denominado AAL2 [ITU 97], que surgiu a partir das necessidades da telefonia celular. Ele foi idealizado para a classe de serviços do tipo VBR (taxa de bits variável) em tempo real, mas houve principalmente uma preocupação com a multiplexação dos usuários para melhor aproveitar a largura de banda e a característica de baixa taxa de bits na transmissão de voz compactada.

Justamente por ser o mais recente protocolo da camada de adaptação, pouca informação está disponível sobre o AAL2 e existe uma grande necessidade de estudos nesta área. Com o intuito de conhecer melhor as limitações e estudar o comportamento deste protocolo, pretende-se utilizar técnicas para simulação de rede e mostrar algumas vantagens de sua aplicação; fornecendo uma ferramenta para novos estudos sobre o assunto.

O objetivo central deste trabalho é montar um ambiente para simulação das camadas de adaptação ATM que nesta primeira versão é concentrada sobre a camada AAL2, por se tratar de um objeto de enfoque atual. Pretende-se, desta forma, proporcionar mecanismos que venham a esclarecer e testar o funcionamento da camada de adaptação sobre diversas circunstâncias de operação. Não só para a pesquisa, como também com o intuito de facilitar a didática no ensino da tecnologia ATM, e para simplificar a compreensão através da utilização do simulador.

Durante o desenvolvimento demonstra-se a avaliação feita em artigos publicados por outros autores, onde foram apresentadas as possibilidades abertas pelo novo protocolo AAL2, tais como a transmissão de voz sobre ATM para substituir a emulação AAL1 dos circuitos T1/E1. O protocolo AAL2 devido ao fato de ser desenvolvido para o tráfego VBR, tem um eficiente uso da largura de banda com grande possibilidade de ajuste, e faz um bom uso do ganho estatístico através dos múltiplos canais sobre uma única conexão ATM. Desta forma, a estrutura do AAL2 pode facilitar a vida dos administradores de rede, pois permite fazer variações no projeto inicial e procurar otimizar a rede para “casar” com as condições de tráfego. Fica claro, que o protocolo AAL2 não substituirá outros já existentes, mesmo porque, todos têm finalidades diferentes. A comparação realizada foi para se identificar os possíveis usos e para quais tipos de aplicações estão limitadas cada camada AAL, e serve para validar o simulador modificado.

No capítulo 2 desta dissertação, apresenta-se uma revisão da tecnologia ATM, tendo-se como maior enfoque a camada AAL2.

No capítulo 3, discutem-se os tipos de serviços e suas aplicações, o que é qualidade de serviço e quais os parâmetros usualmente medidos no seu estudo. Aborda-se também o papel da camada de gerenciamento e sua interferência na rede.

No capítulo 4, são abordados os modelos de tráfego e sua importância. Neste capítulo descrevem-se de forma matemática alguns dos modelos usuais; apresentando seus parâmetros e caracterizando o seu tráfego.

No capítulo 5, enfoca-se o principal assunto deste trabalho, a simulação. Neste, são revistos alguns detalhes referentes a simulação tais como técnicas empregadas e seus problemas. Logo depois, descreve-se o simulador ATM/HFC do NIST, de forma a poder introduzir as mudanças realizadas na sua programação que possibilitaram a simulação do protocolo AAL2 e das fontes de geração de tráfego.

No capítulo 6, apresentam-se artigos referentes a estudos sobre o AAL2 com o intuito de compará-los a testes de simulação realizados e, desta forma, validar as alterações realizadas no simulador. Pretende-se demonstrar também, com base nas alterações feitas e nas normas que definem o protocolo, que o funcionamento do novo módulo do simulador está correto.

No último, expõem-se as conclusões finais e propostas de trabalhos futuros.

## 2 *A Tecnologia ATM*

### 2.1 *O que é ATM*

O ATM foi definido em [CER 97] como : uma tecnologia de comunicação baseada em padrões abertos (não privativos de qualquer grupo ou empresa) que se propõe a servir de transporte comum para diversos tipos de tráfego como dados, voz (áudio), imagem estática ou vídeo. Existem diversas definições e descrições [PRY 95] [PIT 96], porém o que expressa melhor o que é o ATM são suas características.

Pode-se dizer de maneira mais técnica, que é uma tecnologia de **comutação de pacotes orientada a conexão** que utiliza pequenas células de tamanho fixo. Além disso, serve para aplicações de naturezas e requisitos de performance distintos, indo desde serviços que necessitam de tempo real até a transmissão de dados sem vincula temporal. O nome genérico para a técnica de comutação de pacotes, ao qual o ATM pertence, é *Cell Relay*.

O termo "Modo de Transferência Assíncrono" (ATM) foi usado em oposição ao termo "Modo de Transferência Síncrono" (STM), que consistia na utilização de técnicas de comutação de circuitos com alocação de banda fixa para cada circuito (técnicas como TDM - Multiplexação por divisão no tempo ou FDM - Multiplexação por divisão na frequência).

Definido o ATM por suas características, pode-se dizer que é um protocolo que provê serviços não garantidos (sem retransmissão ou correção de erros) de transferência de pacotes de tamanho fixo, denominados células. Este protocolo torna transparente o roteamento e multiplexação da rede, além de fornecer suporte ao controle de fluxo, prioridade de transporte e outras garantias de qualidade de serviço. Pelo fato das redes ATM serem orientadas à conexão, um canal virtual é estabelecido através da rede antes de qualquer transferência, o que garante vantagens na ordem da seqüência dos dados e na definição da largura de banda.

### 2.2 *Origem do ATM*

O ATM foi desenvolvido pelo ITU-T (International Telecommunication Union, Telecommunication Standardization Sector), durante o desenvolvimento da arquitetura B-ISDN (Broadband Integrated Services Digital Network), que por sua vez foi a evolução da arquitetura ISDN.

Por volta de 1984, o CCITT do ITU e alguns fabricantes de equipamentos de telecomunicações começaram a pesquisar a nova tecnologia, e por volta de 1988 começaram a surgir os primeiros resultados na forma das recomendações da série I do ITU. O conceito ISDN tinha as seguintes características : serviços integrados num único suporte, transmissão e comutação digital, e padronizações dos aspectos técnicos e interfaces de acesso. O ISDN integrou os serviços de voz e dados de taxas de transferência até 2,048 Mbit/s. O B-ISDN deverá integrar os serviços que necessitam taxas de transferência superiores (canais de alta velocidade) do suporte digital .

Em 1991 foi constituído o ATM Fórum, um organismo internacional com objetivo de acelerar o uso de produtos e serviços ATM; tentando, assim, acelerar a criação das especificações técnicas. Apesar de não ser um organismo de padronização, ele influencia nas decisões dos grupos do ITU, isto porque é formado pelos principais fabricantes de equipamentos do setor, que fazem questão de que as normas padronizadas fiquem mais parecidas com suas pesquisas. Outra instituição que surgiu auxiliando na difusão deste protocolo e teve entre outros objetivos o de integrar os serviços ATM sobre a plataforma TCP/IP da Internet, foi o *International Engineering Task Force* (IETF).

Em abril de 1991, o ITU-T através da Recomendação I.321, sugeriu o modelo de referência de protocolos (PRM) que viria a ser a base de toda padronização posterior, como pode ser observado abaixo :

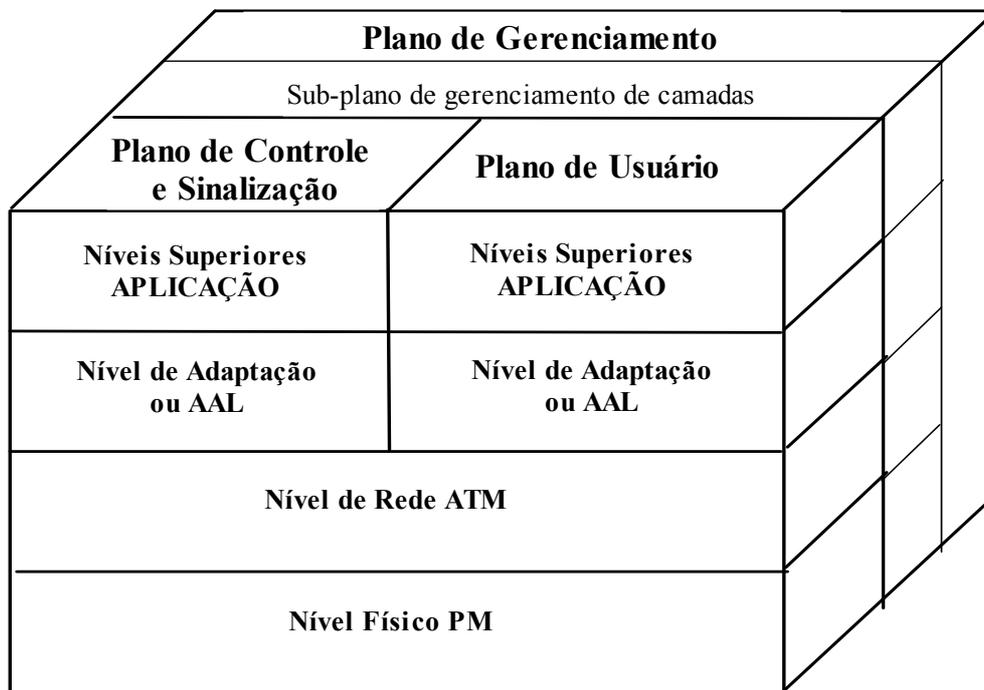


FIGURA 2.1 - Modelo de Referência de Protocolos da B-ISDN

O documento de referência mais importante produzido pelo ATM Fórum, em setembro de 1994, foi o UNI 3.1 (ATM User-Network Interface Specification), para especificar a interface entre uma rede ATM e um equipamento do usuário.

### 2.3 Descrição do ATM

Por ser adotado pelos diversos organismos internacionais, o ATM deveria se adequar ao modelo OSI, mas como isso não ocorreu. Assim, existem diversas propostas, algumas não conciliáveis, para um mapeamento entre os modelos. Porém, com certeza a representação tridimensional da arquitetura B-ISDN é a melhor forma de espelhar com fidelidade a realidade da operação ATM [STA 96][CER 97].

Desta representação tridimensional, podemos observar três planos: o de **controle**, encarregado das funções e protocolos de sinalização; o de **gerenciamento**, responsável pelo estabelecimento das conexões e gerenciamento em todos os níveis; e

por último, o do **usuário**, responsável pela transferência de informações. Cada nível deve proporcionar as funções necessárias, porém, deve ser totalmente transparente aos outros, e ao mesmo tempo manter o isolamento.

Se nos restringirmos ao plano do usuário, o modelo de camadas do ATM pode ser assim exposto :

TABELA 2.1 - Estrutura da hierarquia em uma rede ATM

Níveis Superiores (Aplicações)		
<b>Nível AAL</b>	Subnível de Convergência (CS - <i>Convergence Sublayer</i> )	- Funções dependentes dos protocolos das camadas superiores. - Recuperação relógio fonte (controle CDV). - Montagem e decomposição das PDU SAR.
	Subnível de Segmentação e Remontagem (SAR - <i>Segmentation and Reassembly Sublayer</i> )	- Detecção de perda e inserção de células - Segmentação e reagrupamento das informações
<b>Nível ATM</b>	Subnível de comutação VC ( <i>Virtual Channel</i> ), Subnível de comutação VP ( <i>Virtual Path</i> ) e Subnível de Rota de Transmissão ( <i>Path</i> )	- Controle de fluxo genérico - Geração/Extração do cabeçalho de célula - Gerenciamento e roteamento de Canais/Rotas virtuais - Multiplexação/ demultiplexação de células de conexões distinta de uma mesma interface física
<b>Nível Físico</b>	Subnível Convergência da Transmissão (TC - <i>transmission convergence</i> )	- Adaptação taxa de células com taxa de bit - Verificação/geração CRC de cabeçalho (HEC) - Delimitação das células - Empacot/Desemp células dos quadros de transporte (SDH) - Inserção/Retirada das células de preenchimento - Geração/Recuperação dos quadros de transporte
	Subnível de Dependente do Meio (PMD - <i>physical Medium Dependent</i> )	- Sincronismo de bit - Acesso físico a Rede (HDSL ou FTTC) - Outras funções dependentes do tipo de mídia, tais como temporizações e tempo de acesso

Podem-se citar alguns detalhes de cada camada:

A) CAMADA FÍSICA : desde a especificação UNI 3.1 até hoje, surgiram diversos padrões para o transporte do ATM que diferem em vários aspectos, tais como a velocidade de transmissão (desde 25 Mb/s até acima de 155 Mb/s), o meio físico utilizado (cabo ou fibra ótica), ou ainda a maneira em que os dados estão estruturados.

Os padrões mais difundidos estão baseados no padrão americano SONET (Rede Ótica Síncrona), ou no similar europeu SDH (Hierarquia Digital Síncrona), que surgiram como uma evolução do modo TDM adequado para altas velocidades. Além da frequência de operação, o SDH e o SONET definem a forma em que os octetos são montados, denominada *frame* ou quadro [figura 2.2].

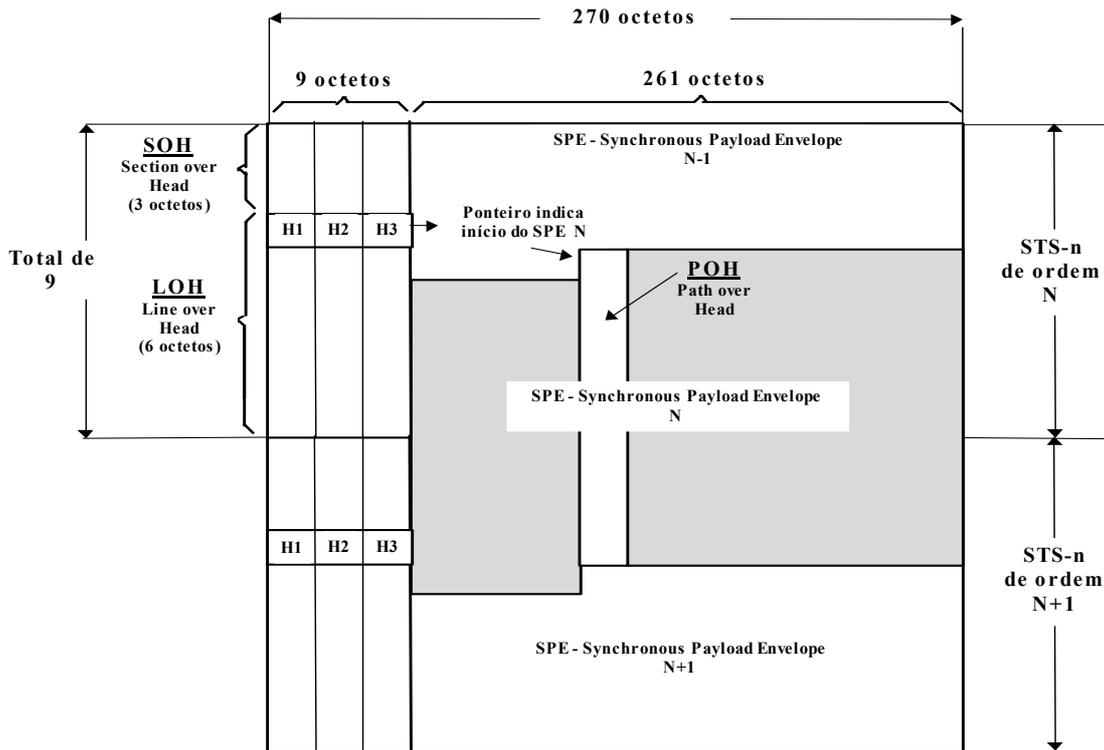


FIGURA 2.2 - Quadros do sistema SDH para frequência básica 155,52 Mb/s (STM-1)

Como pode-se observar no diagrama, existem 81 octetos com funções específicas de delineamento, identificação, verificação de erros, gerência e manutenção [ROC 98]. Dentro da área de carga útil, as células ATM são arrumadas como se ocupassem um *container* imaginário de nove linhas com 261 octetos, dos quais os primeiros de cada linha formam o chamado *path overhead* que traz mais informações necessárias a sinalização, identificação e outros.

Como mostrado anteriormente, montagem e desmontagem destes quadros são funções típicas da subcamada de convergência de transmissão, mas existem padrões que não necessitam destes envelopes para transportar o ATM e o sincronismo é realizado pelo campo HEC das células, conforme o ITU-T I.432.

B) CAMADA ATM : foram definidas as seguintes interfaces [figura 2.3]:

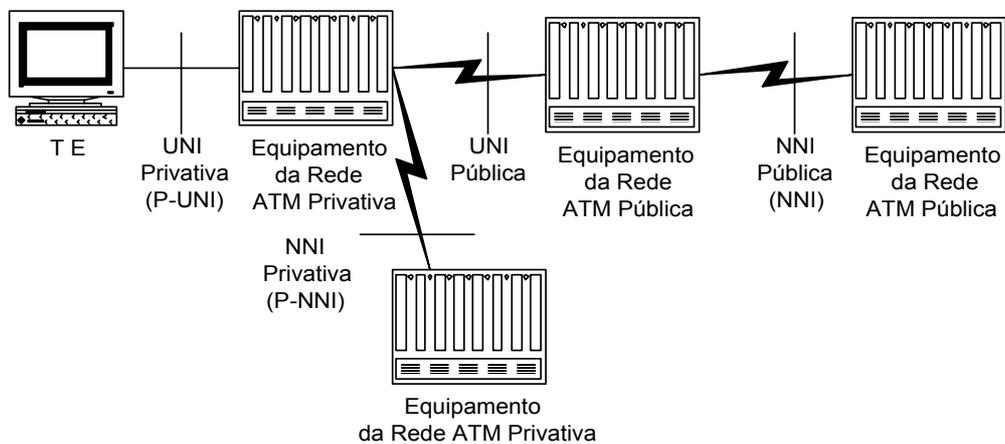


FIGURA 2.3 - Interface padronizadas.

- UNI (User-Network Interface) : padronizada pelos documentos ITU-T I.413 e I.432 e pelo ATM UNI V3.1 do ATM Fórum, especifica a conexão de equipamentos e nós de comutação.

- NNI (Network-Network Interface) : padronizada pelo documento "ATM Broadband Inter Carrier Interface (B-ICI) Specifications V 1.1", especifica a conexão entre switches em uma rede pública.

- PNNI (Private Network Network Interface) : define os procedimentos de sinalização entre switches privados, por exemplo através de um protocolo conhecido como IISP (Interim Interswitch Signaling Protocol).

O formato das células ATM já havia sido definido desde a recomendação I.361 do ITU-T, antes mesmo da UNI 3.1. A célula é formada por 53 octetos, dos quais cinco fazem parte do cabeçalho como mostrado abaixo :

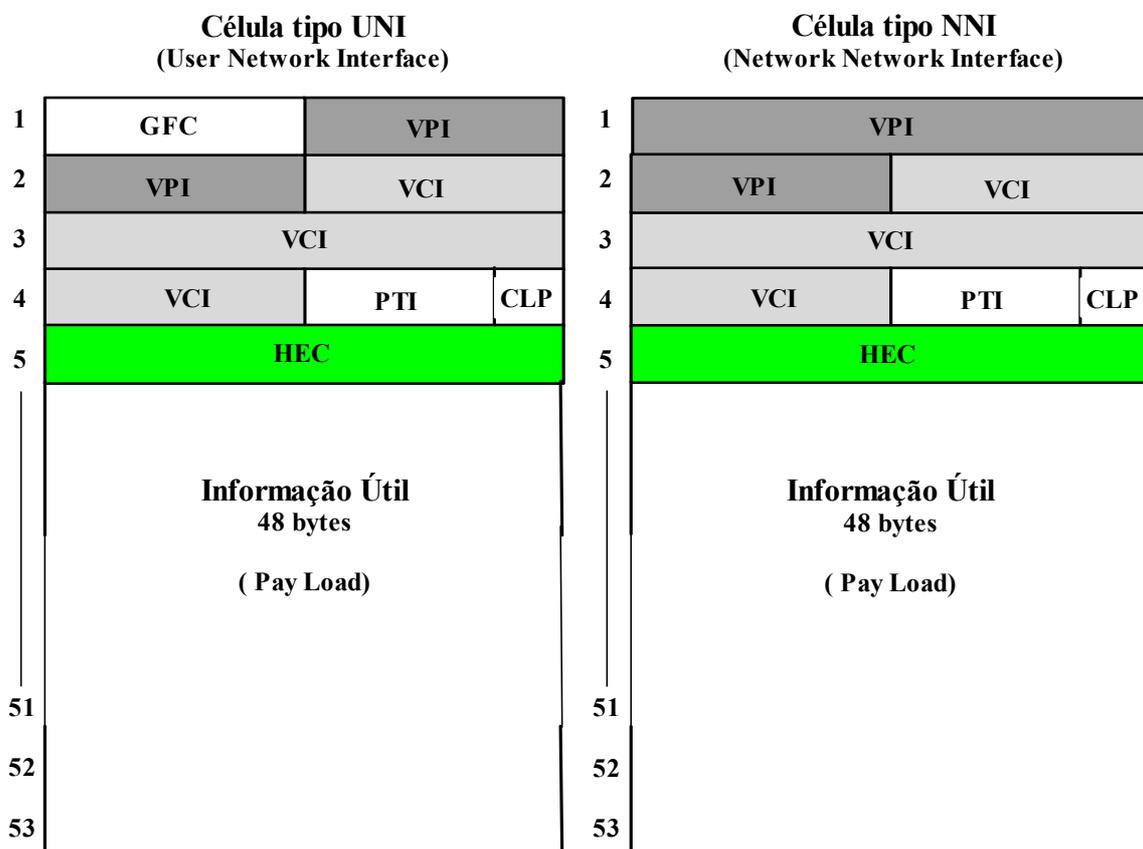


FIGURA 2.4 - Estrutura da célula ATM - Interface UNI e NNI

O significado dos campos das células é o seguinte:

- VPI (Virtual Path Identifier - 8/12 bits, UNI/NNI) e VCI (Virtual Channel Identifier - 16 bits) : identificam um canal virtual em um meio físico de transmissão, para a comutação de células ATM nos switches; a utilização de dois identificadores foi feita para facilitar a administração da rede.
- GFC (Generic Flow Control - 4 bits, UNI) : usado apenas nas células ATM da UNI, tem como função auxiliar o controle de fluxo através dessa interface.

- PTI (Payload Type - 3 bits) : indica o tipo de informação do campo Information Field da célula ATM, (por exemplo: dados de usuário ou de controle).
- CLP (Cell Loss Priority - 1 bit) : associa uma prioridade à célula, no caso de necessidade de descarte de células ATM em situações de congestionamento; células com CLP = 1 podem ser descartadas nessas situações.
- HEC (Header Error Control - 8 bits) : valor calculado para validar o cabeçalho da célula, a partir dos 32 bits restantes desse cabeçalho; é o único campo não gerado na camada ATM, e sim na subcamada TC da camada física.
- Information Field (48 bytes) : também conhecido como Information Payload, constitui os dados das camadas superiores a serem transportados via células ATM.

C) CAMADA AAL : será discutida no item 2.5 .

## 2.4 *Funcionamento do ATM*

Algumas características são marcantes no ATM, a exemplo de alguns protocolos para comunicação de dados, ele utiliza a comutação por pacotes (*cell relay*), no entanto, estabelece circuitos virtuais entre fonte e destino, semelhante a sistemas que utilizam comutação de circuitos (telefonia).

A utilização de comutação de pacotes leva vantagem sobre a comutação de circuitos por usar melhor a banda passante das linhas. Porém, a grande desvantagem do uso das tecnologias de pacotes tradicionais é o fato de terem comprimento variável e poderem chegar de maneira randômica.

Um dos segredos do ATM é a utilização de **entidades de tamanho fixo**, denominadas células, pois torna mais fácil a previsão dos atrasos inerentes da transmissão; favorecendo os projetos de rede e dos serviços suportados por ela. Desta forma, abre-se o campo para sua utilização com aplicações que necessitam de estreita sincronização.

Outra vantagem do tamanho fixo é a previsibilidade do hardware, ou seja, facilita o dimensionamento de buffers e estruturas de controle, permitindo paralelismo e escalabilidade dos recursos internos dos *switches*, o que os tornam mais rápidos. A grande desvantagem é o **overhead** introduzido pela presença obrigatória do cabeçalho, que no caso da camada ATM é de 10% (5 octetos de 53).

Pelo fato de estabelecer circuitos virtuais, o ATM é chamado de uma tecnologia **orientada a conexão**, o que traz a vantagem de garantir a ordem de transferência das células de um extremo ao outro. Para realizar a conexão do PVC (Circuito Virtual Permanente) ou SVC (Circuito Virtual Comutado) o ATM utiliza os identificadores do cabeçalho da célula. Os circuitos virtuais são multiplexados de forma dinâmica num mesmo meio físico, e conectam os diversos comutadores (*switches*) e equipamentos terminais, sempre com ligações ponto a ponto.

A subdivisão do identificador de uma conexão virtual em duas partes chamadas de canal (VC) e caminho (VP) tem por objetivo agrupar as conexões similares entre pontos comuns, obtendo as seguintes vantagens :

- simplificação no estabelecimento da conexão, pois facilita a determinação da rota;
- simplificação da administração no que se refere a controle de performance, tipo de serviço fornecido e gerenciamento;
- aumento da confiabilidade e performance, com a redução do número de entidades para controlar em cada nível da conexão.

O estabelecimento de uma conexão VP inclui cálculo de rotas, alocação de recursos e registro da informação do seu estado. Já o estabelecimento de uma conexão VC inclui a disponibilidade de um VP com capacidade de suportar as características de QoS necessárias para este. O plano de sinalização é o responsável pela realização destas conexões através de uma série de operações coordenadas, que trocam mensagens de controle e podem estabelecer, interromper ou verificar o *status* [figura 2.5].

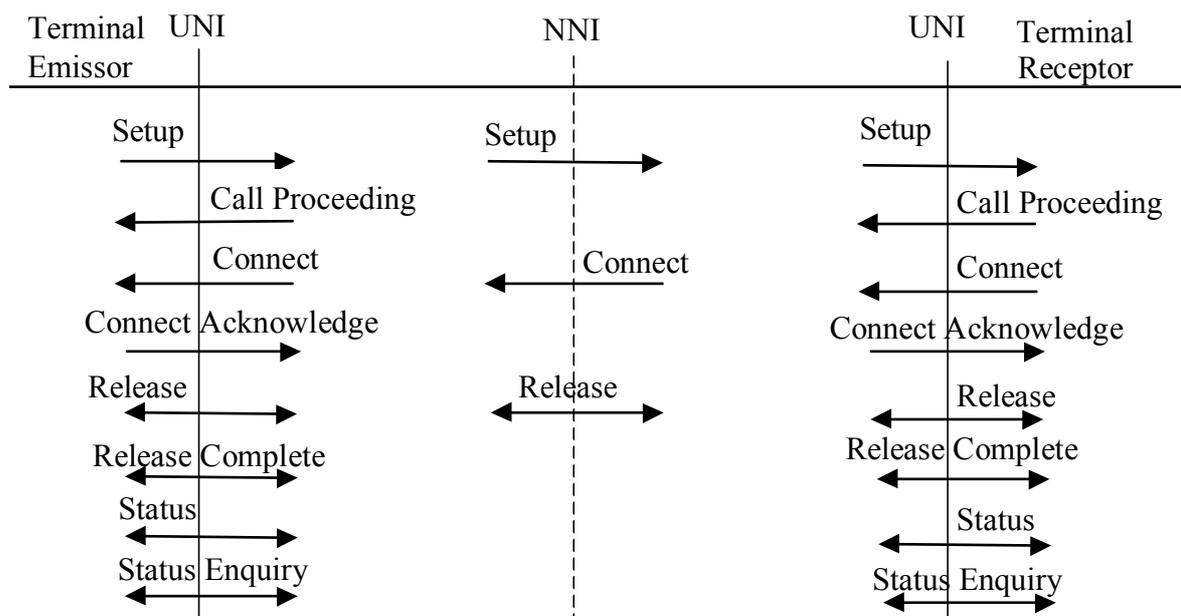


FIGURA 2.5 - Mensagens de controle de conexão ponto-a-ponto

Uma conexão ponto a ponto é denominada VCC (conexão virtual de canal) e pode conter diversos *links* denominados VCL cada qual com seu identificador único VCI, que, não precisa ser o mesmo para todos VCL do VCC [figura 2.6]. De forma semelhante ocorre nos VPC (conexão virtual de rota).

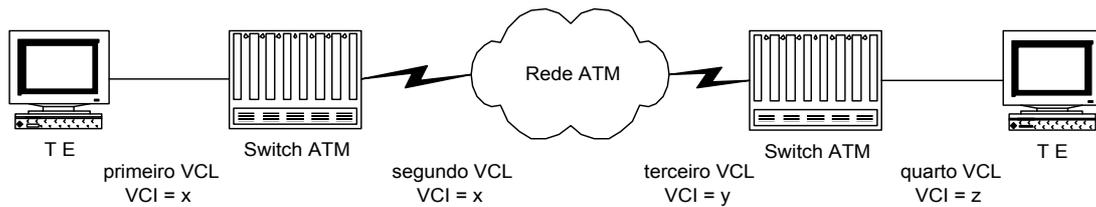


FIGURA 2.6 - Exemplo de um VCC formado por 4 VCLs com VCIs diferentes

Deve-se observar que em cada comutação acontecem alterações nos cabeçalhos das células para a troca de VPI ou VCI, logo deve ser acompanhada da geração de um novo octeto de HEC. As altíssimas velocidades de operações são garantidas pelo tamanho reduzido do cabeçalho.

Para o endereçamento dos equipamentos ATM na rede, é necessário um identificador único. Devido ao fato da rede ser orientada a conexão, este endereço é utilizado somente durante o seu estabelecimento, não sendo mais necessário depois, visto que as células usam os identificadores da conexão.

Os endereços podem ser definidos pelo usuário nas redes privadas, mas nas públicas devem ser fornecidos pela autoridade responsável pelo serviço de rede. Cada endereço ATM possui 20 octetos e tem seu formato baseado no NSAP (Network Service Access Point). Existem três formatos diferentes de endereço [figura 2.7], mas, a UNI 3.1 estabelece que qualquer TE deve conseguir conectar-se a outro independente do formato adotado por cada um.

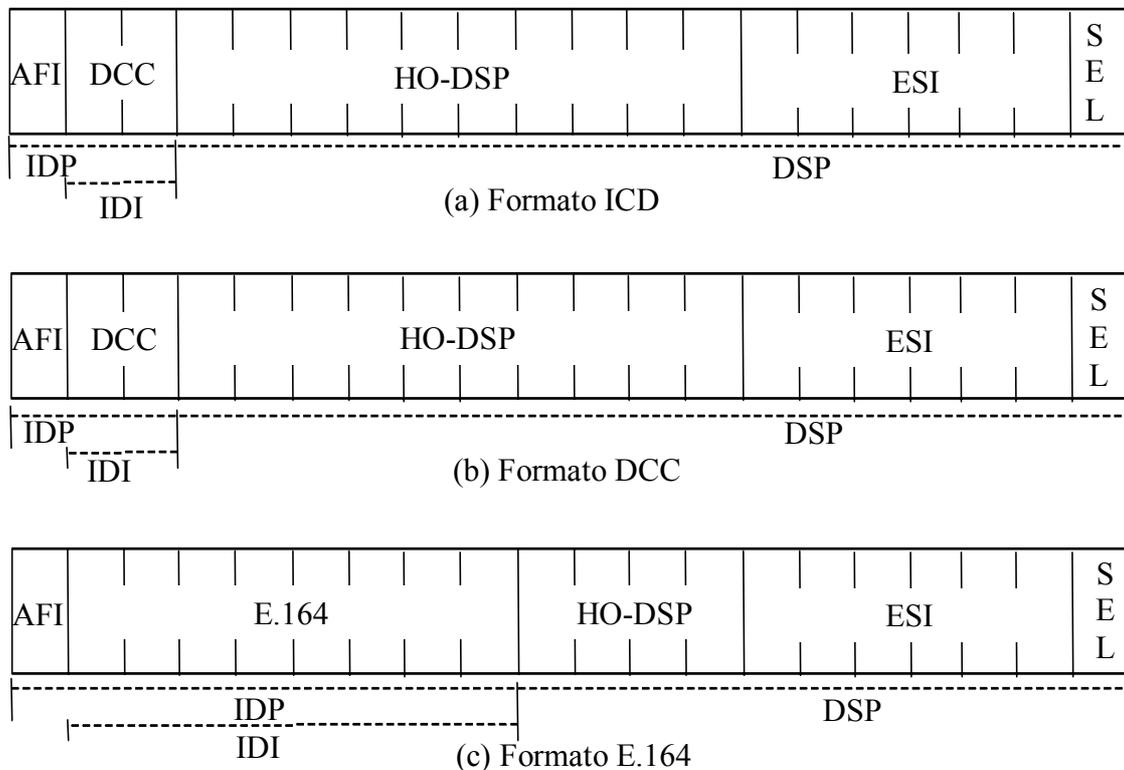


FIGURA 2.7 - Formatos de endereços ATM para redes privadas

Os campos do endereço são definidos como :

- IDP (Initial Domain Part) : formado por duas partes :
  - ⇒ AFI (Authority and Format Identifier) : identifica qual o formato do campo IDI, e por conseqüência, a autoridade administrativa que define o restante do endereço.
  - ⇒ IDI (Initial Domain Identifier) : pode ter um dos formatos abaixo :
    - ◇ ICD (International Code Designator) : fornecido pelo British Standards Institute e designa uma organização internacional;
    - ◇ DCC : fornecido na norma ISO 3166 e especifica o país ao qual o endereço está associado;
    - ◇ E.164 : fornecido pela provedora local de ISDN e especifica um número de 15 algarismos (inclusive números telefônicos).
- DSP (Domain Specific Part) : formado por três partes :
  - ⇒ HO-DSP (High Order DSP) : sua interpretação depende da autoridade especificada no IDP, mas este campo deve ser estipulado de forma a facilitar o roteamento da rede ATM;
  - ⇒ ESI (End System Identifier) : é um identificador globalmente único para um mesmo IDP + HO-DSP;
  - ⇒ SEL (Seletor) : tem significado apenas para o TE, e não serve para o roteamento ATM.

Resumindo, para dois TEs estabelecerem uma conexão, existe primeiro a identificação dos equipamentos na rede através dos switches ATM, posteriormente, o plano de sinalização verifica os recursos disponíveis e estabelece esta conexão. Com o VPI e o VCI definidos os TEs têm o seu caminho virtual pelo qual podem trocar células sem necessidade de usarem qualquer endereçamento posterior além dos identificadores.

## ***2.5 As Camadas de Adaptação ATM***

A tecnologia ATM está longe de ser considerada pronta, e devido ao fato de ser proposta para integrar todo tipo de serviço possível, surgem questões até hoje ainda não resolvidas. A estrutura hierárquica do ATM deixou para a camada AAL a difícil missão de especificar protocolos para todos os tipos de aplicações fazerem uso da mesma rede ATM.

### ***2.5.1 Descrição do AAL***

Visando dar suporte aos diversos tipos de serviços, e manter o QoS para estes, foram propostos diversos protocolos AAL, pois seria complicado juntar todas as exigências num único conjunto de regras. Esta camada é a responsável por deixar a rede

ATM transparente para as aplicações dos níveis superiores. Para isso, a camada AAL realiza certas funções essenciais, tais como :

- segmentação e remontagem dos 53 octetos das células;
- multiplexação dos diversos canais;
- sincronismo das informações;
- detecção e correção de erros;
- detecção de células perdidas ou estranhas.

Aliás, este é o principal motivo da existência desta camada, ou seja, realizar tarefas mais complexas deixando a camada ATM mais simples e funcional para tornar os *switchs* mais simples e rápidos.

Em vista das diversas categorias de serviço, foram definidas classes sobre as quais surgiram os protocolos AAL. A classificação considerou somente três aspectos :

- Relação temporal entre origem e destino : foi examinada a sensibilidade da aplicação com relação ao valor do atraso das células ou a variação deste atraso. Ex: telefonia, transmissão de vídeo, etc ...
- Taxa necessária para transmitir as informações: pode ser constante ou variável. Ex: vídeos que sofreram compressão (VBR) ou não (CBR).
- Modo de conexão : foi verificado se a aplicação é orientada a conexão ou não. Isto é diferente do fato da rede ATM ser orientada a conexão. Ex: telefonia é orientada a conexão e as transferências de dados não.

Assim, as classes e protocolos definidos seguem a tabela 2.2 . Logo observou-se a semelhança dos protocolos AAL 3 e 4, que acabaram se transformando no AAL3/4 e devido a sua complexidade acabaram por definir um novo protocolo mais simples para as mesmas classes, o AAL 5.

TABELA 2.2 - Classes de serviço para o ATM

	Classe A	Classe B	Classe C	Classe D
Taxa de Informações	Constante	Variável	Variável	Variável
Relação Temporal Ori./Dest.	Necessário	Necessário	Não necessário	Não necessário
Modo de Conexão	Orientado a conexão	Orientado a conexão	Orientado a conexão	Não orientado a conexão
Protocolo	AAL 1	AAL 2	AAL 3/4 ou AAL 5	AAL 3/4 ou AAL 5

Além destas classes o ITU-T recentemente definiu a classe X onde os requisitos de tempo ficam a encargo do usuário e a classe Y onde a largura de banda da camada ATM pode ser alterada mesmo depois da conexão.

Sobre cada AAL podem-se examinar alguns detalhes.

A) AAL 1 : por ser usado para serviços do tipo CBR, tem mecanismos para garantir esta transferência constante, tais como: manipulação de CDV, métodos de obtenção de relógio entre fonte e destino, correção de erros de um bit (CRC 3) no número de seqüência, etc.

B) AAL 2 : será vista no item [2.6 ].

C) AAL 3/4 : foi definido para serviços do tipo VBR sem vinculação temporal. Existem dois modos de funcionamento, o de mensagem e o de fluxo (para pacotes de comprimento longo). Realiza funções como : segmentação, remontagem, multiplexação, sequenciação, detecção e correção de erros.

Também existem dois tipos de serviços, o confiável, garantido pela retransmissão das células perdidas ou com erros; e o não confiável, em que as SDUs podem ser perdidas ou entregues incorretas. Este protocolo está dividido em duas subcamadas : a SAR (subnível de segmentação e remontagem) e a CS (subnível de convergência).

D) AAL 5 : é semelhante ao AAL 3/4 , e foi criado por fabricantes de equipamentos com o intuito de aliar mais simplicidade, melhor capacidade de detecção de erros e menor overhead (estudos apontam uma redução de 20% para 13% de perda [CER 97]). A principal diferença entre AAL5 e AAL3/4 reside no controle de erro, que antes era feito em cada segmento e passou a ser feito sobre a unidade de informação recebida.

Este protocolo também é dividido em subcamadas denominadas : SAR, CPCS (subnível de convergência parte comum) e SSCS (subnível convergência específica do serviço - que pode ser nula). A camada SAR não suporta multiplexação (deixada para a camada SSCS) nem utiliza cabeçalho, e a delimitação do SAR-SDU é feita pelo campo PTI da célula ATM.

## **2.6 O Protocolo AAL 2**

Neste item será resumida a recomendação I.363.2, padronizada em setembro de 1997, pelo ITU-T para o protocolo AAL 2 da camada de adaptação ATM em B-ISDN [ITU 97]. O protocolo AAL2 foi desenvolvido para atender o tráfego VBR com orientação a conexão e vinculação temporal, ou seja, destina-se a serviços do tipo voz e vídeo comprimidos em tempo real. Na verdade, teve suas características influenciadas pelas necessidades da área de telefonia celular, onde os engenheiros de telecomunicações buscavam um protocolo que pudesse atender a integração de mais esse serviço ao ATM.

Numa visão geral, este protocolo tem um eficiente uso da largura de banda com grande possibilidade de ajuste, e faz um bom uso do ganho estatístico através dos

múltiplos canais sobre uma única conexão ATM. Infelizmente acrescenta mais *overhead* no preenchimento da célula ATM, mas tendo em vista que foi idealizado para baixa taxa de transferência de dados, seu efeito final é vantajoso nas aplicações a que se destina.

### 2.6.1 O Histórico da Padronização

Desde março de 1993, quando o ITU-T (International Telecommunication Union) aprovou a recomendação I.363 [ITU 93], que especifica o formato das camadas de adaptação AAL, o protocolo AAL2 havia sido proposto para os tipos de serviço VBR em tempo real e orientados a conexão. Porém, na realidade, somente os protocolos AAL1 e AAL5 é que conseguiram ganhar adeptos e foram o grande foco dos usuários e fabricantes de equipamentos ATM. Nesta norma, a camada AAL foi dividida em duas subcamadas, a de convergência (CS) e a de segmentação e remontagem (SAR), cada qual com tarefas específicas do que se esperava naquela época.

Em setembro de 1997, o grupo de estudos 13 do ITU-T aprovou em Toronto a recomendação I.363.2 que especifica o AAL2. O novo documento sofreu influência dos setores de engenharia de comunicação móvel e trouxe consigo características como baixa taxa de bits e multiplexação de usuários. Diferente do que recomendava a I.363, dividiram a subcamada de convergência em uma parte comum (CPCS) e outra parte específica do serviço (SSCS). Esta norma limitou-se a subcamada da parte comum (CPS) do protocolo que compreende a subcamada SAR e a CPCS; abrindo a possibilidade de definirem diversas subcamadas SSCS de acordo com a necessidade da aplicação.

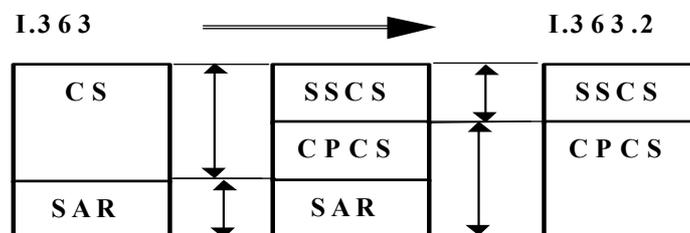


FIGURA 2.8 - Relação entre as subcamadas definidas nas normas

O AAL2 pode ser usado diretamente na subcamada CPS sem a assistência de uma subcamada SSCS. Porém, o ITU-T já definiu duas novas recomendações para o protocolo AAL2, uma para a camada SSCS genérica denominada I.366.1 [ITU 98] que foi padronizada em junho de 1998, e a outra para uso em linhas de entroncamento denominada I.366.2 [ITU 96b] e padronizada recentemente, em fevereiro de 1999.

### 2.6.2 Primitivas de Comunicação

As primitivas estabelecem um padrão de comunicação entre as camadas e subcamadas. Cada primitiva tem a definição dos campos que devem ser transferidos na interface limite entre os protocolos. O AAL2 faz uso dos serviços da camada ATM definidos na Recomendação I.361 [ITU 99a] do ITU-T, e por isso, utiliza as mesmas primitivas para transferir dados para o ATM-SAP.

As primitivas que atravessam o AAL-SAP na direção das camadas superiores são definidas nas recomendações específicas de cada protocolo SSCS. Se esta subcamada for nula, as primitivas são equivalentes as primitivas da subcamada CPS [tabela 2.3] porém, com a identificação ALL-UNITDATA.request e AAL-UNITDATA.indication para manter a consistência dos nomes definidos no SAP.

A interação com o plano de gerenciamento também provém primitivas para a subcamada CPS indicar erros (MAAL-ERROR.indication) e receber o aviso (MAAL-SEND.request) de que o ATC da conexão em vigor admite novas submissões para transmitir. Existem também, as primitivas de comunicação de gerenciamento ponto a ponto para troca de informações necessárias a este protocolo.

A interação com o plano de controle ficou fora do escopo da Recomendação I.363.2, e simplesmente especifica que este protocolo usa um circuito virtual (SVC) que já tenha sido estabelecido no nível AAL 2, e não nos canais individuais contidos dentro do nível.

TABELA 2.3 - Primitivas usadas na interação entre as camadas

Parâmetros	Primitivas		Comentários
Entre a camada ATM e a subcamada CPS			
	ATM-DATA request	ATM-DATA indication	
ATM-SDU (INFO)	obrigatório	obrigatório	48 octetos de dados do usuário
ATM-User-to-User Indication (AUU)	obrigatório	obrigatório	1 bit de informação do usuário
Loss Priority	obrigatório (SLP)	obrigatório (RLP)	CLP=1 incrementa a possibilidade da rede ATM descartar a célula
Congestion Indication (CI)	opcional	obrigatório	CI=1 indica que congestionamento foi encontrado no início ou durante a transferência.
Entre as Subcamadas CPS e SACS			
	CPS-UNITDATA request	CPS-UNITDATA indication	
CPS-interface Data (CPS-INFO)	obrigatório	obrigatório	1..45 (default) ou 1..64 octetos de dados do usuário
CPS-User-to-User indication (CPS-UUI)	obrigatório	obrigatório	5 bits de informação do usuário (somente valores "0".."27" são permitidos)
Entre a subcamada CPS e a camada de gerenciamento			
	MAAL-UNITDATA request	MAAL-UNITDATA indication	
CPS-interface Data (CPS-INFO)	obrigatório	obrigatório	1..45 (default) ou 1..64 octetos de dados de gerenciamento
CPS-User-to-User indication (CPS-UUI)	obrigatório	obrigatório	5 bits de informação de gerenciamento (somente valores "30".."31" são permitidos)
CPS-Channel Identifier (CPS-CID)	obrigatório	obrigatório	Um identificador de 8 bits para indicar qual o canal está sendo ou deverá ser usado

### 2.6.3 Descrição Geral

Como afirmou-se em seções anteriores, o AAL2 está subdividido em duas partes denominadas de CPS (Subcamada da Parte Comum) e SACS (Subcamada de Convergência de Específica do Serviço) [figura 2.9]. Diferentes protocolos SACS podem ser definidos para serviços de usuários, inclusive podendo ser uma subcamada nula e meramente mapear as primitivas entre as fronteiras.

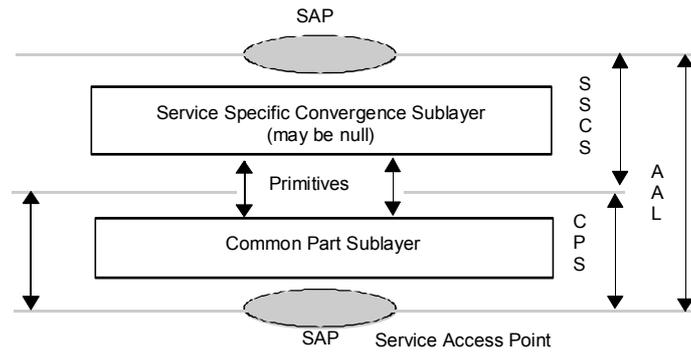


FIGURA 2.9 - Estrutura do Protocolo AAL tipo 2

O protocolo AAL 2 proporciona uma conexão ponto a ponto entre duas AAL - SAP e o usuário tem a capacidade de selecionar o QoS desejado para transportar o AAL-SDU. Múltiplas conexões AAL podem ser associadas a uma única conexão na camada ATM, sendo que a multiplexação ocorre na subcamada CPS. O usuário seleciona o QoS através da escolha do AAL-SAP usado para transferir os dados [figura 2.10].

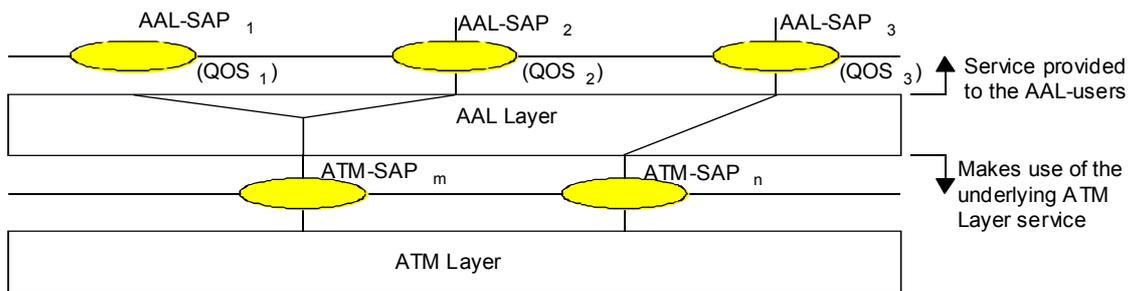


FIGURA 2.10 - Relação entre o AAL-SAP e o ATM-SAP

A subcamada CPS é capaz de transferir CPS-SDUs de um usuário CPS a outro através da rede ATM, mas o serviço proporcionado segue algumas características:

- os usuários CPS suportados são as entidades SSCS e a camada de gerenciamento;
- o serviço é não assegurado, ou seja, os CPS-SDUs são transferidos integralmente ou perdidos, mas nunca retransmitidos;
- a operação é ponto a ponto e os múltiplos canais de AAL2 são multiplexados e demultiplexados, mas a integridade da seqüência dos CPS-SDUs é mantida para cada canal;
- o canal dentro do AAL2 é um canal virtual bidirecional, ou seja, deve usar o mesmo identificador nas duas direções;
- os canais no AAL2 são estabelecidos sobre um **único** Circuito Virtual Permanente (PVC) ou Chaveado (SVC) da camada ATM.

### 2.6.4 Formatação e Codificação da Subcamada CPS

É montado um pacote CPS com estas informações recebidas das camadas superiores e o identificador da entidade que enviou. O pacote CPS consiste de três octetos no cabeçalho (CPS-PH) seguido de um corpo (CPS-PP) conforme a figura 2.11 .

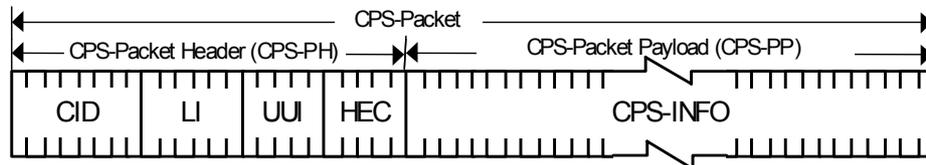


FIGURA 2.11 - O pacote da Subcamada da Parte Comum

O cabeçalho consiste de quatro partes:

#### A) Identificador de Canal (CID - 1 octeto)

O valor do CID identifica o canal interno do AAL2 e é bidirecional, o que significa, como dito anteriormente, que deve-se usar o mesmo identificador nas duas direções. O valor "0" não é usado porque os octetos zerados são usados na função de preenchimento ("padding") do CPS-PDU. Os outros valores seguem a tabela abaixo :

TABELA 2.4 - Códigos do campo CID

Valores de CID	Uso
0	não usado
1	reservado para processos ponto a ponto da camada de gerenciamento
2..7	reservado
8..255	Identificação da entidade CPS do usuário AAL2

A discriminação entre os dois tipos de usuários é feita pelo campo UUI. Através do CID, a subcamada CPS proporciona a multiplexação de até 248 entidades SSCS em uma única conexão VCC ATM.

#### B) Indicador de Comprimento (LI - 6 bits)

O campo LI é um valor binário igual ao número de octetos do CPS-PP menos um. O valor padrão do máximo comprimento do CPS-PP é 45 octetos, mas pode ser mudado para 64 octetos pelos processos de sinalização e gerência. Este valor é específico do canal interno do AAL2 e não necessita ser comum a todos os canais, porém, para um único CID, todos os CPS-PP devem ter um mesmo tamanho. Quando o comprimento é de 45 octetos, os valores de 45..63 são ignorados<sup>1</sup>.

<sup>1</sup> Os octetos no CPS-PP são numerados de "0" a "44" ou de "0" a "63".

### C) Indicação de Usuário para Usuário (UUI - 5 bits)

O campo UUI tem dois propósitos, transportar informação específica transparentemente através da camada CPS, e distinguir entre os tipos de usuários CPS. A distinção é obtida pelo valor do campo, se estiver na faixa "0".."27" é uma entidade SSCS, e se for entre "30".."31" é para camada de gerenciamento. Os valores "28".."29" são reservados para futuras padronizações.

### D) Controle de Erro do Cabeçalho (HEC - 5 bits)

O receptor usa o conteúdo do campo HEC para detectar erros no CPS-PH através de CRC. O transmissor introduz este campo através do cálculo do resto da divisão (módulo 2) do produto de  $x^5$  e os primeiros 19 primeiros bits do CPS-PH, pelo polinômio  $x^5 + x^2 + 1$ . O resto polinomial deve ser inserido no campo HEC com o coeficiente do termo  $x^4$  no mais significativo bit do campo.

Estes pacotes CPS são usados para formar os 48 octetos do CPS-PDU que será o ATM-SDU da camada ATM logo abaixo. Devido a multiplexação proporcionada pelo campo CID e como o tamanho é variável, podem ocorrer de cada célula ATM carregar diversos pacotes CPS ou o inverso, um pacote CPS ser dividido em algumas células. Desta forma, para a recepção poder controlar o número de pacotes CPS contidos no CPS-PDU existe um octeto de cabeçalho denominado campo inicial (STF - Start Field) restando 47 octetos para dados conforme a figura 2.12 . Na seção seguinte pode-se compreender melhor como ocorre esta divisão.

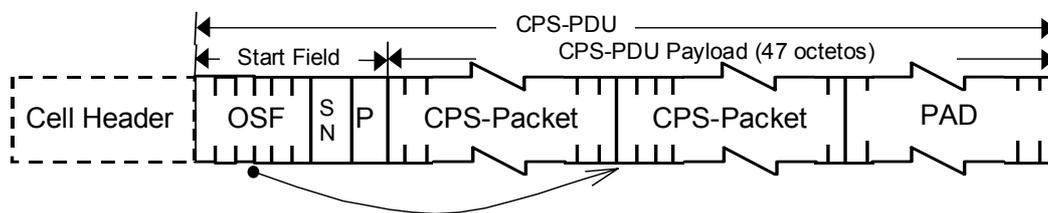


FIGURA 2.12 - Exemplo de CPS-PDU com dois pacotes CPS e preenchimento de PAD

O STF consiste dos seguintes subcampos :

#### A) Campo Offset (OSF - 6 bits)

Este campo carrega o valor binário do offset, medido em número de octetos, entre o fim do STF e o começo do próximo pacote CPS ou na ausência deste, até o início do campo PAD. O valor 47 indica que não há um limite neste CPS-PDU, e valores maiores que 47 não são admitidos.

#### B) Número de Seqüência (SN - 1 bit)

Este campo é usado para numerar (módulo 2) o fluxo de CPS-PDUs, e junto ao campo de paridade forma um mecanismo básico de controle para tentar detectar perdas de células através do STF.

#### C) Paridade (P - 1 bit)

Este bit é usado pelo receptor para detectar erros no STF. O transmissor carrega este bit com um valor tal que a paridade seja par.

### 2.6.5 Processamento da Subcamada CPS

A função de multiplexação na subcamada CPS funde vários fluxos de pacotes CPS numa única conexão ATM. Os métodos de temporização (scheduling) dos diferentes fluxos e o possível uso de prioridades não são especificados na Recomendação I.363.2.

O campo de dados (conhecido como *payload*) pode conter zero, um ou mais pacotes CPS. Quando não usado este campo é preenchido com octetos de valor zero (campo PAD). Como dito anteriormente, o pacote CPS pode também sobrepor um ou dois limites de células ATM. O ponto onde o pacote CPS é particionado, pode ocorrer em qualquer local do pacote, incluindo no cabeçalho CPS-PH, veja na figura 2.13 um exemplo.

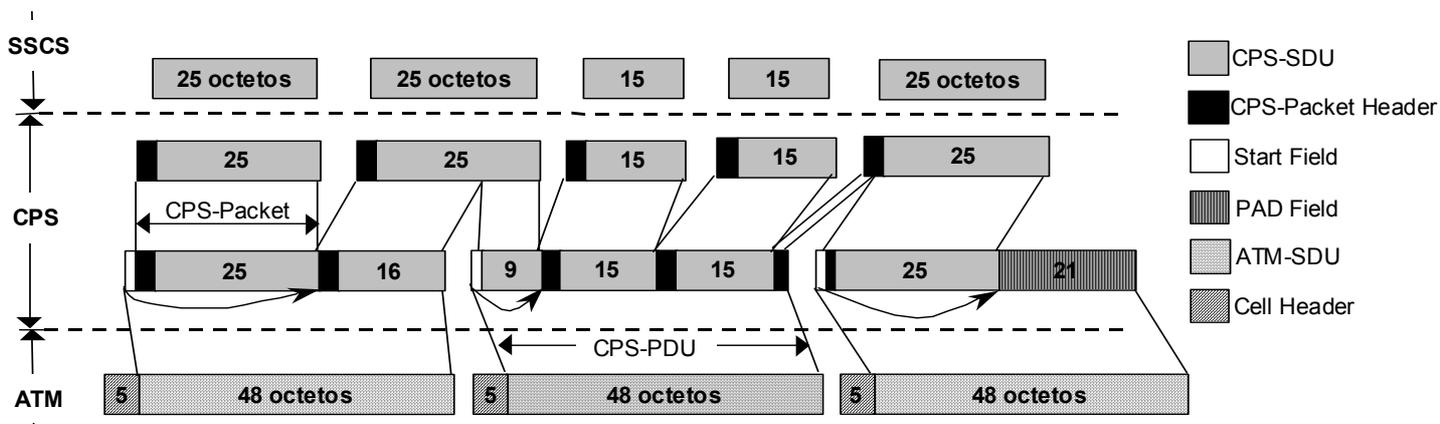


FIGURA 2.13 - Exemplo de multiplexação na camada AAL2

Neste exemplo, pode-se observar o uso de pacotes de tamanhos variáveis (25 e 15 octetos) e como uma única célula pode conter mais que um pacote. Veja o caso da segunda célula, que contém o final de um pacote, dois outros pacotes CPS inteiros e parte do cabeçalho do próximo. Além disso, fica fácil de compreender como um pacote das camadas superiores é dividido dentro do CPS-PDU; podendo no caso de pacotes maiores (64 octetos) chegar até a prolongar-se por até três células.

O anexo 1 contém figuras da norma I.363.2 que podem dar mais uma contribuição para entender como os pacotes são submetidos a subcamada CPS para empacotamento e multiplexação. Esses pacotes que vão sendo reunidos, podem vir de uma ou várias entidades SSCS ou mesmo da camada de gerenciamento.

O controle de tempo é garantido por um relógio de "uso combinado" denominado Timer\_CU que gera eventos para a máquina de estados da transmissão. Se o CPS-PDU é totalmente preenchido, então ele é enviado imediatamente (desde que exista autorização da camada de gerenciamento). Se o CPS-PDU é parcialmente preenchido, ele aguardará a chegada de mais informação até preencher totalmente ou o tempo do relógio expirar.

O controle do tempo no empacotamento é um dos fatores chave no emprego deste protocolo e será abordado nos capítulos seguintes. O sistema e os procedimentos da subcamada CPS são especificados em diagramas SDL presentes na norma [ITU 97],

e serviram de base para os diagramas SDL da programação do módulo que são mostrados no anexo 2 e discutidos em mais detalhe nas seções seguintes.

### 2.6.6 Transmissor CPS

A operação do transmissor é modelada como uma máquina de estados [figura 2.14] como mostrada abaixo:

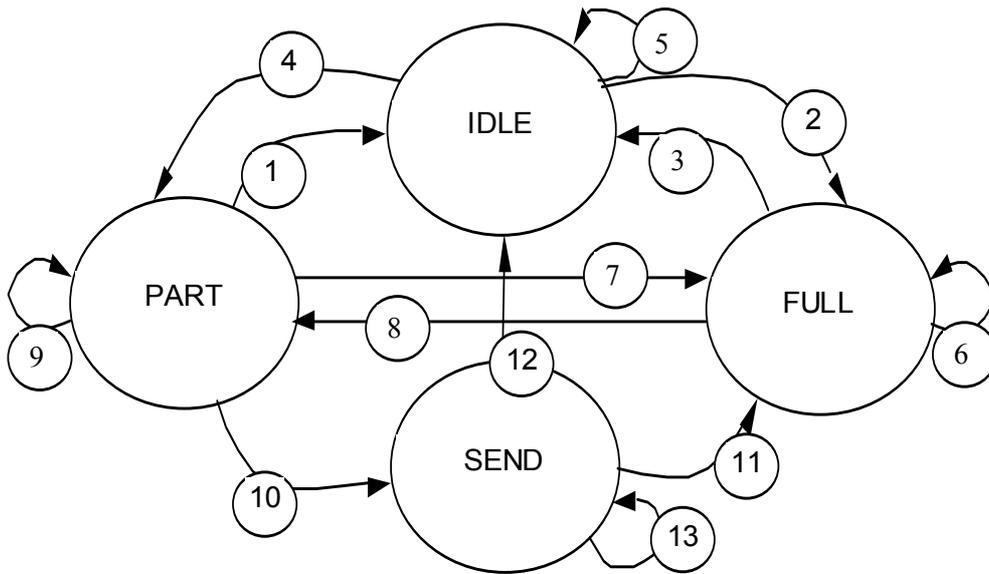


FIGURA 2.14 - Máquina de estado da transmissão

Para esta máquina de estados funcionar corretamente utilizam-se as seguintes variáveis de estado:

- *permit* : quando no estado "verdadeiro" indica que o plano de gerência deu permissão para transmissão;
- *rem* : indica o número de octetos (comprimento) do pacote CPS que ainda falta ser preenchido no CPS-PDU;
- *part* : indica o número de octetos do CPS-PP para serem colocados nos próximos CPS-PDUs;
- *split* : indica o número de octetos do CPS-PH para serem colocados no próximo CPS-PDU;
- *seq* : variável usada para ordenar o campo SN dos pacotes CPS-PDU;
- *ptrBUF* : ponteiro para indicar o próximo octeto livre no buffer do CPS-PDU;
- *tmp* : variável para cálculos aritméticos.

A operação do transmissor faz uso de um relógio denominado Timer\_CU de "uso combinado" para assegurar que os pacotes com um ou mais octetos empacotados aguardem no máximo a duração deste relógio antes de serem colocados para

transmissão. De outra forma, a atual transmissão irá se realizar logo após a recepção da primitiva MAAL-SEND.request. Aliás, se as camadas de gerência da conexão ATM requerem o envio do CPS-PDU cada vez que esta primitiva é recebida, o relógio torna-se desnecessário. Embora esta exigência não exista atualmente, está prevista na norma.

Para descrever a máquina de estados, apresenta-se cada um deles:

### A) Estado IDLE

Neste estado o Timer\_CU está parado, o buffer do CPS-PDU está vazio, e o campo OSF do STF aponta para o início do CPS-PDU payload (valor "0"). Podem ocorrer os seguintes eventos :

- receber uma primitiva MAAL-SEND.request do plano de gerência, que será guardada na variável de estado *permit*, indicando a permissão para transmitir;
- receber uma primitiva CPS-UNIDATA.request ou MAAL\_UNIDATA.request, o que inicia Timer\_CU, e o CPS-PH é construído.

Se a variável *permit* for “verdadeiro” no momento da recepção da primitiva e o número de octetos contidos no pacote CPS for igual ou maior que 44, todos os octetos do completo CPS-SDU recebidos são copiados até preencher o CPS-PDU e o pacote carregado é enviado para a camada ATM. Assim, *permit* volta a ser “falso” e a variável *rem* contém o número de octetos restantes para transmitir. O próximo estado depende do valor de *rem*, se for “0” permanece no estado IDLE (5), se for menor que 44 passa para o estado PART (1) e se for maior ou igual a 44 vai para o estado FULL (2).

Se *permit* for “falso” no momento da recepção da primitiva e o número de octetos contidos no pacote CPS for maior ou igual a 44, o processo continua no estado FULL (2); caso contrário, o processo vai para o estado PART (1).

### B) Estado PART

Neste estado o Timer\_CU está correndo, porém, o pacote ainda tem espaço para mais octetos. Podem ocorrer os seguintes eventos :

- receber uma primitiva MAAL-SEND.request do plano de gerência, que será guardada na variável de estado *permit*, indicando a permissão para transmitir;
- receber uma primitiva CPS-UNIDATA.request ou MAAL\_UNIDATA.request, o que inicia a construção do CPS-PH. O pacote é copiado para a área restante e caso ultrapasse para o próximo CPS-PDU, as variáveis *split* e *part* são calculadas. A máquina permanece no estado PART (9) se o CPS-PDU ainda contiver espaço vago e o Timer\_CU não expirou.

Se *permit* for “verdadeiro” e o pacote for preenchido completamente, então ele é passado para a camada ATM, mas se ainda restaram octetos para transmitir que não preenchem totalmente o próximo CPS-PDU (neste caso o Time\_CU é reiniciado), então permanece no estado PART (9); caso contrário, se o pacote foi preenchido exatamente com os octetos faltantes, ele volta para o estado IDLE (4)

depois da transmissão. Se não houver autorização de transmissão (*permit*=false) e o CPS-PDU foi preenchido, o processo continua no estado FULL (7).

- o relógio *Timer\_CU* expira o tempo programado; indicando que se deve transmitir o pacote sem aguardar mais dados. Se a variável *permit* for verdadeira o pacote é preenchido com o PAD, o *Time\_CU* é parado e o CPS-PDU é transmitido para a camada ATM voltando para o estado IDLE (4). Se a variável *permit* for “falso” então o processo passa para o estado SEND (10).

### C) Estado FULL

Neste estado o CPS-PDU está completo e aguardando a permissão para despachar o pacote. Se as variáveis *split* e *part* indicam sobreposição de pacotes, então o *Time\_CU* continua funcionando. Podem ocorrer os seguintes eventos :

- receber uma primitiva MAAL-SEND.request do plano de gerência, e a partir desta autorização o processo passa o CPS-PDU para camada ATM através da primitiva ATM-DATA.request. Não existindo mais dados para transmitir, o processo volta para o estado IDLE (3), ou então ele pode prosseguir no estado PART (8), se não houver octetos suficientes para preencher outro CPS-PDU. Caso contrário, permanece no estado FULL (6) se tiver octetos suficientes para preencher outro pacote;

- receber uma primitiva CPS-UNIDATA.request ou MAAL\_UNIDATA.request, este fato será registrado, porém, a Recomendação I.363.2 não especifica o que fazer neste caso, em que as informações fornecidas pela subcamada SCS excederam o ATC da conexão ATM;

- se o *Time\_CU* expirar, este fato será lembrado quando o processo tiver continuidade, mas também neste caso não é discutido na norma que atitude a camada de gerência irá tomar.

### D) Estado SEND

Neste estado o *Time\_CU* expirou, e o processo está aguardando a permissão do plano de gerenciamento. Podem ocorrer os seguintes eventos :

- receber uma primitiva MAAL-SEND.request do plano de gerência, o que inicia o preenchimento dos octetos restantes com o PAD e a partir desta autorização, envia o CPS-PDU através da primitiva ATM-DATA.request, onde o processo continua no estado IDLE (12);

- receber uma primitiva CPS-UNIDATA.request ou MAAL\_UNIDATA.request, o que inicia a construção do CPS-PH, o pacote é copiado para a área restante e caso ultrapasse para o próximo CPS-PDU, as variáveis *split* e *part* são calculadas. O processo permanece no estado SEND (13) se o CPS-PDU ainda contiver espaço vago, aguardando a permissão de envio. Se o pacote for preenchido exatamente com os octetos faltantes, ou caso os dados ultrapassem para o próximo CPS-PDU, o processo continua no estado FULL (11) e o *Time\_CU* é reiniciado para contar o tempo do próximo pacote.

### 2.6.7 Receptor CPS

Sua operação é modelada como uma máquina de estados com um único estado (IDLE) e a cada transição o processo retorna a este estado. Para o funcionamento do receptor foram definidas as seguintes variáveis de estado :

- *expct* : indica o valor esperado nos próximos CPS-PDU para completar um pacote que ultrapassou o limite do CPS-PP;
- *len* : indica o comprimento do CPS-PP;
- *split* : indica o valor esperado nos próximos CPS-PDU para completar o CPS-PH que ultrapassou o limite do CPS-PP;
- *ptrEXT* : ponteiro para marcar a próxima informação a ser processada;
- *seq* : variável usada para indicar o próximo número de seqüência esperado;
- *INFO\_Buffer* : área temporária para armazenar e montar o CPS-PP;
- *PH\_Buffer* : área temporária para guardar e analisar o CPS-PH.

O receptor também mantém os seguintes parâmetros:

- *Max\_CPS-SDU\_Length* : indica o máximo tamanho em octetos do CPS-SDU, que é transportado pelos canais AAL 2 de uma conexão ATM. Este valor é estipulado pelos processos de gerência ou sinalização, e pode assumir os valores de 45 ou 64;
- *Max\_SDU\_Deliver\_Length* : idêntico ao anterior, porém para cada canal em particular, o que corresponde ao valor entregue aos CPS dos usuários. Este parâmetro admite os valores de 1 a 45 ou 64, porém, deve respeitar a regra :  $Max\_SDU\_Deliver\_Length \leq Max\_CPS-SDU\_Length$ .

A descrição do funcionamento do receptor pode ser feita seguindo os seguintes passos:

**1 - Verificação de Paridade** : quando recebe uma primitiva ATM-DATA.indication, o STF (Start Field) do CPS-PDU é verificado, existindo erro de paridade o plano de gerência é notificado e o CPS-PDU é descartado. Se houver um pacote parcialmente recebido aguardando a segunda ou terceira parte, também é descartado. Volta-se ao estado IDLE para aguardar a próxima primitiva.

**2 - Verificação de Seqüência** : se o campo SN indicar a existência de um erro de seqüência, este é sinalizado ao plano de gerência, descartando o pacote. E havendo algum pacote parcialmente recebido também é descartado, continuando no passo 7.

**3 - Complementa CPS-PH Particionado** : se o CPS-PH anterior ultrapassou o limite do CPS-PDU, então aguarda-se o próximo pacote para processar o CPS-PH por inteiro. Assim, depois que ele chega, um ou dois octetos irão completar o PH\_Buffer, o HEC é verificado, e se houver falha o plano de gerência é notificado e o pacote parcialmente recebido é descartado, continuando no passo 7. Porém, se o HEC não

contiver erros, o campo LI é usado para o comprimento do CPS-PP através da variável *expct*.

4 - **Verifica expct** : se o valor de *expct* excede o número de octetos restantes no CPS-PDU, logo este CPS-PP ultrapassa os limites do pacote, significando que ocupa outros CPS-PDUs. Então o OSF é comparado com "47", sendo igual, os octetos são copiados para o *INFO\_Buffer* e a variável *expct* recalculada para o valor de octetos faltantes, assim sendo, este CPS-PDU está processado. A máquina volta ao estado IDLE para aguardar a última parte do CPS-PP.

Se a comparação for diferente de "47" o plano de gerência é notificado e o pacote parcialmente recebido é descartado, continuando no passo 7.

5 - **Verifica tamanho da parte final do CPS-PP** : se o comprimento esperado (*expct*) não excede os octetos livres do CPS-PDU, então esta é a parte final do pacote sendo montado. Assim, o campo OSF é comparado com a variável *expct*, se não for igual, o plano de gerência é notificado e o pacote parcialmente recebido é descartado, continuando assim no passo 7 para a extração de novas informações.

Ou não havendo pacote sendo montado, então este é um novo pacote e o número de octetos esperado é "0", continuando no passo 8.

6 - **Complementa CPS-PP Particionado** : se realmente for uma parte final de um pacote, esta informação é adicionada ao final do *INFO\_Buffer* que agora representa o completo CPS-PDU. O processo continua no passo 11, transmitindo as informações para as camadas superiores.

7 - **Verifica o campo OSF** : se o campo OSF contiver valor 47, nenhuma outra informação pode ser extraída do CPS-PDU, voltando-se então, ao estado IDLE; caso contrário, se OSF for menor que 47 a variável ptrEXT é recalculada para o próximo cabeçalho indicado pelo OSF, e a extração de futuras informações continuam. Porém, se a comparação for maior que 47, o plano de gerência é notificado deste erro.

8 - **Verifica se é PAD** : se existirem mais octetos para serem processados, mas o ponteiro ptrEXT aponta para um octeto que tem valor "0", então este octeto pertence ao campo PAD, estando portanto, encerrado o processamento deste CPS-PDU, retornando ao estado IDLE.

9 - **Carrega novo CPS-PH** : se existirem mais octetos para serem processados, mas o ponteiro ptrEXT aponta para um octeto que tem valor diferente de "0", então os próximos três octetos são assumidos como CPS-PH sendo copiados para o *PH\_Buffer* e a verificação do HEC é feita. Se esta verificação for falha, o plano de gerência é notificado, o resto do CPS-PDU é descartado, voltando-se ao estado IDLE.

Se existirem menos de três octetos, esses são copiados para o *PH\_Buffer* e a variável *split* é calculada para o número de octetos faltantes no cabeçalho, além disso, como não é possível verificar o HEC, o processamento deste CPS-PDU termina retornando ao estado IDLE.

10 - **Calcula o tamanho do CPS-PP** : se a verificação do HEC não detecta erros, o campo LI é usado para determinar o comprimento do CPS-PP. Se o comprimento do CPS-PP for maior que o número de octetos não processados, o restante

do CPS-PDU é copiado para o *INFO\_Buffer*, e a variável *expct* é calculada para o número de octetos que faltam para completar o comprimento. Este CPS-PDU está terminado e o processo volta para IDLE; aguardando o restante do pacote para montar.

**11 - Verifica tamanho máximo** : se este comprimento não excede os octetos restantes no CPS-PDU, o pacote é copiado para o *INFO\_Buffer*. Caso o comprimento deste CPS-PDU, exceda o valor de *Max\_SDU\_Deliver\_Length*, o pacote é então descartado e o plano de gerência notificado. Senão, o processo continua.

**12 - Verifica o tipo de usuário** : o campo UUI é verificado, se o valor estiver entre “0”..“27”, é então uma informação de um usuário CPS, continuando assim no passo 13; caso contrário, se o valor estiver entre “28”..“29”, o plano de gerência é notificado através de um MAAL-ERROR. Se o valor estiver entre “30”..“31”, esta informação vai para o plano de gerência através da primitiva MAAL-UNIDATA.indication, pois se trata de uma comunicação ponto-a-ponto de gerência. Continua no passo 14.

**13 - Verifica o identificador** : antes de enviar para o usuário CPS deve-se identificar a qual SAP ele está associado através do campo *PH\_Buffer.CID*. Caso exista alguma inconsistência, a camada de gerência é notificada e o pacote perdido. Senão, a informação é passada para o CPS do usuário com a primitiva CPS-UNIDATA.indication.

A informação entregue consiste do CPS-PDU passado pelo parâmetro CPS-INFO e da informação usuário para usuário no parâmetro CPS-UUI das respectivas primitivas. O CPS-CID é distribuído pelo plano de gerência.

**14 - Ajuste do ponteiro** : o ponteiro ptrEXT é ajustado pegando-se o valor do OSF+1 ou somando-se com o comprimento do campo LI, dependendo se o último pacote recebido foi particionado ou não. Isto serve para localizar o próximo CPS-PH no caso de vários pacotes no mesmo CPS-PDU. E o processamento continua como no passo 8.

### 2.6.8 *A Subcamada de Convergência Específica do Serviço (SSCS)*

Nas seções anteriores, somente foram abordadas as características da subcamada CPS do protocolo AAL2, mas nesta seção, é apresentado um pouco da subcamada SSCS da padronização I.366.1 [ITU 98]. Esta norma merece atenção, pois visa proporcionar condições de uso do AAL2 para pacotes longos, porém, esta parte do protocolo não será utilizada nos capítulos seguintes. Primeiro deve-se notar a subdivisão interna desta subcamada a fim de separar os serviços.

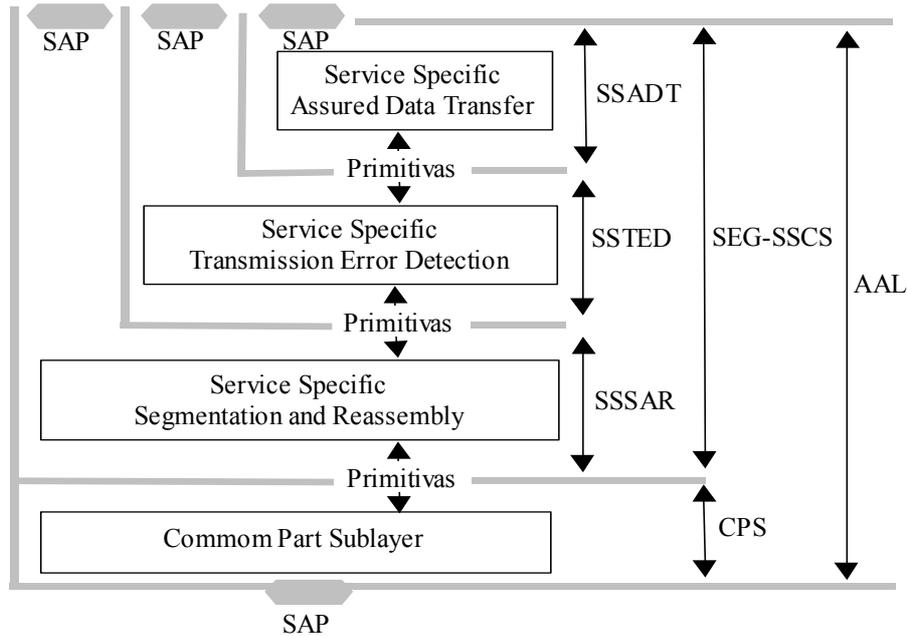


FIGURA 2.15 - Estrutura da SSSAR dentro do AAL2

### A) Serviço Específico de Segmentação e Remontagem (SSSAR)

Esta parte é responsável por receber pacotes com até no máximo de 65568 octetos, e segmenta em porções iguais ao comprimento fixado para o canal que pode variar de 1 a 64 octetos. Este tamanho máximo pode ser estipulado pela camada de gerência ou sinalização. Na verdade, o pacote SSSAR-PDU é a carga de informação do pacote CPS, como definido na figura abaixo :

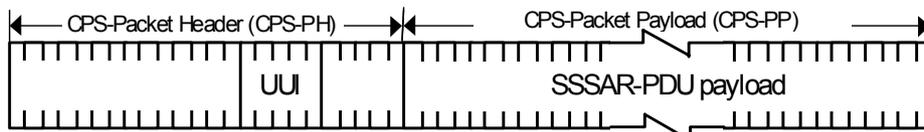


FIGURA 2.16 - Formato do SSSAR-PDU

O funcionamento é bastante simples e tem por base o campo UUI. Este campo pode conter um valor entre 0 e 26, recebido das camadas superiores e transportado transparentemente, ou conter o valor 27 necessário na recepção. O pacote longo é partido e todas as partes são enviadas a subcamada CPS, levando o campo UUI recebido, com exceção da última parte que leva o valor 27.

Na recepção SSSAR à medida que os pequenos pacotes CPS chegam, são agrupados até a chegada do último, identificado pelo campo UUI igual a 27. Não havendo erros de comprimento o pacote é entregue a camada superior. Este serviço não é assegurado, sendo descartado qualquer pacote errado ou que tenha ultrapassado determinado tempo de espera. O relógio que marca a espera é opcional, e deve ser usado para identificar a perda da última parte do pacote.

### B) Serviço Específico de Detecção de Erros de Transmissão (SSTED)

Esta parte é responsável pela detecção de erros, tais como comprimento impróprio, formatação incorreta e erros CRC. O serviço não é assegurado, ou seja, não

existe retransmissão de dados. Pode receber pacotes de até 65535 octetos, que são colocados no campo "payload" seguindo o formato definido abaixo :

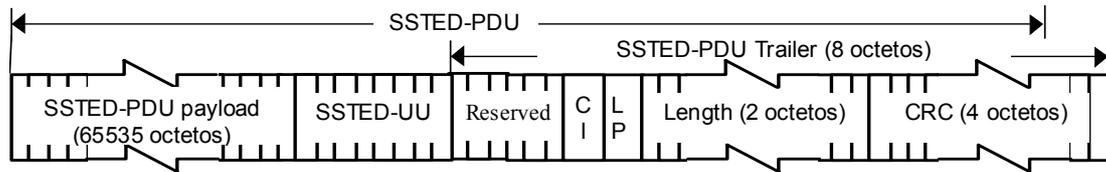


FIGURA 2.17 - Formato do SSTD-PDU

Os campos LP (Loss priority) e CI (Congestion Indication) são proporcionados para fornecer compatibilidade com o CPCS do AAL5. E o campo UU é transportado transparentemente entre usuários SSTD.

### C) Serviço Específico de Transferência de Dados Assegurada (SSADT)

Este serviço é idêntico ao definido na recomendação Q.2110 (SSCOP - Serviço Específico do Protocolo Orientado a Conexão). Pode receber pacotes de até 65531 octetos e não entrega pacotes corrompidos, ao contrário, tenta corrigi-los através de retransmissão da informação. Contém um mecanismo de controle de fluxo e outro de "keep alive" para manter ativa a conexão mesmo quando ocorrem longos períodos de ausência de transferência de dados.

### 3 QoS e o Tipo das Aplicações

Ao fazer uso da rede ATM cada usuário acessa a camada AAL através de um SAP (*Service Access Point*). Para cada SAP está associado um conjunto de parâmetros de qualidade de serviço, que definem valores como o tempo de atraso ou perda de células. Assim, o usuário deve escolher o AAL-SAP que melhor atender a seus requisitos de serviço com o intuito de manter a melhor qualidade da sua aplicação.

Nesta visão, enfocam-se características do ATM com finalidade de demonstrar a complexidade da manutenção das atividades da rede, e a necessidade de estudos direcionados ao tráfego de todos os tipos de aplicações. Discutem-se de forma genérica os diversos fatores que influenciam a qualidade de serviço de uma rede real, apesar de nas simulações normalmente se considerar de forma mais simplificada a maioria deles.

#### 3.1 Qualidade de Serviço (QoS)

Para a camada AAL realizar a difícil tarefa de reunir tão larga faixa de características para as distintas aplicações, precisamos compreender o que significa Qualidade de Serviço (QoS - *Quality of Service*) e porque é tão importante estabelecer parâmetros para entender e controlar esta qualidade.

Da forma como foi definida pelo ITU-T, a qualidade de serviço está relacionada ao desempenho da rede e o grau de satisfação do usuário. Para quantificar o QoS desejado devem ser especificados valores de parâmetros desse desempenho. É na fase de estabelecimento da conexão que as aplicações caracterizam o tráfego a ser agregado para terem acesso à rede ATM. O conjunto de parâmetros usados durante o estabelecimento da conexão é chamado *descriptor de tráfego*.

A idéia de QoS corresponde a uma exigência de qualidade e características de tráfego desejadas para manutenção de um comportamento da rede, a fim de obter a melhor garantia do serviço com a devida especificação. Para tanto, defini-se o QoS como uma combinação de parâmetros de tráfego apropriados para cada aplicação ou serviço. O grande dilema passa a ser; garantir o QoS e otimizar os recursos da rede para a crescente demanda.

Como será visto a frente, o plano de gerência utiliza diversos métodos de controle de tráfego e congestionamento, necessitando para tanto, avaliar e manter diversos parâmetros. Para transportar as células pela rede, segundo os parâmetros da QoS-NP (*network parameters*), a rede negocia uma contrapartida com o usuário, formada outro conjunto de parâmetros conhecido por QoS-UP (*user parameters*), e que caracteriza o tráfego a ser fornecido. Os principais parâmetros usados para controle de tráfego e garantia do QoS em redes ATM são apresentados na tabela 3.1 abaixo:

TABELA 3.1 - Parâmetros de Qualidade de Serviço (QoS)

Tipo de Parâmetro de QoS	Abreviatura	Significado	Observação
<b>QoS-UP</b> <b>Descritor de tráfego</b> Parâmetros relacionados com a carga a ser oferecida pelo usuário	<b>PCR</b>	<i>Peak Cell Ratio</i>	Taxa máxima na qual o usuário pretende transmitir células
	<b>MBS (ou CBS)</b>	<i>Maximum Burst Size (Cell Burst Size)</i>	Número máximo de células que podem ser enviadas, ponta a ponta, na taxa de pico PCR, sem violar a SCR
	<b>CDVT</b>	<i>Cell Delay Variation Tolerance</i>	Especifica a variação que pode ser tolerada no atraso (CDV)
	<b>SCR</b>	<i>Sustainable Cell Rate</i>	Taxa média de geração de células medido sobre um grande intervalo
	<b>BT</b>	<i>Burst Tolerance</i>	Representa a variação máxima que a taxa de geração de células pode ter em relação à taxa SCR. $BT = (MBS - 1) ((1/SCR) - (1/PCR))$
	<b>MCR</b>	<i>Minimum Cell Rate</i>	Taxa mínima de células por segundo que o usuário considera aceitável
<b>QoS-NP</b> <b>Descritor de qualidade de Serviço</b> Parâmetros de desempenho da rede	<b>CTD</b>	<i>Cell Transfer Delay</i>	Atraso médio entre fonte e destino
	<b>CDV</b>	<i>Cell Delay Variation</i>	Uniformidade de chegada (variação média do atraso das células)
	<b>CLR</b>	<i>Cell Loss Ratio</i>	Fração de células que não chegaram para o total de células enviadas
<b>Parâmetros Fixos da Rede</b> (não negociáveis)	<b>CER</b>	<i>Cell Error Rate</i>	Fração de células que chegam com um ou mais bits errados em relação ao total enviado
	<b>SECBR</b>	<i>Severely Errored Cell Block Ratio</i>	É a fração (M/N) de um bloco de N células das quais M ou mais células estão com erro
	<b>CMR</b>	<i>Cell Missinsertion Rate</i>	Número de células/s que são entregues a um destino errado, devido a erro de cabeçalho não detectado.

Quando se fala de qualidade de serviço, em uma conexão, deve-se levar em consideração o tipo de aplicação e a performance desejada para ela. O efeito coletivo destes parâmetros é que irá determinar o tráfego correto e garantir largura de banda necessárias para todas as categorias de serviços que o ATM transporta [figura 3.1].

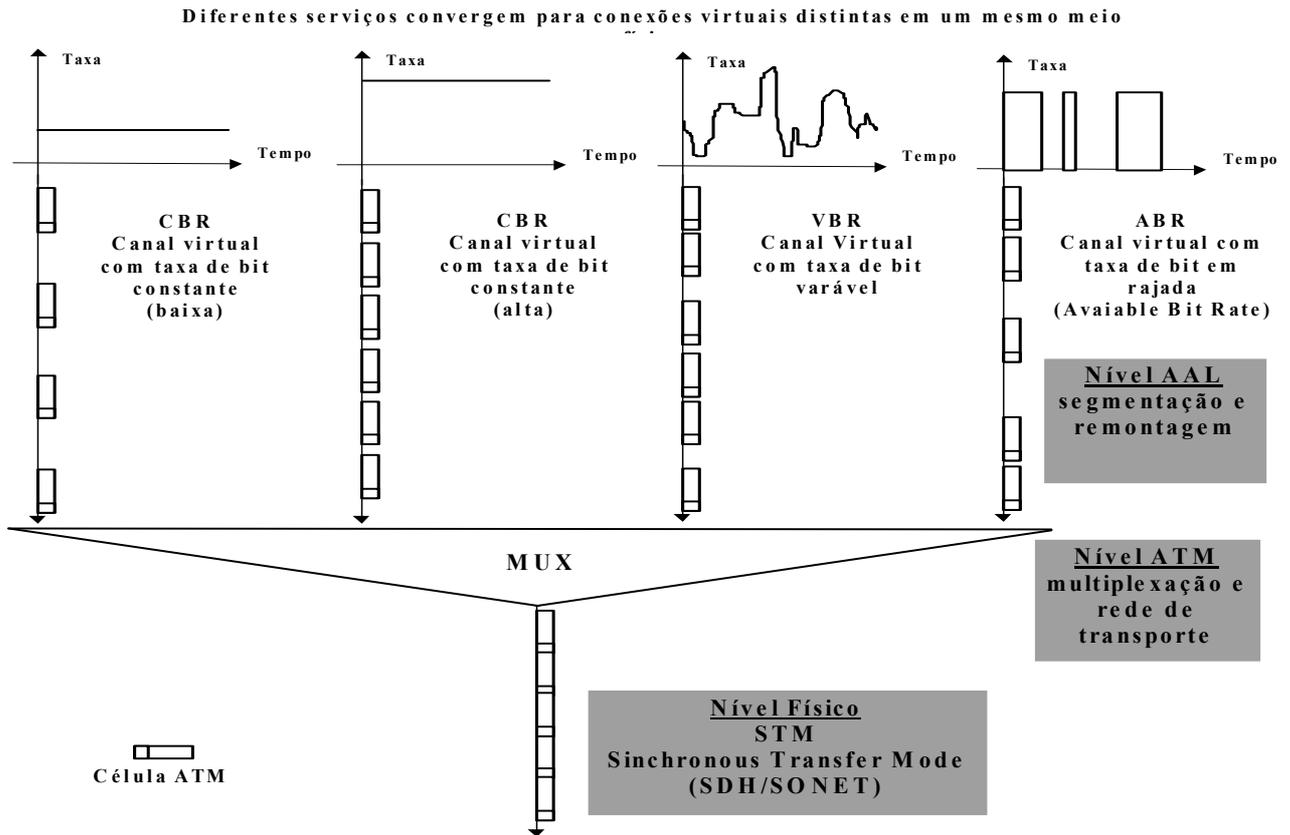


FIGURA 3.1 - Diagrama do uso do ATM em diversos tipos de tráfego

Existem outros parâmetros de QoS ligados ao controle de chamada tais como : atraso no estabelecimento da conexão, atraso de desconexão e probabilidade da aceitação da conexão. Assim, a partir do conjunto destes parâmetros se definem os chamados contratos entre os equipamentos ATM e se determinam as características de tráfego que darão o suporte necessário para integrar todos os tipos de serviços do ISDN.

Desta forma, podemos dizer que QoS é um conjunto de parâmetros de desempenho da rede que refletem a qualidade e disponibilidade do serviço oferecido. Infelizmente, ainda hoje, são poucos os levantamentos estatísticos mais compreensíveis e precisos da maioria das aplicações, deixando uma necessidade de soluções mais realísticas. Porém, foram através destas primeiras categorias de serviços que se definiram as AALs para adequar o tráfego da aplicação a camada ATM.

### 3.2 As Categorias de Serviço

O plano de gerenciamento fica durante todo o tempo monitorando os parâmetros e trocando pacotes com informações sobre a performance entre os diversos nós que compõem a rede. Isto serve para manter-se informado e poder negociar com o usuário, qual o QoS possível ou desejado na formação do contrato durante a admissão da conexão.

Por categoria de serviço ATM (ATC - ATM Transport Capability) entende-se o conjunto de procedimentos de caracterização de tráfego e requisitos de gerenciamento

de rede adequados a um tipo de serviço (ou aplicação). O ATM Forum definiu as principais categorias conforme a tabela abaixo:

TABELA 3.2 - Categorias de serviços do ATM

Serviços	CBR taxa de bits constante	VBR-rt taxa de bits variável (real-time)	VBR-nrt taxa de bits variável (non-real-time)	ABR taxa de bits disponível	UBR taxa de bits não especificada
Parâmetros					
CLR	Especificado				Não Especificado
CTD	Especificado			Não Especificado	
CDV	Especificado		Não Especificado		
TD Traffic Descriptors	PCR/ CDVT	PCR/CDVT SCR/BT		PCR/CDVT MCR/ACR	PCR/CDVT
FC (Flow Control)	Não			Sim	Não

O ATM Forum, no documento "*Traffic Management Specifications*" [ATM 96], definiu estas classes de serviços oferecidas pelas redes ATM, de acordo com os requisitos de QoS do tráfego. O ITU-T, na recomendação I.371 de 1996 [ITU 96b], definiu categorias semelhantes as do ATM Forum, com exceção da nova categoria chamada de ATM *block transfer* (ABT), que foi definida para aplicações com necessidade de transferência de informação em blocos de células. Veja alguns detalhes sobre as categorias do ATM Forum.

- CBR (Constant bit rate) : para aplicações em tempo real que exigem taxas de transferência constante, ou seja, uma largura de banda fixa. Normalmente, utiliza-se o protocolo AAL1 para garantir este tráfego, e em se tratando de voz e vídeo costuma-se adotar o método de emulação de circuitos reservando recursos durante todo o tempo da transmissão, independente de existirem dados ou não. Esta forma de tráfego é fácil de garantir e monitorar, mas consome grande parte da largura de banda disponível sendo desta forma bastante onerosa.
- VBR (Variable bit rate) : para aplicações em tempo real ou não, mas cujas taxas de transferência são variáveis, tais como aplicações de voz e vídeo comprimidos. Se a aplicação for em tempo real, o CDVT é um o principal parâmetro a ser observado, pois altos valores podem acarretar em graves problemas. Neste tipo de serviço tem-se um ganho significativo na utilização da largura de banda, porém, torna-se mais difícil seu gerenciamento.
- ABR (Available bit rate) : para aplicações que tem taxa de transferência em rajada (comutação de pacotes). O principal parâmetro é a taxa de conexão alocada (ACR) que é definida entre  $MCR \leq ACR \leq PCR$ . As aplicações deste tipo de categoria de serviço devem tolerar atrasos e variações de tráfego sem problemas ou perda de qualidade [JAI 96].
- UBR (Unspecified bit rate) : para aplicações da qual não se conhece o comportamento e se baseia praticamente no PCR. Esta categoria de serviço é semelhante a anterior, porém, deve ser usada para aplicações de comunicação de dados mais tolerantes a perda de células, tais como transferência de

arquivos e serviço de correio eletrônico. Isto porque esta categoria não provê qualquer garantia de QoS e as garantias de desempenho devem ser providas pelos protocolos das camadas de cima. Além disso, se houver congestionamento, as primeiras células escolhidas para descarte são as marcadas com CPL= 1, em seguida, as pertencentes as conexões UBR.

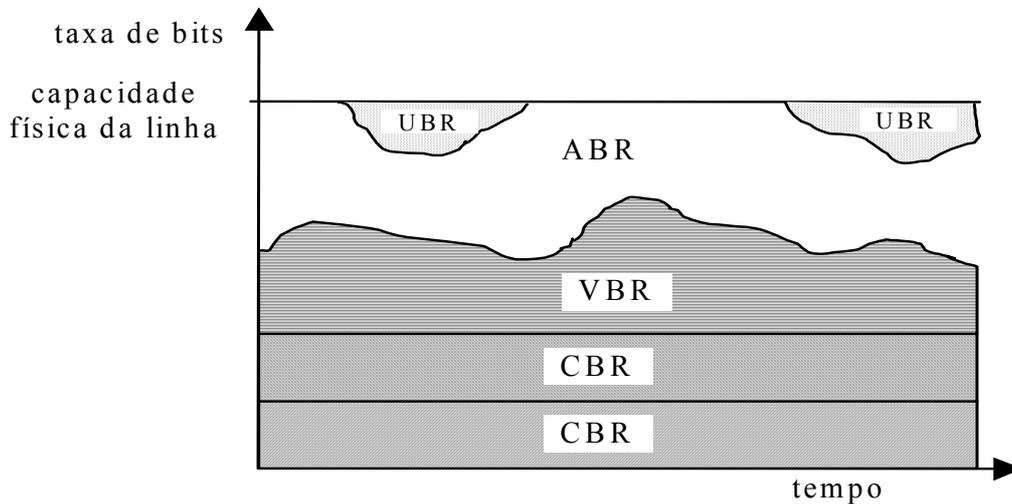


FIGURA 3.2 - Exemplo de divisão da largura de banda

No diagrama acima está exemplificado como é dividida a largura de banda entre todas estas categorias de serviço; de forma a garantir o QoS das principais aplicações e, no entanto, manter a multiplexação do numeroso e variado campo de atuação do ATM.

### 3.3 *Plano de Gerência do ATM*

É importante ter uma noção do plano de gerência, pois ele é o responsável direto pelo funcionamento correto de todos os protocolos. Neste plano é que algumas das atividades responsáveis pela manutenção da QoS ocorrem, seja prevenindo a ocorrência de problemas ou atuando sobre eles para minimizar a deterioração de serviços.

Como este não é o enfoque principal deste trabalho, pretende-se somente resumir algumas funcionalidades para ter conhecimento de sua complexidade, sem no entanto aprofundar-se. Assim, conforme resumo exposto em [CYC 97], podemos dividir o gerenciamento em:

#### **A) OAM - Operations, Administration and Maintenance**

Consiste na execução de funções responsáveis pela manutenção das camadas ATM e física dos dispositivos de uma rede, implementadas através do tráfego de células com funções especiais através de canais reservados. Para isso, as camadas são divididas em cinco níveis de transportes, com funções próprias de cada nível conforme a seguir.

TABELA 3.3 - Funções por nível de gerência

CAMADA ATM	Virtual Channel Level (nível F5)	Funções referentes a manutenção de um circuito virtual.
	Virtual Path Level (nível F4)	Funções referentes à manutenção de uma rota entre dois pontos de uma rede ATM.
CAMADA FÍSICA	Transmission Path Level (nível F3)	Funções referentes à manutenção da ligação entre entidades que tratam células ATM (por exemplo, entre switches ATM).
	Digital Section Level (nível F2)	Funções referentes à manutenção da ligação entre entidades responsáveis apenas pela comutação de seqüências de bits, sem tratamento a nível de células ATM.
	Regenerator Section Level (nível F1)	Funções referentes à manutenção da ligação física entre entidades responsáveis pelo tratamento dos sinais digitais da interface.

### **B) ILMI (Interim Local Management Interface)**

Consiste na ativação de um módulo de software conhecido como UME (UNI Management Entity). Para cada UNI dos dispositivos de uma rede ATM (switches, computadores, etc.); essas UMEs mantêm informações referentes a sua UNI, e podem trocar essas informações com UMEs adjacentes via protocolo SNMP/AAL (Simple Network Management Protocol via AAL), através de canais virtuais reservados; permitindo o gerenciamento remoto dessas interfaces.

### **C) Controle de Tráfego e Congestionamento em Redes ATM**

O controle de tráfego em redes ATM é uma tarefa importante e relativamente complexa devido às naturezas distintas dos serviços existentes, especialmente os serviços de tempo real, onde são necessários requisitos de qualidade de serviço muito restritivos e únicos, conforme discutidos anteriormente.

Este controle é exercido por um conjunto de ações com o objetivo de otimizar o uso dos recursos da rede e evitar o congestionamento. Infelizmente as redes ATM apresentam dificuldades para que este controle seja eficaz, tais como o limitado número de bits do cabeçalho para exercer controle ou as altas taxas de transmissão.

A estratégia do controle de tráfego do ITU-T e do ATM Forum é essencialmente baseada em determinar quando uma nova conexão ATM pode ser aceita. Assim, aplicam-se os descritores de tráfego para determinar o perfil de QoS, e feita a aceitação é estabelecido um contrato de tráfego. As funções de controle servem para garantir este contrato, fazendo com que a rede suporte o tráfego exigido; evitando que o usuário exceda os limites negociados.

Como descrito nas seções anteriores, é o plano de gerenciamento que cuida para manter o QoS das aplicações e uma situação indesejável é o congestionamento decorrente da sobrecarga da rede. Como ele nem sempre pode ser evitado, devem existir mecanismos que possibilitem a recuperação da condição normal de operação da rede o mais rápido possível [HES 98].

- **Controle de tráfego** - As principais técnicas empregadas para controle de tráfego são as seguintes.

- Gerenciamento dos Recursos da Rede (Network Resource Management) : consiste em agrupar canais virtuais (VCs) usados para serviços similares em caminhos virtuais comuns (VPs), simplificando assim o controle de tráfego que passa a ser um grupo de conexões de natureza comum.
- Controle de Admissão de Conexão (Connection Admission Control - CAC) : este é um método preventivo de congestionamento. Supõe-se um contrato de tráfego entre a rede e o usuário que solicita conexões à rede. Este contrato é baseado nos parâmetros de QoS; desta forma, pedidos de conexão para um dado serviço, solicitando mais recursos do que os acertados ou disponíveis, serão negados.
- Controle de Parâmetros de Usuários (User Parameter Control - UCP) : utiliza a monitoração de uma conexão a fim de verificar se o tráfego através dela está dentro dos limites contratuais, sendo que em caso de “violação” desses limites, as células “excedentes” devem ser simplesmente descartadas ou assinaladas como tais (campo CLP = 1 no seu header), para posterior descarte em outro ponto da rede. Este mecanismo pode ser aplicado também nas interfaces NNI (NPC - Network Parameter Control).
- Controle de Prioridade (Priority Control) : este método emprega simplesmente o descarte de células baseado no campo CLP do header da célula ATM.
- Conformação de Tráfego (Traffic Shaping) : pode ser descrito como o condicionamento do tráfego a um certo comportamento pré-determinado, isto é, a um tipo de tráfego padrão, só descartando as células que não estejam nele enquadradas. Esta conformação é feita com base no descritor de tráfego da fonte e permite adaptar o fluxo de informação para um determinado suporte ATC.
- Gerenciamento Rápido de Recursos (Fast Resource Management) : consiste em permitir a locação de recursos além dos contratados por um certo intervalo de tempo, desde que tais recursos estejam disponíveis durante este intervalo.
- Controle Genérico de Fluxo (Generic Flow Control - GFC) : trata-se de um mecanismo de controle por realimentação no sentido da rede para o usuário, somente atuando nas interfaces UNIs, nunca nas NNIs. O equipamento terminal que esteja usando GFC só poderá transmitir se determinados *flags* de controle permitirem, e não houver células de conexões não controladas para serem transmitidas. Este mecanismo gerou polêmica e foi duramente criticado [TAN 96], devido a sua limitada aplicabilidade, porém, os próprios organismos de padronização (ATM Forum e ITU-T) são omissos neste aspecto.

- **Reação a Congestionamento** - As atitudes a serem tomadas em situações de congestionamento são as seguintes.

- Selective Cell Discarding : é semelhante ao mecanismo “Priority Control” de controle de tráfego, porém, aqui não apenas as células com o campo CLP = 1 são descartadas, mas também células com CLP = 0, caso o descarte das anteriores não seja suficiente para restabelecer a operação normal da conexão. A seleção deve

ocorrer de acordo com o tipo de serviço, descartam-se primeiro as células de conexões UBR e assim por diante.

- **Explicit Forward Congestion Indication** : consiste em assinalar nas próprias células ATM essa situação de congestionamento (no campo Payload Type do seu header), de forma que as aplicações envolvidas possam contornar essa situação, diminuindo se possível, a taxa de geração de células.

### 3.4 *Conexões a Redes Locais*

Não se pode pensar nos dias de hoje em nenhuma rede que não tenha forma de se comunicar com a INTERNET, e o ATM não teria sucesso se não apresentasse soluções para esse problema. Além do mais, não se troca de tecnologia de repente, existe a necessidade de se manterem aplicações já desenvolvidas e com isso conectar as chamadas redes herdadas.

Esta seção apresenta resumos rápidos com o intuito de divulgar as soluções adotadas e esclarecer como o problema da garantia de qualidade de serviço pode ser maior do que se pensa, principalmente interligando redes tão diferentes como o ATM e o IP. Existem três técnicas, atualmente, que são.

#### **A) Emulação de Redes Locais (LANE)**

Nesta técnica, a rede ATM se comporta como se contivesse protocolos MAC para possibilitar o suporte transparente aos outros protocolos de rede das camadas superiores. Existem ainda problemas que estão sendo tratados pelo ATM Forum, tais como os serviços de *broadcast* e *multicast*, os endereçamentos das redes herdadas, e manter serviços não orientados a conexão sobre a rede ATM. São necessários diversas entidades para realizar a função de emulação, entre elas : o LEC (cliente), o LECS (servidor de configuração) e LES (servidor de controle).

#### **B) IP Clássico**

A vantagem de usar o IP diretamente sobre o ATM, é que pode-se ter acesso aos mecanismos de garantia de QoS da rede ATM. O IETF vem desenvolvendo a noção de Internet com Integração de Serviços (*Integrated Service Internet*), que procura introduzir uma série de melhorias no IP, de forma a permitir o suporte a serviços integrados ou multimídia. Sua função será prover serviços às aplicações enquanto a rede ATM garantirá as QoS solicitadas. Entre os novos protocolos que estão sendo desenvolvidos pelo IETF para essa finalidade, estão o Protocolo de Reserva de Recursos (RSVP - *Resource Reservation Protocol*), o IPv6 e o Protocolo de Transporte de Tempo Real [DOR 96].

#### **C) MPOA (Multi Protocol Over ATM)**

Uma das principais funções do MPOA é prover conexão fim a fim entre *hosts* na camada 3, pertencentes a redes heterogêneas diretamente ligadas à rede ATM ou às redes herdadas. Estas podem ou não estar usando LANE, conseguindo com isso aproveitar as vantagens oferecidas pela QoS do ATM. A vantagem em relação as

técnicas anteriores, é a redução da latência devido a conexão direta, e a redução de overhead. O MPOA utiliza a técnica de roteamento virtual que consiste na separação do cálculo de rota do nível 3, permitindo, assim, o uso de *Virtual LANs* (VLANs) inteligentes e uma melhor migração de estações que usam o LANE [DOR 96]. No nível de adaptação é utilizada a AAL5 e para sinalização, o padrão estabelecido pela UNI 3.1. É composto por componentes lógicos - cliente MPOA (MPC), servidor MPOA (MPS), e fluxos de informações entre estes componentes.

## 4 Modelos de Fontes de Tráfegos VBR

A importância da caracterização da fonte de tráfego é essencial para todo estudo de simulação, independente de ser uma rede ATM ou qualquer outro protocolo. Neste estudo procura-se um modelo que se aproxime das fontes de tráfego gerada pela voz em equipamentos de telefonia móvel, com o intuito de simular uma situação mais próxima da principal finalidade a que se destina o protocolo AAL 2.

Como exposto nos capítulos anteriores, antes de se projetar ou desenvolver funções de rede de comunicação é necessário conhecer as características e requisitos de tráfego que irão ser utilizados, principalmente quando se fala de uma larga faixa de serviços. Infelizmente, mesmo para uma dada classe de serviço não existe um comportamento típico. Desta forma, normalmente convive-se com diversos modelos de fontes de tráfego para simular uma mesma rede ATM e pesquisar seu comportamento.

As características das fontes de voz têm sido estudadas há várias décadas, durante o desenvolvimento das redes de telefonia, sendo por isso, relativamente bem conhecidas, porém, as propriedades normalmente dependem fortemente do esquema de codificação adotado, o que deixa ainda muito a se pesquisar.

### 4.1 A Representação Matemática dos Modelos

As funções de distribuição de probabilidade são tipicamente utilizadas para representar os modelos na sua forma matemática. Como as fontes de tráfego VBR normalmente têm um comportamento não determinístico, podem ser comparadas a variáveis aleatórias e podem-se usar métodos de estatística para modelar tais comportamentos.

Supondo-se que a probabilidade de uma variável aleatória não assuma valores superiores a "x" é  $P(X \leq x)$ , então esta forma uma função crescente de "x", pois será a soma da probabilidade  $P(x)$  com todas as probabilidades dos valores inferiores a "x"; a esta função denomina-se distribuição de probabilidade.

Freqüentemente utilizam-se determinados parâmetros para caracterizar melhor a função distribuição. É de interessante saber onde fica o centro da distribuição ou mesmo o grau de afastamento dos pontos em relação a este centro, indicando se a distribuição é dispersa ou concentrada. Com esse intuito, utilizam-se os momentos da função, e o momento de ordem "n" pode ser representado por:  $E(X^n) = \sum p(x_i) \cdot x_i^n$ .

Assim, ao primeiro momento da função denomina-se média, que representa o valor central e pode ser obtido pela seguinte fórmula:  $E(X) = \sum p(x_i) \cdot x_i$ . Outro parâmetro importante é o segundo momento, que se denomina coeficiente de variação quadrático e representa justamente a dispersão dos valores em relação a média que é expresso por:  $\text{Var}(x) = \sigma^2 = E(x^2) - [E(x)]^2$ . Algumas vezes também utiliza-se o terceiro momento, mas os demais são raramente usados.

Essas funções de distribuição são empregadas para criar o modelo de tráfego, e podem ser usadas tanto no modelo analítico como para controlar a simulação de

eventos. O mais comum contexto de modelagem é o de fila, onde o tráfego é oferecido como uma rede de filas e várias medidas de performance podem ser calculadas [FRO 94].

Pode-se considerar que um tráfego simples consiste das chegadas (*arrivals*) de entidades discretas, tal como as células. Matematicamente, apresenta-se esse processo como uma seqüência de instantes de chegadas  $T_1, T_2, \dots, T_n, \dots$ , medidos a partir de uma origem. Outra descrição equivalente desse processo é através dos intervalos de tempo entre as chegadas, representada por uma seqüência aleatória não negativa  $\{A_n\}_{n=1}^{\infty}$ , onde  $A_n = T_n - T_{n-1}$  é o comprimento do intervalo de tempo separando a  $n$ -ésima chegada de sua antecessora. Desta forma, pode-se dizer que  $T_n = \sum_{k=1}^n A_k$ .

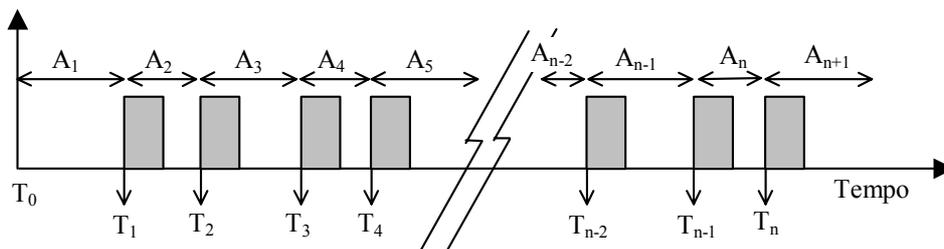


FIGURA 4.1 - Esquema representando as chegadas no tempo

Supondo que se associe a cada chegada uma probabilidade de ocorrer  $P(A_n)$ , pode-se deduzir que a contagem de chegadas tem a probabilidade  $P(T_n \leq t)$  e será uma função obtida a partir da soma de todas as probabilidades inferiores a  $n$ -ésima chegada. Assim, podemos dizer que se trata de uma função de distribuição de probabilidade.

Já tráfegos compostos consistem de grupos de chegadas (*batch*) a partir de um instante  $T_n$ . Para descrever corretamente essa característica, alguns utilizam outra seqüência aleatória não negativa  $\{B_n\}_{n=1}^{\infty}$ , onde  $B_n$  é o número de unidades no grupo. Este não será nosso caso, mas é importante entender um caso particular conhecido como tráfego em rajadas (*burstiness*) que é muito comum em redes de banda larga, principalmente, se utilizarem serviços com voz ou vídeo comprimido. O tráfego em rajadas apresenta-se na forma visual de agrupamentos, onde existe uma grande seqüência de intervalos  $\{A_n\}$  relativamente curta seguida por um intervalo longo, depois outros intervalos curtos e assim por diante. Esta modelagem é mais complexa, pois existe uma autocorrelação entre os intervalos. Existem maneiras mais fáceis de se obter medidas para representar essas rajadas, sendo que as mais comuns são : a proporção da taxa de pico para a média, que é pouco representativa e dependente do comprimento do intervalo utilizado na medida; e o coeficiente de variação, definido como a proporção do desvio padrão para a média ( $\text{Var}[A_n] / E[A_n]$ ).

Outro dado importante para se conhecer na modelagem de tráfego é o significado dos modelos de renovação. Nestes modelos, os intervalos de tempo  $A_n$  são independentes e identicamente distribuídos (IID). Este tipo de processo é matematicamente simples, porém, tem a desvantagem de não manter a autocorrelação dos intervalos que seria necessária para representar o tráfego em rajadas. Esta carência de dependência temporal, pode fornecer uma falsa medida de performance superior a real.

## 4.2 Tipos de Modelos de Tráfego

O enfoque aqui é somente apresentar alguns dos mais comuns modelos de tráfego para introduzir o conhecimento necessário aos próximos capítulos. Existe um campo muito grande de estudos sobre modelos de tráfego e seria inviável fazer uma abordagem muito abrangente, visto que este não é o enfoque deste trabalho. Na verdade, todos os tipos apresentados a seguir seguem uma ordem para se compreender uma pequena evolução.

### 4.2.1 Processo de Poisson

É um dos mais antigos modelos, datando do primórdio da telefonia. O Processo de Poisson é caracterizado como um processo de renovação onde cada intervalo entre chegadas é exponencialmente distribuído com taxa  $\lambda$  :  $P\{A_n \leq t\} = 1 - e^{-\lambda t}$ . Desta forma, a função de distribuição associada é:  $P\{T_n < t\} = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$ , que significa a probabilidade da n-ésima chegada no tempo "t" levando-se em conta todos os independentes  $A_n$ .

O processo de Poisson emprega algumas elegantes propriedades analíticas. Primeiro, a superposição de independentes processos resulta num novo processo de Poisson cuja taxa é a soma dos anteriores. Segundo, o número de chegadas em intervalos separados é estatisticamente independente (propriedade de independente incrementos), simplificando o problema de filas. Terceiro, conforme o teorema de Palm, este tipo de processo é bastante comum em aplicações que compreendem um grande número de fluxos de tráficos independentes, embora nem sempre essa agregação de tráficos resulte em um processo de Poisson.

Por ser um processo sem memória (renovação), os valores futuros não dependem da história passada. A dependência no tempo pode ser obtida fazendo o parâmetro  $\lambda$  dependente do tempo. Existe também o Processo de Poisson Composto que é definido pela especificação da distribuição do tamanho do grupo (*batch*),  $B_n$ , independente de  $A_n$ .

### 4.2.2 Processo de Bernoulli

O Processo de Bernoulli é análogo ao de Poisson, porém, para o tempo discreto, inclusive os processos compostos e dependentes no tempo. O tempo neste caso é particionado, e cada partição de tempo é denominada slot. A probabilidade de uma chegada em qualquer slot é  $p$ , independente dos outros slots.

A função distribuição que corresponde ao número de chegadas é a seguinte binomial :  $P\{N_k = n\} = \binom{k}{n} p^n (1-p)^{k-n}$ , onde "n" está entre 0 e "k". O tempo entre as chegadas é geometricamente ligado ao parâmetro  $p$  da seguinte forma :  $P\{A_n = j\} = p(1-p)^j$ , onde  $j$  é um inteiro não negativo.

### 4.2.3 Processo de Markov

Diferente dos anteriores, que eram processos de renovação, o processo de Markov introduz dependência na seqüência aleatória  $\{A_n\}$ , e conseqüentemente tem potencialidade de capturar o tráfico em rajadas.

Esse processo assemelha-se a uma máquina de estados onde o tempo de permanência em cada estado "i" é dado por uma distribuição exponencial com parâmetro  $\lambda_i$  que depende somente de i. A probabilidade de salta para o estado "j" é dado por  $p_{ij}$ , formando assim uma matrix  $P=[p_{ij}]$  de acordo com o número de estados.

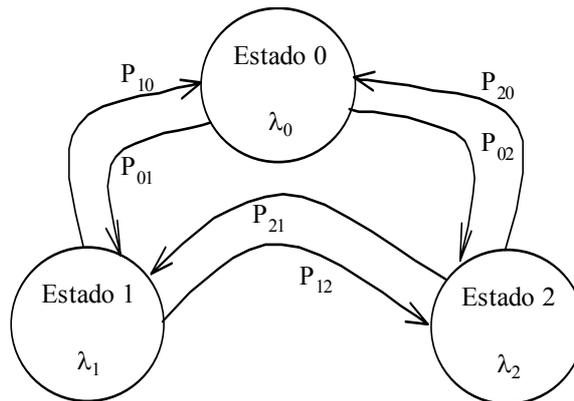


FIGURA 4.2 - Representação simbólica dos estados no modelo de Markov

A interpretação do modelo é que cada salto refere-se a uma chegada e o tempo entre as chegadas é o tempo em cada estado; resultando na discutida dependência entre os intervalos. O modelo discreto no tempo segue o mesmo princípio, porém, o tempo em cada estado é dado em números inteiros de slot, que podem representar desde únicas unidades (células) como até grupos (*batch*).

O modelo de renovação de Markov tem a simplificação de que as probabilidades são somente dependentes do estado atual, diferente do caso genérico que pode ser dependente dos dois estados ligados ao salto. Uma subclasse do modelo de renovação de Markov é o MAP (Processo de Chegada de Markov) que emprega um tipo de renovação chamada tipo fase. Nela, o estado zero é modelado como uma absorção e os demais estados são transientes, e a chegada acontecerá quando ocorrer a absorção. Porém, o estado de reinício não é sempre o mesmo e depende do estado anterior a absorção.

### 4.2.4 Modelo ON/OFF

Este é um caso particular do anterior, na verdade, uma simplificação. É uma cadeia de Markov de dois estados que pode ser modelada por três parâmetros: a probabilidade de transição do estado ativo para o inativo; a probabilidade de transição do estado inativo para o ativo; e a taxa de emissão de pacotes no estado ativo que, neste caso, é fixa.

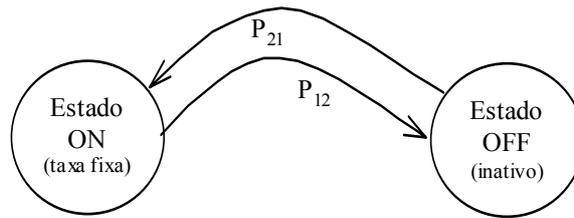


FIGURA 4.3 - Modelo ON/OFF

Uma fonte ON/OFF é uma fonte de tráfego em rajadas definida como uma fonte que alterna entre períodos ativos de emissão (períodos ON) e períodos inativos (períodos OFF). Durante um período ativo os pacotes são gerados a uma taxa de pico constante (caracterizando uma rajada), enquanto que no período inativo nenhum pacote é emitido. Pode ser modelada no tempo contínuo com os dois períodos distribuídos exponencialmente com médias  $1/\mu$  e  $1/\lambda$ , e  $T_x$  em bit/s de taxa de transmissão no estado. Quando utiliza-se o tempo discreto, ele é dividido em intervalos fixos (*slots*) e no máximo uma célula é transmitida por *slot*. Os três parâmetros passam a ser:  $(1 - q)$ ,  $(1 - p)$  e  $T_x$ .

#### 4.2.5 Processo de Markov Modulado

Esta classe de processos é extremamente importante, pois introduz a noção de estado na descrição do fluxo de tráfego. A idéia é o estado do processo de Markov controlar (modular) a lei de probabilidade que rege o mecanismo de tráfego. Este tipo de processo acaba sendo mais complicado que os processos de Markov anteriores, pois o tempo de espera em cada estado não fica restrito a uma distribuição exponencial.

#### 4.2.6 Processo de Poisson Modulado por Markov - MMPP

Este é o mais comum processo de Markov Modulado que se conhece, pois combina as qualidades de ambos os processos. Neste, o mecanismo de modulação estipula o estado corrente "k", e as chegadas ocorrem como se fosse um processo de Poisson com taxa  $\lambda_k$ . Ao mudar de estado, muda-se para outro processo de Poisson com outra taxa.

O modelo MMPP pode ser usado em diversos caminhos, mas o importante é que um único tráfego pode ser modelado mudando várias vezes de taxa de transmissão, uma para cada estado. A matrix de transição de Markov  $Q=[Q_{ij}]$  pode ter intervalos com distribuição exponencial para o chaveamento entre estados, ou pode ser estimada através de algum dado empírico. Pode-se estender o MMPP para agregar diversas fontes de tráfego independentes, onde os estados podem corresponder a diferentes números de fontes ativas com diferentes taxas de transmissão.

Existe um caso particular do MMPP que é largamente usado para modelos de tráfego de voz, onde existem somente dois estados, um ativo e associado a um processo de Poisson com determinada taxa, e outro inativo ou com taxa zero. Por esta razão, é denominado Processo de Poisson Interrompido.

### 4.2.7 Processo de Poisson Interrompido - IPP

O IPP é um processo de Poisson que fica alternativamente num estado ON por um exponencialmente distribuído período de tempo e num estado OFF por outro independente período de tempo também exponencialmente distribuído. Durante o período ativo (estado ON), o intervalo de tempo entre pacotes é exponencialmente distribuído (da maneira de Poisson), enquanto nenhum pacote é gerado no período de silêncio (estado OFF). É assumido que todos os processos são mutuamente independentes.

Se chamarmos  $1/\sigma_a$  e  $1/\sigma_s$  respectivamente a duração média dos períodos ativo e de silêncio, e chamarmos  $\lambda$  da taxa de geração de pacotes durante o período ativo, então a função de distribuição de probabilidade dos período ativo  $\Pr\{X \leq t\}$ , do de silêncio  $\Pr\{Y \leq t\}$ , e da taxa de geração de pacotes  $\Pr\{Z \leq t\}$  são respectivamente :

$$\Pr\{X \leq t\} = 1 - \exp(-\sigma_a t) \quad \Pr\{Y \leq t\} = 1 - \exp(-\sigma_s t) \quad \Pr\{Z \leq t\} = 1 - \exp(-\lambda t)$$

Além disso, se  $\pi_a$  e  $\pi_s$  denotarem que o processo ON/OFF de dois estados de Markovian está no período ativo e silencioso respectivamente. Então :

$$\pi_a = \sigma_s / (\sigma_a + \sigma_s) \quad \pi_s = \sigma_a / (\sigma_a + \sigma_s)$$

Agora, se denominarmos  $\phi(s)$  a transformada de Laplace do intervalo entre pacotes e considerarmos o processo ON/OFF logo após a chegada de um pacote, o tempo até o próximo evento é o mínimo entre  $(X, Z)$ . Este tempo é exponencialmente distribuído com taxa  $(\lambda + \sigma_a)$ . O próximo evento será uma chegada de pacote com probabilidade  $\lambda / (\lambda + \sigma_a)$  ou uma mudança de estado para silêncio com probabilidade  $\sigma_a / (\lambda + \sigma_a)$ . Ao final do período de silêncio o processo se repete. Assim, após algumas simplificações em  $\phi(s)$  nós obtemos :

$$\phi(s) = \frac{\lambda(\sigma_s + s)}{s^2 + (\lambda + \sigma_s + \sigma_a)s + \sigma_s \lambda}$$

O primeiro momento do intervalo de tempo entre pacotes  $E(X)$  e o coeficiente de variação quadrático  $c^2$  é então :

$$E(X) = (\sigma_a + \sigma_s) / \lambda \sigma_s \quad c^2 = 1 + 2\lambda \sigma_a / (\sigma_a + \sigma_s)^2$$

### 4.2.8 Processo de Bernoulli Interrompido - IBP

Este Processo é a versão discretizada do Processo de Poisson Interrompido. Nele o tempo é parcionado, e cada partição de tempo tem o comprimento igual a duração média da célula. Cada partição, denominada slot, pode estar no estado ativo ou em silêncio. No período ativo, cada slot pode conter uma célula com probabilidade  $\alpha$  ou estar vazio com probabilidade  $1 - \alpha$ , enquanto no período de silêncio nenhuma célula chega.

Supondo que se esteja no período ativo (independente do slot conter uma célula ou não), o próximo slot estará também no período ativo com probabilidade  $(1 - p)$ . De

maneira idêntica, caso esteja-se no período de silêncio, a probabilidade do próximo slot permanecer neste período é  $q$ , e de mudar de estado para ativo é de  $(1 - q)$ .

Desta forma, o período ativo e o de silêncio são ambos geometricamente distribuídos com respectivamente os seguintes tempos médios  $1/(1 - p)$  e  $1/(1 - q)$ , que podem ser representados por:

$$\Pr(X = x) = (1 - p) \cdot p^{x-1} \quad \Pr(Y = y) = (1 - q) \cdot q^{y-1} \quad x, y \geq 1$$

Similar ao caso de tempo contínuo, a solução das probabilidades do estado estável iniciar em ativo  $\pi_a$ , e de silêncio  $\pi_s$ , são dadas por :

$$\pi_a = (1 - q)/(2 - p - q) \quad \pi_s = (1 - p)/(2 - p - q)$$

A média  $E(X)$ , o coeficiente de variação quadrática  $c^2$  e o terceiro momento  $E(X^3)$  do intervalo de tempo entre chegadas de células de um IBP, podem ser obtidos da transformada  $Z$ ,  $\phi(z)$ , e são definidos em termos dos parâmetros  $(p, q, \alpha)$ . Se for utilizado o símbolo  $\rho = 1/E(X)$  que representa a probabilidade do slot conter uma célula, definido como utilização da fonte, e ainda assumirmos que no período ativo todo slot contém uma célula ( $\alpha = 1$ ), podemos fazer simplificações e obter as seguintes equações:

$$E(X) = \frac{2 - p - q}{(1 - q)} \quad c^2 = \frac{(1 - p)(p + q)}{(2 - p - q)^2} \quad E(X^3) = \frac{6(1 - \rho)q}{\rho(1 - q)^2} + \frac{3c^2 + 3 - 2\rho}{\pi^2}$$

#### 4.2.9 Outros Modelos e Estudos

Os modelos até agora apresentados servem na sua maioria para representar fontes de tráfego de voz, e já dão uma idéia de como pode ser abrangente e complexo este assunto. No caso do protocolo que se deseja simular, ele foi idealizado a princípio para ser utilizado com tráfego de voz, então é suficiente que se utilize um desses modelos anteriores. Mas caso considerem-se outros estudos [SIL 2000][FRO 94], existem ainda alguns modelos mais indicados para outras aplicações, como os listados abaixo:

- **Modelo de Fluxo Contínuo:** são modelos que geram células continuamente (como um fluxo), ao invés de gerarem em instantes individuais do tempo, mas funcionam de forma similar ao modelo ON/OFF. Normalmente, são apropriados a casos onde unidades individuais são relativamente numerosas na escala de tempo escolhida.
- **Modelo de Tráfego Auto-similar ou Fractal :** nestes modelos, o tráfego manifesta-se como uma invariância em relação à escala de tempo, e é empregado para fontes de tráfegos que apresentam a característica de explosividade, tais como as redes locais.
- **Processos Autoregressivos :** é utilizado para modelar seqüência de quadros, e por isso, com freqüência aparece como sendo um grupo de modelos de interesse para tráfego de fontes de vídeo. Estes processos servem para tentar representar a autocorrelação entre os vários pacotes de dados.

- Modelo ON/OFF de Pareto : este tipo de modelo exibe a característica de alta variabilidade (efeito Noé), que se acredita ser o fator responsável por gerar a auto-similaridade. Alguns estudos [SIL 2000] apontam para o uso deste modelo no caso de multiplexação estatística de fontes heterogêneas, por representar bem a explosividade dos tráfegos de redes ethernet e se aproximar mais da realidade.

### 4.3 *Simulação da Fonte de Tráfego*

Analisando os modelos expostos nas seções anteriores, escolheu-se utilizar fontes de tráfego do tipo ON/OFF pela sua simplicidade. Principalmente, tendo em vista o que o principal objetivo não é analisar qualquer parâmetro de performance e sim verificar se o protocolo de adaptação AAL2 pode ser simulado corretamente. Isto deve-se ao fato de não existirem exigências em relação a obter dados com precisão para casos extremos ou peculiares, e sim a necessidade de simulações em regime permanente. O modelo ON/OFF é largamente usado e se aproxima o suficiente do real para o que se deseja.

Pode-se utilizar a categoria CBR para transmissão de voz, mas como exposto anteriormente, devido a grande parte da conversação ser de intervalos de silêncio, tem-se um grande desperdício de largura de banda. Desta forma, a categoria de serviços VBR é mais indicada e também é mais complexa de modelar.

Os serviços de voz têm exigências de QoS bem mais severas do que serviços de dados, porém, com uma largura de banda mais restrita que serviços de vídeo. Em relação ao desempenho da rede, as aplicações de voz exigem que não se tenham grandes atrasos na transmissão para evitar sobreposição de diálogos, quando surge a necessidade de usar a palavra câmbio para saber que o outro terminou de falar. Exige-se também que a variação deste atraso seja restrita para evitar problemas de sincronismo e não gerar descontinuidade no som. Porém, podem tolerar uma certa perda de células sem grande degradação da qualidade.

O período ativo de uma fonte de voz pode ser comparado a algumas rajadas de conversa, considerando que um período silencioso corresponde a várias durações nas quais o locutor pausa para respirar, organiza pensamentos, ou escuta ao outro locutor. Pode-se chegar a 65 % de silêncio na transmissão em cada direção. Além disso, pesquisas têm demonstrado que o período ativo se aproxima razoavelmente bem de uma distribuição exponencial, enquanto a duração do período de silêncio é um pouco menos representado por essa distribuição [JAI 99].

Mesmo assim, os mais frequentes modelos de fontes de voz da literatura assumem que a duração dos dois períodos, silêncio e atividade, são distribuições exponenciais. A precisão deste modelo tem sido validada em uma única fila em que o número de fontes de voz é multiplexado. Tem-se mostrado que este modelo de dois estados com distribuição exponencial é razoavelmente bom se mais que 25 fontes de voz são multiplexadas, e fica bem menos satisfatório se este número for inferior a 10 [ONV 95].

Assim, uma proximidade da fonte de tráfego mais comum utilizada com o protocolo AAL2 é suficiente para realizar as simulações, e neste caso, procura-se um

representação de tráfego de voz ou voz comprimida. Infelizmente, pouco se conhece sobre o comportamento dos sinais comprimidos. Por outro lado, o modelo ON/OFF se aplica muito bem as fontes de voz e deste modo foi empregado para os testes realizados conforme será exposto nos próximos capítulos.

### *4.3.1 Criação de Fontes de Tráfego*

Para realizar a simulação foi escolhido um simulador conforme será visto mais a frente, porém, depois de realizada a escolha, foi verificado que existiam alguns problemas a serem enfrentados. O simulador tem entre suas fontes de tráfego (internamente conhecidas por aplicações ATM), vários modelos já predefinidos, inclusive um denominado VBR.

A documentação original do NIST, indica este modelo VBR como sendo o de Poisson e é na verdade um modelo ON/OFF constante, o que pode gerar complicações caso alguém utilize. Apesar disso, pode-se fazer uso de uma das facilidades do simulador que permite alterar o código fonte, e desta forma, podem-se acrescentar novos componentes. A alteração constitui basicamente na criação de um componente com o modelo um pouco modificado para as funções especificadas, tais como as deste capítulo, e na alteração de algumas outras partes do simulador, a fim de que ele reconheça a nova parte criada.

Foi criado um modelo ON/OFF exponencial a partir do modelo de fonte VBR, simplesmente colocando o comprimento do período ativo como uma variável aleatória que segue uma função exponencial de média igual ao parâmetro de usuário “Mean Burst Length”. Para que os dois períodos se comportassem da mesma forma, modificou-se o período entre rajadas colocando também uma função idêntica a anterior de média igual ao parâmetro “Mean Interval Between Bursts”. Os testes mostraram um comportamento aceitável, dentro do que era esperado para iniciar as simulações.

## **5 Simulação do AAL2**

Neste capítulo, descreve-se brevemente o assunto de simulação de redes e seus desafios. Depois comenta-se sobre a escolha do simulador usado, passando para uma descrição geral do funcionamento interno com o intuito de compreender melhor as mudanças realizadas. A utilização do simulador é bem simples, intuitiva e bem documentada, por isso é pouco abordada. Para finalizar, explica-se o novo módulo, sua utilização e quais foram as alterações do programa fonte original e o porquê delas.

### **5.1 Simulação de Redes**

#### **5.1.1 Comentários**

A simulação é a imitação da operação do mundo real ou sistemas no tempo [SOR 90]. O uso do computador para simular baseia-se numa abordagem experimental com o intuito de tornar-se útil no suporte à tomada de decisões, principalmente quando os testes em sistemas reais não podem ser realizados. Pretende-se assim, obter uma forma fácil para prever o desempenho de um sistema ou comparar alternativas na busca da melhor solução.

O enfoque da simulação deve ter como objetivo construir ambientes virtuais e a partir deles disponibilizar medições, a fim de obter conclusões sobre os problemas que se deseja discutir. Sempre lembrando que qualquer solução requer que o problema seja bem definido [JAI 91], e desta forma, pode-se descobrir o melhor mecanismo para obter esta solução, seja ele por simulação ou não.

#### **5.1.2 Definições e Conceitos**

O processo de simulação é um retrato dinâmico das mudanças do estado de um sistema ao longo do tempo [BAN 95]. Para fazer uso deste processo, é fundamental obter-se um modelo do sistema em estudo. O modelo é uma descrição deste sistema de forma abstrata. Este é um ponto chave, pois se este sistema pode ser descrito por um conjunto de variáveis, cuja combinação represente um único estado, então a manipulação dos valores destas variáveis simula o comportamento do sistema. Uma linguagem de simulação fornece estruturas para criar este modelo e traduzi-lo em uma representação lógica que pode ser exercitada experimentalmente em um computador.

Existem diferenças nestes modelos para simulação, que são representações através de estruturas lógicas para mimetizar o comportamento de um sistema [HEU 90]. Os modelos analíticos, são definidos por uma série de equações matemáticas, através das quais o comportamento do sistema é obtido pela atribuição de valores aos parâmetros e à solução destas equações. A solução analítica é empregada onde existe a possibilidade dessa representação matemática, e quase sempre, para que isto ocorra devem-se fazer simplificações da realidade, o que pode limitar seu uso a problemas específicos ou restritos a uma área. A resolução pode ser complexa e em muitos casos empregam-se as chamadas técnicas de aproximações.

Como mencionado, a simulação é um experimento estatístico que observa o comportamento do modelo no decorrer do tempo. Logo, existe a necessidade de dados para alimentar o experimento, podem ser obtidos de medidas do sistema real ou mais usualmente, de valores aleatórios produzidos por um algoritmo de acordo com a distribuição de probabilidade desejada (discutido no capítulo anterior). O produto final são os valores dos parâmetros que se desejava estudar, normalmente alguma característica difícil de se obter no sistema real. Aliás, as medidas do sistema real referem-se as técnicas de obtenção de dados estatísticos através de equipamentos específicos colocados sobre o sistema, ou muitas vezes, por variáveis mantidas pelo próprio sistema, que nesse caso são os equipamentos que formam os nós da rede.

A simulação é normalmente dividida de acordo com os modelos usados, podendo ser de estado contínuo ou discreto, indicando se as variáveis adotam ou não valores continuamente dentro de um determinado intervalo. Ainda pode ser classificada em relação ao tempo como contínua ou discreta, ou seja, muitos modelos dividem o tempo em "slots". Os modelos recebem diversas outras classificações de acordo com seu comportamento, tais como, probabilísticos ou determinísticos, dinâmicos ou estáticos, lineares ou não, abertos ou fechados, estáveis ou instáveis [BIH 99].

Mas, uma classificação [SOR 90] interessante é a divisão dos modelos discretos em orientados a eventos, orientados a processo ou orientados ao exame da atividade. A primeira delas é a mais comum e usualmente empregada na área de redes de comunicação, e compreende que a mudança de estado está relacionada a ocorrência de um evento. Por evento, entende-se um estímulo que ocorre no sistema a partir do qual ocorre alguma atividade, por exemplo, o recebimento de um pacote pelo roteador. Normalmente, modelos de rede são de estado discreto, tempo contínuo, probabilístico, dinâmico e não linear, mas são conhecidos somente como modelos orientados a eventos.

### *5.1.3 Comparação com outras Soluções*

A simulação é empregada para resolução de algum tipo de problema definido na fase de projeto, tal como acompanhar o desempenho de uma rede. É comum compará-la então com outro tipo de solução que pode nos fornecer a mesma solução. Normalmente, no caso das redes de comunicação, essa comparação é feita em relação às medições ou em relação às soluções analíticas.

A solução analítica é geralmente mais rápida, porém, costumam-se aplicar muitas hipóteses de simplificação para se modelar o sistema, o que o torna normalmente mais específico. Por outro lado, o resultado obtido é exato já que se trata da solução de equações matemáticas, mas esse valor pode não representar a realidade se não houve cuidados nas simplificações anteriores. Outro problema é quando a solução não é imediata e deve ser realizada através de técnicas de aproximações, ou ainda a limitação de parâmetros.

Como pode-se perceber, a simulação tem uso mais abrangente e pode ser aplicada em casos mais complicados, porque proporciona uma série de alternativas que podem ser comparadas. Infelizmente, as simulações podem falhar com frequência, e os principais fatores estão no desenvolvimento do programa e na escolha das técnicas estatísticas. Um dos principais motivos da utilização de simuladores é para avaliação de sistemas ainda não existentes, seja pelo custo da construção ou para fins de projeto.

Mas, pode ser também empregada para experimentações, a fim de determinar os limites de sistemas, onde seria necessário construção de protótipos.

A medição direta das redes reais é de longe a mais precisa de todas as técnicas, e oferece a visão necessária a respeito da qualidade de serviço provida. Através delas, pode-se fazer o planejamento de expansões e modificação de parâmetros da rede, porém, muitas vezes são necessários equipamentos caros e pessoas especializadas. Nesse dimensionamento deve-se também conhecer bem o ambiente de teste a fim de não generalizar casos específicos. Além do custo, a flexibilidade de manipulação dos parâmetros pode ser um fator limitante, seja por dificuldade de configuração ou pelo fato de estar em operação, o que durante os testes podem prejudicar o desempenho normal da rede. De qualquer forma, as conclusões e validade dos dados também ficam sujeitas a comprovação, haja visto que sempre existe a possibilidade de erros de campo não previstos.

O certo é que todas as soluções têm seu espaço, e no intuito de validar os resultados é sempre indicado que se usem duas técnicas distintas de forma a aumentar a garantia dos mesmos e não correr riscos de erros. É comum utilizar-se um modelo analítico e posteriormente realizar simulações para comparar os resultados, ou em outro caso, desenvolver um modelo de simulação e comparar seus resultados a medidas de redes reais. Um fator de escolha pode ser o custo de desenvolvimento, e se for assim, pode-se dizer que a solução analítica é a mais barata, a simulação seria intermediária e as medições as mais dispendiosas.

#### 5.1.4 Problemas e Desafios

Existem diversos problemas a serem vencidos para que a simulação possa ser considerada de confiança. No caso específico das redes ATM, cujos perfis de tráfego normalmente são muito complexos e variáveis, os problemas podem se tornar mais graves, correndo o risco da simulação ser ineficiente ou até mesmo inviável [JIR 2000]. Alguns exemplos de desafios a serem vencidos pelos simuladores são :

- as grandes redes hierárquicas e suas possíveis características que complicam o sistema, como por exemplo, o roteamento escalonável e tolerância a falhas. Geralmente, as simulações preferem modelos de pequenas redes pela sua simplicidade implícita;
- as redes multiprotocolos, uma vez que a tecnologia caminha na direção da interconexão de todo tipo de redes de comunicação, isto porque na simulação é menos complexo criar modelos para um protocolo de cada vez;
- a necessidade de longas escalas de tempo, com o intuito de permitir a estimativa apurada de probabilidades, evitando a interpretação incorreta de transitórios de longa duração, que podem ser originários de tráfegos dominados por correlações de tempo de longa duração (tráfego auto-similar). Além disso, as longas escalas garantem a simulação correta de ocorrência de eventos raros, como a taxa de perda de células em *switches* ATM, cuja probabilidade é muito pequena;
- a necessidade de estudos e técnicas para estimar com precisão o tamanho e número das amostras, além de outras variáveis estatísticas. É desta forma, dimensionar o

experimento em relação ao tempo necessário de simulação e garantir a convergência dos resultados.

Sobre a programação do sistema podem-se ainda discutir mais estes problemas:

- a verificação é muito importante, pois mesmo que sejam tomadas precauções, sempre podem existir erros de programação. Além disso, normalmente devem-se empregar analistas ou programadores que conheçam os princípios da simulação, e neste caso a experiência conta muito;
- a validação é outro problema enfrentado, pois é necessário que a simulação represente o sistema real e não contenha suposições incorretas ou tenha um comportamento atípico. Somente assim, pode-se ter confiança nos dados gerados;
- a geração de números aleatórios pode causar problemas, tendo em vista que é a partir destes números que são criadas as distribuições de probabilidade dos tráfegos da rede. É importante conhecer bem o gerador, e escolher as sementes com cuidado para manter a independência das amostras;
- o nível de detalhamento desejado no desenvolvimento do simulador é um fator limitante, pois dependendo do problema que se pretende solucionar; pode-se necessitar de muito tempo de programação, ou ainda um aumento da complexidade dos algoritmos, elevando assim, os custos do projeto.

## ***5.2 Aquisição e Análise dos Resultados***

Normalmente, sistemas de simulação possuem algum grau de incerteza associado aos seus elementos, por isso eles evoluem no tempo de modo imprevisível. Estes sistemas são conhecidos como estocásticos e requerem a utilização de conceitos probabilísticos para caracterizar seus elementos, e, por conseqüência, os dados de saída dependem de interpretações estatísticas.

### **A) Conceitos Gerais**

Às vezes pode-se alimentar um modelo de simulação com dados reais, porém, isto não é o mais comum, pois o processo de medição e gravação pode tornar-se caro, consumir muito tempo, ou quando não planejado de forma adequada, pode não ser representativo. Como exposto no capítulo 5, os simuladores são alimentados, na maioria dos casos, por modelos de tráfego que seguem alguma expressão matemática ou processo conhecido, porém, são gerados de forma aleatória.

Um outro fator importante para a correta simulação é a geração de números aleatórios, entretanto, o processo mais comum usado é empregar uma equação recursiva que, infelizmente, pode produzir uma seqüência de números determinística, sendo assim denominada de pseudoaleatória. A realidade estatística dos resultados da simulação depende do grau de aleatoriedade dos números, e, por isso, empregam-se testes para determinar esta característica, mas nunca existe garantia total de que a seqüência gerada seja apropriada.

Tanto na definição das entradas, quanto na coleta de dados do desempenho, a partir do modelo podem existir problemas na conversão dos dados para uma forma utilizável. Alguns destes tratamentos são benéficos para descrever características importantes, tais como tabelas de distribuição de frequência ou de frequência acumulada, mas deve-se tomar cuidado com a perda de informação devido ao agrupamento de dados.

Um importante conceito de estatística é a distinção entre a população e a amostra de uma variável aleatória. Por população compreendem-se todas as possíveis observações da variável e por amostra uma parte desta população. Normalmente, deseja-se obter os parâmetros da população, tais como, a média ou variância desta variável, e desta forma a partir das amostras (cada simulação), pretende-se inferir esses valores. As amostras de saída podem servir inclusive para identificar a distribuição da população, porém, neste caso é desejável a aplicação de um teste de hipótese, como o Qui-quadrado ou o Kolmogorov-Smirnov [BAN 95].

Existem vários tipos de erros ligados a simulação, alguns próprios do modelo ou lógicos, porém, os mais difíceis de se detectar são a estimação imprecisa dos parâmetros de entrada. Além disso, a própria simulação introduz erros concernentes à variabilidade dos resultados estatísticos produzida pelos números randômicos. Os transientes que ocorrem antes das variáveis atingirem seu estado de equilíbrio é outra fonte de erro no cálculo da média. E, por fim, os diferentes números aleatórios produzem diferentes resultados, o que leva ao problema da escolha da semente dos geradores pseudo-aleatórios para evitar o perigo do uso de um único conjunto de número.

## B) O Cálculo dos Parâmetros

Deve-se lembrar que o resultado das simulações são amostras observadas de variáveis aleatórias, e como conseqüência, qualquer assertiva deve levar em consideração essa variabilidade inerente, ou seja, cada simulação irá produzir saídas probabilísticas. Assim, a influência estatística do parâmetro desejado é normalmente obtida a partir da média e da variância das amostras, conforme as equações abaixo:

$$\bar{X}_I = E[X_i] = \frac{\sum_{i=1}^I X_i}{I} \quad \text{Var}[X_i] = E[(X_i - E[X_i])^2] = \frac{\sum_{i=1}^I (X_i - \bar{X}_I)^2}{I-1}$$

Se as amostras são independentes e identicamente distribuídas, conforme demonstrado em [SOR 90], pode-se assumir as seguintes simplificações para o cálculo do parâmetro:

$$E[\bar{X}_I] = E[X_i] \quad \text{Var}[\bar{X}_I] = \frac{\text{Var}[X_i]}{I}$$

Baseado em dois teoremas, pode-se caracterizar o comportamento de  $X_i$  quando o número de amostras  $I$  tende ao infinito. A lei dos grandes números estabelece que quando  $I$  cresce,  $X_i$  se aproxima de  $E[X]$ . E o teorema do limite central, mostrando que a distribuição da soma de  $I$  amostras se aproxima de uma distribuição normal para valores grandes de  $I$ . A partir destas considerações, podem-se aceitar as equações anteriores, desde que se possa determinar um intervalo de confiança. Este intervalo é

onde o parâmetro deve se encontrar, com certo grau de probabilidade denominado nível de confiança. Para um nível de  $(1 - \alpha)$  pode-se descrever o intervalo como sendo  $(\bar{X}_I - t_{\alpha/2, f} \cdot \sigma_{\bar{X}}, \bar{X}_I + t_{\alpha/2, f} \cdot \sigma_{\bar{X}})$  onde  $t_{\alpha/2, f}$  é uma distribuição "student" com  $f = I - 1$  graus de liberdade. Quando  $I$  tende ao infinito pode-se utilizar a distribuição normal  $Z_{\alpha/2}$  com média 0 e desvio padrão 1 no lugar da "student".

A expressão do intervalo de confiança apresenta o desvio padrão da média calculada, e este valor pode ser obtido por uma das equações abaixo:

$$\sigma_{\bar{X}}^2 = \frac{1}{I(I-1)} \sum_{i=1}^I (X_i - \bar{X}_I)^2 \quad \sigma_{\bar{X}} = \sqrt{\text{Var}[\bar{X}_I]}$$

Assim, calculando o desvio padrão e obtendo-se em uma tabela de distribuição normal o valor ( $Z_{\alpha/2}$ ) correspondente a precisão  $\alpha$  desejada, pode-se determinar o intervalo de confiança da média calculada.

### C) A Condição Inicial do Modelo

O estado de partida da simulação é de fundamental importância no resultado desejado, e pode depender do sistema modelado, ou ainda, se o interesse é avaliar o estado transitório ou permanente. As regras mais usadas para inicializar o sistema são: o estado "vazio"; o estado de maior ocorrência; e o de equilíbrio. A primeira dessas regras é, na verdade, o estado zerado das variáveis e poucas vezes representa o sistema real. A segunda, depende do conhecimento anterior do sistema para determinar o seu comportamento. A terceira é normalmente obtida pelas médias de outras amostras ou modelos analíticos, a fim de diminuir o transitório.

Independente do estado inicial do simulador, pode-se utilizar a técnica de truncamento de dados para reduzir qualquer tendência no resultado que seja proveniente dessas condições iniciais. O que ocorre realmente neste caso, é um retardo na coleta de estatísticas, pulando o período transiente, porém, deve-se determinar até qual ponto o descarte deve acontecer. Logicamente, quando se usam métodos como o ciclo regenerativo ou subintervalos (comentados no item a seguir), pode-se muitas vezes ignorar esses cuidados, devido ao fato do estado inicial de cada amostra ser o estado final da amostra anterior.

Alguns autores tentam determinar um período padrão para se descartar, apesar de isto depender fortemente do modelo sendo simulado. Em [BAN 95] foi sugerido que o tempo total simulado seja no mínimo dez vezes o tempo de descarte, isto porque um valor muito pequeno aumenta o número de amostras e conseqüentemente melhora a precisão da média, mas pode ter uma influência negativa no cálculo da variância, e se for muito grande afeta de forma inversa.

### D) Tempo de Simulação x Número de Amostras

Devido as dependências entre as amostras é necessário um grande número de simulações para estimar os parâmetros desejados. Neste sentido, podem-se citar quatro métodos básicos [SOR 90]:

- Replicação: emprega várias execuções separadas, onde cada uma serve para obter uma amostra da variável e calcula-se uma única média. Este método tem a propriedade da independência das amostras, e é o mais usado para análise de transitórios. Infelizmente, desperdiça tempo com cada execução.

- Subintervalos: neste método a execução é dividida em subintervalos e para cada um deles é calculado o valor da média da amostra. Assume-se a independência dos intervalos, porém, deve-se escolher bem a largura dos intervalos para não ter covariância entre as medidas. A vantagem é se realizar uma única execução longa para obter a estimativa.

- Ciclos Regenerativos: semelhante a anterior, utiliza uma única execução dividida, mas utiliza os estados da simulação para definir os ciclos, e cada um deles representará uma amostra. Normalmente inicia num estado de regeneração, ou seja, aquele que em geral o comportamento futuro dependa somente dele.

- Estimulação Espectral: utiliza a análise espectral para estimar a variância através da autocovariância das amostras. Também realiza-se numa única simulação longa e como os dois anteriores, reduz o tempo de execução e não necessita de uma preocupação com a escolha dos valores iniciais das variáveis do modelo.

Para obter resultados mais precisos, o desejado seria um grande número de amostras de comprimentos longo, mas isso, normalmente, demanda muito tempo. Caso utilizem-se seqüências de curto comprimento, pode ter uma maior polarização na estimativa do parâmetro devido às condições iniciais. Contudo, a redução do número de amostras pode piorar o cálculo da variância da média, devido ao fato de se ter um conjunto pouco representativo da população.

A literatura sugere um mínimo entre 25 a 30 subintervalos em cada replicação, isto, porque acima deste valor pode-se melhorar o cálculo da média, mas piorar o cálculo da variância, e abaixo dele ocorre o inverso. De qualquer forma, se existe possibilidade de incrementar o tempo de simulação, normalmente aconselha-se manter o mesmo número de subintervalos, porém, maiores.

### **E) Tempo de Parada Automática**

Com o intuito de encontrar o final ideal da simulação, podem-se empregar vários métodos. O objetivo é encontrar um ponto de parada sem prejuízo para a estimativa da medida e sem consumir longos períodos em replicações. O que se costuma fazer é determinar um tempo de duração para a simulação, mas, dessa maneira pode-se ter amostras de tamanhos diferentes ou que ainda não convergiram para próximo do valor esperado. Outro método simples é parar quando se tiver atingido determinado tamanho de amostra que seja considerado suficiente.

Uma alternativa de parada é especificar o número de entidades processados no sistema, mas é importante assegurar que neste estado do modelo não exista nenhum comportamento tendencioso, para garantir que a amostra seja significativa. Mas, realmente, são os métodos de parada automática que facilitam a determinação do tempo ideal, pois manipulam os resultados durante a execução a fim de calcular quando a estimativa se encontra dentro da tolerância desejada, sendo, por isso, bastante empregados.

A parada automática normalmente é feita através de uma comparação feita por uma equação de teste para verificar se o erro obtido está dentro do intervalo de confiança. Pode-se considerar a fórmula abaixo para este teste, onde  $\varepsilon$  representa o erro calculado na replicação R :

$$\frac{t_{\alpha/2, R-1} S_o}{\sqrt{R}} \leq \varepsilon$$

## F) Os Testes Realizados no Simulador NIST

Infelizmente, o simulador NIST não tem um módulo para cálculo da inferência estatística. Tudo que o pacote apresenta são alguns scripts em shell para filtrar um parâmetro, obter a média, valor máximo, mínimo, e alguns outros. Nada como cálculo de intervalos de confiança, desvio padrão. Não existe forma de determinar uma parada automática a não ser pelo tempo. Porém, existe como determinar um período de tempo antes da aquisição de dados para o arquivo de log, que é determinado na linha de comando pela opção "-w warmuptime" onde este tempo deve ser em microsegundos.

Utilizou-se a geração de logs a partir de um período inicial de quatro segundos para estabilizar os resultados, e depois, adotou-se o princípio de carregar os dados para a planilha de dados "excell" e calcular os valores necessários. Como recomendado na documentação do NIST, utilizou-se um tempo de simulação máximo de 42 segundos, porém, esta limitação não é realmente necessária, pois o simulador não apresenta problemas acima deste período, o que parece ocorrer é que os números randômicos podem se repetir.

Por opção, utilizou-se cada simulação como uma amostra para a obtenção da média e variância. Mas, infelizmente, devido ao tempo curto para realização das simulações foram realizadas menos replicações do que se desejava. Apesar disso, o intervalo de confiança pode ser calculado com certo cuidado, e dentro de uma precisão de 95 % pode-se verificar que o intervalo é razoavelmente próximo da média.

## 5.3 A Escolha do Simulador

De forma a obter nossa própria análise do protocolo AAL2, é interessante ter como estudá-lo sem que realmente seja necessário construir uma rede e utilizar aplicativos que façam uso dele, pois isto seria dispendioso. Mesmo que se tenha disponibilidade de uma rede ATM, a monitoração dela pode não ser tarefa fácil. Como visto nas seções anteriores, é prática comum se procurar simular as redes de comunicação com o intuito de se obter mais informações e poder comparar alternativas.

Na pesquisa do simulador não foi feita uma busca exaustiva desse tipo de ferramenta, apenas foram analisadas algumas linguagens de uso específico e informações disponíveis na INTERNET sobre programas já desenvolvidos. Por fim, foi escolhido o simulador de rede ATM/HFC que se apresentou como sendo propício devido suas qualidades. A escolha deveu-se a diversos fatores, mas o fato do simulador já estar em uso pelo LABCOM da UFRGS pesou muito, pois facilita a troca de conhecimento e agrega valor. Outro fato importante é a garantia de um programa desenvolvido pelo Instituto Nacional de Padrões e Tecnologia (NIST) dos Estados

Unidos, que tem diversas publicações e pesquisas na área de redes ATM, além de aparentemente ainda manter o desenvolvimento deste simulador (versão 4.1).

O simulador de rede ATM/HFC proporciona um ambiente de teste flexível para estudos e avaliação de redes ATM e HFC. É uma ferramenta bastante útil para modelar e testar redes, permitindo que se possa monitorar alguns aspectos através de medidores gráficos ou logs. No caso de necessitar novos componentes, o código fonte é fornecido e algumas informações interessantes para modificar os módulos estão disponíveis. É empregada a técnica de simulação orientada a eventos, o que o aproxima da literatura, pois é bastante utilizada e eficiente. Foi desenvolvido em linguagem C, apesar de a programação não seguir os padrões das linguagens orientadas a objeto, existe uma tentativa de se aproximar com o emprego de um conceito de estruturas para cada entidade, mas que a torna um pouco complexa.

Um fator importante na simulação de qualquer rede é a modelagem do gerador de tráfego, que como foi visto, está intimamente ligada a caracterização da aplicação e a qualidade de serviço desejada. O simulador do NIST apresenta uma boa quantidade de fontes de tráfego e possibilita a criação de novas. Sendo o objetivo principal simular e estudar o comportamento de um protocolo que foi criado para carregar sinais de voz comprimido, a fonte de tráfego que buscamos é do tipo VBR de baixa taxa de bits. Segundo estudos verificados nas seções anteriores, um dos modelos mais empregado para fontes de voz e que apresenta uma precisão variando entre satisfatória e boa, é o modelo ON/OFF exponencial, que está presente neste simulador (apesar de algumas discordâncias na documentação).

Por ter sido desenvolvido para plataformas UNIX e utilizar o Sistema X Windows, apresenta as facilidades de uma interface gráfica. Apesar deste tipo de interface acarretar em mais tempo de processamento durante as simulações, ela tem uma vantagem importante, pois proporciona a visualização rápida da estrutura de rede durante a fase de criação e por isso facilita principalmente a utilização didática da ferramenta.

Depois de discutidas as razões da escolha do simulador NIST, pode-se dizer que é um bom começo para quem pretende simular um novo protocolo, principalmente, por que seria impraticável criar um novo ambiente de simulação com essas qualidades e em curto prazo de tempo, seja a partir do C ou qualquer outra linguagem específica.

#### **5.4 Descrição do simulador**

O simulador é visto pelo usuário como um conjunto de *componentes* trocando mensagens entre si. Estes componentes representam várias entidades da realidade de uma rede de comunicação, por exemplo, os switches ATM, o link, o equipamento terminal, etc. Também as entidades não concretas são representadas por componentes, como as aplicações ATM, responsáveis por gerarem o tráfego de informação, que alimentam a rede, e o eliminam ao final do seu curso. Veja a figura 5.1 .

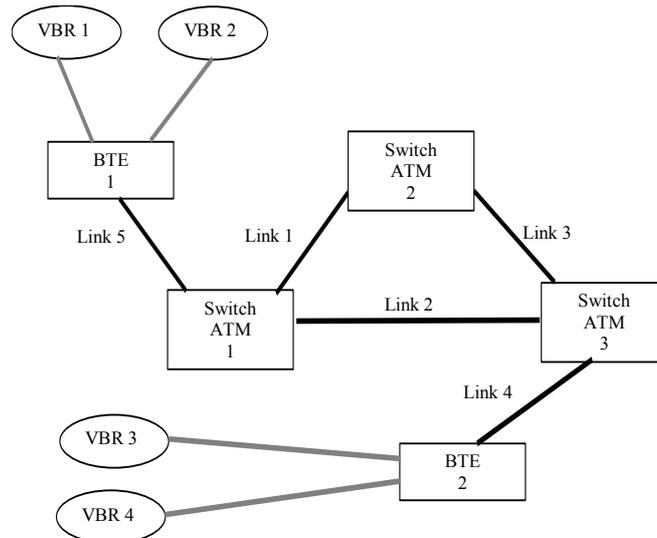


FIGURA 5.1 - Exemplo de rede ATM criada no simulador

Por trás destes componentes, existe a parte não visível do simulador, é aquela que executa as tarefas programadas para cada um deles, controla o tempo e sincroniza os eventos. Cada componente é feito de uma série de rotinas que descrevem sua ação na rede, e são reunidas num único arquivo que a partir de agora chamaremos de módulo. Além desse conjunto de módulos de componentes, existem outros que são internos do simulador e realizam as tarefas de comunicação e controle de todos os componentes.

É preciso entrar em detalhes de algumas partes do simulador e explicar como se processam certas atividades com o intuito de compreender algumas das modificações realizadas no código fonte original. Merecem destaque a descrição do "gerenciador de eventos" e do "mecanismo do tempo" que são o coração do simulador, e assim sendo ficam em atividade durante todo o tempo de execução.

Na criação de um novo módulo para adicionar um novo componente ao simulador, deve-se observar a interação com o demais módulos já existentes. Isto ocorre por meio da construção de rotinas cuja interface já está pré-estabelecida e que todo módulo deve conter. Esta interação também pode ser realizada por meio de variáveis globais que, normalmente, são usadas para guardar alguma situação atual do simulador. Assim, essas rotinas ficam responsáveis pela inicialização, comunicação, e modificação do estado do componente, mas não devem influenciar na lógica central do simulador.

Outro importante ponto são as rotinas de saída e a geração dos relatórios, que no fundo são os produtos esperados da simulação. É a partir deste ponto que se iniciam as avaliações e por essa razão é indispensável sua discussão aqui.

#### 5.4.1 O Tempo e o Gerenciador de Eventos

Na teoria, pode-se entender o simulador como uma máquina de estados definida pelo valor das variáveis do sistema. Cada mudança nas variáveis define um novo estado do sistema, e esta transição para esta nova situação é realizada por um evento. Logo, eventos são ações internas ao simulador com o intuito de modificar o seu estado. Este

simulador é controlado por eventos e a interação entre os componentes deve ser expressada desta forma, ou seja, através da troca de eventos.

O gerenciador de eventos, também conhecido por escalonador de eventos, é responsável pela sincronização das tarefas. Nele existe uma fila de eventos ordenada cronologicamente, esperando para ocorrerem na simulação em um tempo futuro. A variável que contém o valor do tempo simulado chama-se de relógio de simulação. Conforme pode-se observar na figura 5.2, o gerenciador de eventos controla a ordem de disparo de cada evento de acordo com o relógio do simulador (1). Quando ocorre o disparo, a execução é transferida para a rotina responsável por tratar aquele evento (2), que por sua vez irá gerar novos eventos, os quais deverão ser colocados na fila (3).

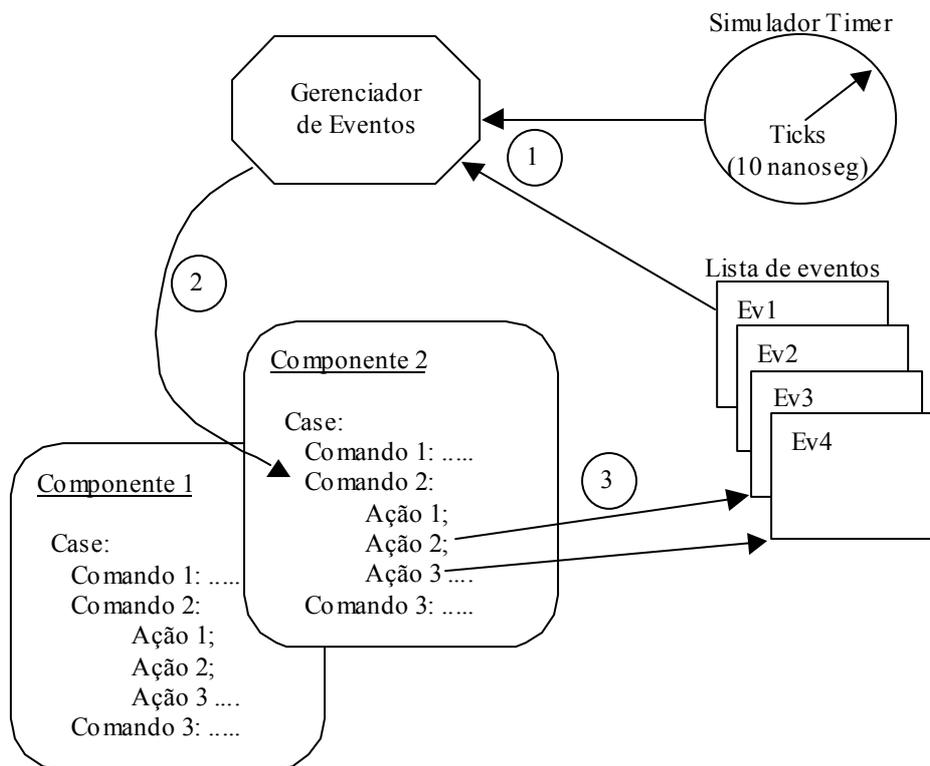


FIGURA 5.2 - Esquema representando o escalonamento de eventos no simulador

Em primeiro lugar, é importante ressaltar que este simulador é orientado a eventos discretos, significando que cada nova transição de estado somente ocorre nos momentos dos eventos. Durante o tempo entre os eventos, o simulador permanece no mesmo estado. O tempo é também discreto para nosso simulador, porém, isto não significa que a continuidade do tempo esteja ligada a continuidade de estado e vice-versa.

A divisão do tempo é feita em unidades de "ticks" e é mantido em um inteiro sem sinal (32 bits). Assim, de acordo com a documentação do NIST, para o valor padrão do "tick" que é de 10 nanosegundos, pode-se chegar a um total de 42 segundos de simulação. Este valor não pode ser mudado pelo usuário normal do simulador, somente no código fonte durante a compilação. Porém, se aumentar o valor do "tick" significa mais tempo de simulação também significa uma imprecisão maior devido ao espaço de tempo entre dois estados consecutivos da simulação.

O tempo de simulação é o elemento que determina a seqüência de eventos. O escalonador é responsável pelo avanço deste tempo, que segue a política do tempo unitário, ou então adota a abordagem movida a eventos, isto é, avança diretamente para o tempo do evento mais próximo do tempo atual. Assim, se em determinado momento não houver eventos com tempo de ocorrência menor ou igual ao do relógio do simulador, então ele será atualizado para o tempo do próximo evento (aquele que está agendado para ocorrer antes na fila cronológica) e quando ele for disparado será eliminado da fila. Os eventos podem ser agendados para o mesmo tempo, porém, não há garantia da ordem em que eles ocorrerão. Se ocorrer agendamento de eventos para o tempo passado, serão executados no presente, porém são considerados ilógicos.

Este avanço do tempo é realizado com o intuito de acelerar a simulação e não causa problema algum desde que se tenha em vista o caráter discreto do estado da simulação que é considerado constante no intervalo. Isto torna-se mais insignificante em se tratando de variáveis também discretas, tais como a quantidade de pacotes de determinado ponto da rede. O método do tempo com incremento unitário e constante, normalmente não é utilizado, pois causaria grande ineficiência.

#### 5.4.2 Rotinas de Tratamento de Eventos

Existem três classes de eventos que são os comandos, os regulares e os privados. Os comandos são responsáveis pelo funcionamento do simulador e sua interface gráfica, tendo menos importância durante a simulação. Os eventos regulares são aqueles necessários para a troca de informação entre os componentes e merecem bastante atenção, por exemplo, o evento "ev\_busy" que define o estado de ocupado do componente para os demais. Já os eventos privados são destinados ao próprio componente, e por isso, acabam sendo bastante usados, mas interferem pouco no estado dos outros componentes.

Todo componente em seu módulo de programação tem uma rotina chamada *action* que é responsável por chamar corretamente a próxima rotina de acordo com o tipo de evento enviado. Assim, se o evento disparado chama um determinado componente com o parâmetro de tipo de evento sendo "ev\_receive", então o gerenciador passa o controle a esta rotina que se encarrega de chamar a seguinte responsável pelo recebimento de pacotes. O ciclo se fecha quando as rotinas da programação do componente agendam novos eventos [figura 5.2].

Todas estas rotinas modificam o estado do componente e, conseqüentemente, o estado do simulador através das suas variáveis globais. As variáveis internas devem ter seus valores definidos no início da simulação através da rotina de inicialização que é chamada pelo evento de comando do tipo "ev\_reset". Algumas delas são definidas como parâmetro e neste caso, são lidas do arquivo de configuração da rede, mas as normais iniciam com seus valores padrão.

Outra rotina importante é disparada pelo evento de comando do tipo "ev\_create" de criação do componente, que é a responsável por armazenar espaço na memória dinâmica para o componente e suas variáveis. Esta gerência de memória dinâmica é uma importante parte na programação em C, pois pode gerar graves problemas como estouro de memória se mal realizada. Durante o processo de simulação algumas

entidades, tais como os pacotes, devem ser criadas e destruídas, assim deve-se estar sempre limpando o espaço que foi utilizado por elas.

### 5.4.3 *Conexão entre os Componentes e as Rotas*

Entre as rotinas necessárias em cada módulo de programação de um componente, existem algumas específicas para o funcionamento da rede ATM, sendo praticamente obrigatórias. Elas são disparadas por eventos de comando durante a criação do diagrama da rede quando estamos conectando um componente a outro.

O primeiro evento é o de adicionar um vizinho ("ev\_neighbor"), que serve para indicar quando um componente tem uma conexão física com outro, gerando uma lista de vizinhos. A adição à lista ocorre sempre que se coloca uma ligação no diagrama unindo dois componentes, e da mesma forma ao retirar-se a ligação remove-se o componente da lista.

O outro tipo de evento ("ev\_make\_route") é gerado durante a criação de uma rota no diagrama da rede. Naturalmente, toda comunicação ATM é baseada nos VPCs que estabelecem rotas fixas entre as aplicações, porém, o simulador NIST não foi preparado para simular o estabelecimento de VPCs entre os BTEs, sendo então necessário no processo de desenvolvimento do diagrama que o projetista indique quais são as rotas. Ele inicia a rota clicando sobre uma aplicação, depois sobre o próximo componente da rota, e assim por diante até chegar a aplicação destino.

Fica claro que a lista de vizinhos gerada na primeira etapa do diagrama é utilizada pelas funções internas do simulador para verificar a coerência da rota sendo indicada, e logicamente, as rotas inválidas não são aceitas antes de iniciar a simulação. Se não houver rotas saindo de uma aplicação podem ocorrer alguns problemas inesperados no simulador NIST e, infelizmente, não existe forma de visualizar facilmente a falta delas.

### 5.4.4 *O Registro das Saídas*

Toda simulação deve ter no mínimo um tipo de registro das variáveis consideradas importantes para que possa ser analisada ao final do processo. Neste caso, existem duas formas de visualização dos resultados, uma através de arquivo de LOG e outra através de medidores ("meter") na interface gráfica. Independente da forma escolhida para a saída, somente alguns dados podem ser visualizados, e isto depende da escolha realizada durante a programação do módulo.

Cada componente na sua criação é composto de diversas variáveis para seu funcionamento correto, mas podemos separá-las em três grandes grupos: as variáveis comuns, os parâmetros de configuração e os parâmetros de saída. Os parâmetros normalmente se diferenciam das variáveis comuns por possuírem algum significado no mundo real, tal como a largura de banda utilizada por determinado link.

Como citado anteriormente, os parâmetros de configuração são pré-definidos pelo usuário no arquivo de configuração que é lido pelo simulador e provavelmente permanecem fixos durante todo o processo. Porém, os parâmetros de saída somente

recebem algumas flags na criação do arquivo de configuração, que servem para indicar se devem gerar logs, ou serem visualizados através dos medidores gráficos, ou quem sabe ambas as alternativas. Estes flags são simples caracteres lidos na inicialização, tal como "2b" ativa o log e "2a" desativa, porém, para os medidores existem algumas informações extras a serem lidas, como as dimensões do quadro a ser exibido e o tipo de medidor (binário, de barras, histórico no tempo, etc...).

Apesar dos medidores serem mais fáceis para acompanhar o que está ocorrendo na rede simulada, eles não permitem uma forma de registro permanente, a menos que se use algum outro recurso de aquisição da tela gráfica do linux. A imagem gráfica pode trazer uma visão geral do comportamento do parâmetro, porém, é ruim na determinação de valores precisos.

Além disso, para o uso dos medidores é importante conhecer bem o tipo do dado real observado para a escolha do tipo de representação mais adequada, e ainda saber como e quando são gerados esses dados para não correr o risco de visualizar gráficos errados. Um exemplo é o caso do valor de utilização da largura de banda no componente "Link", que é uma variável calculada a cada 10 pacotes recebidos e quando visualizada com um diagrama de histórico no tempo de uma escala maior acaba sempre parecendo constante.

Assim, o método tradicional de logs sempre é preferido ao anterior, gerando um arquivo com uma lista de valores e o tempo em que eles ocorreram. Com esta lista em mãos, fica mais simples gerar os gráficos necessários posteriormente ou fazer qualquer outra análise que se faça necessário. Junto ao simulador NIST existem alguns scripts para retirar do arquivo de logs os dados mais comuns necessários, tais como o valor máximo, o mínimo, a média, a soma, etc...

## ***5.5 Criação do Componente AAL2***

O simulador desenvolvido pelo NIST não vislumbra a camada de adaptação do protocolo ATM. Na sua proposta, o simulador apresenta o equipamento terminal de banda larga (BTE), ligado aos switches através de links, servindo de interface diretamente para as aplicações ATM. Essas aplicações geram e consomem o tráfego que circula na rede (veja a figura 5.1), sem se preocuparem com a camada AAL. Algumas aplicações é que, dependendo da sua natureza, geram o tráfego de acordo com modelos propostos na literatura, e o simulador se encarrega de mostrar como ele se comporta na rede sem que exista preocupação com as camadas superiores a de rede.

Na proposta de simular o AAL2 a primeira idéia que surge é de modificar o equipamento terminal para se comportar seguindo as normas dos protocolos de adaptação, porém, isto iria requerer modificações drásticas no módulo que define o BTE criando o BTE-AAL2. Mas essa idéia, apesar de viabilizar a simulação do AAL2, não permitiria que um equipamento terminal pudesse ter mais de um protocolo AAL, o que restringiria as possibilidades de simulações futuras. Veja o caso (1) da figura 5.3 .

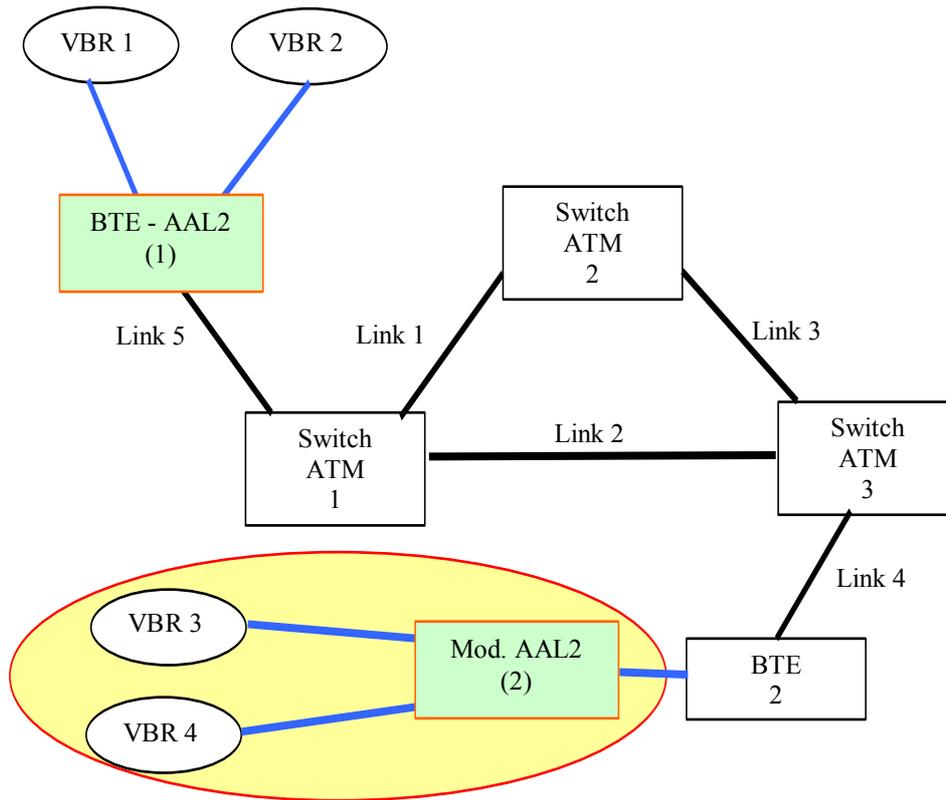


FIGURA 5.3 - Duas idéias propostas de modificação do simulador

A idéia desenvolvida foi a segunda (caso 2), onde se criou um novo módulo intermediário. Para o resto da rede ATM assemelha-se a um gerador de tráfego ou aplicação, mas para as aplicações ele recebe o tráfego gerado como se fosse um BTE. Esta proposta não tem o problema anterior e permite que o BTE receba também o tráfego direto de outras aplicações, mas exigiu algumas modificações na estrutura de classes de componentes do simulador original.

Existem 4 classes específicas na estrutura de rede do simulador, são elas : "switch\_class", "bte\_class", "connection" e "link". Só são consideradas links as ligações entre BTE e switch ou entre switch e switch, pois as ligações entre BTE e as aplicações não representam um componente da rede. A classe conexão ("conection") é utilizada para as aplicações e representa a fonte geradora de tráfego, logo nosso componente AAL2 deve ser desta classe. Porém, as aplicações ligadas ao novo componente devem ser de uma nova classe que foi denominada subconexão ("subconnetion").

Desta forma, evita-se de modificar qualquer parte no módulo do BTE, porém, devem-se criar novos módulos de aplicações específicos para o uso com o novo componente. Isto não é tão ruim, porque um dos propósitos expostos na documentação do simulador original, é a possibilidade de criarem-se novas fontes de tráfego e assim testar modelos propostos. O único cuidado adicional que surge após as mudanças é o de iniciar a criação a partir de uma das classes; dependendo se a simulação desejada pretende testar a camada AAL ou não.

### 5.5.1 Modificando as Aplicações

Para realizar os testes necessários ao componente AAL2 foi criado um módulo para um componente gerador de tráfego VBR denominado "VBRSubConnection" que, na verdade, é a modificação do módulo existente "VBRConnection". A principal mudança realizada no módulo VBR foi a criação de alguns parâmetros novos, e nas rotinas de envio e recebimento de mensagens que antes continham uma interface para um componente BTE, e agora são para um componente AAL2.

Outro ponto importante, no simulador, é que a tentativa de ligação deste novo componente VBR diretamente ao BTE deve ser recusada, isso é realizado através de uma pequena modificação na rotina de adicionar componentes vizinhos (evento "ev\_neighbor"), comentada anteriormente.

### 5.5.2 Mudando os Pacotes e as Rotas

Mais duas modificações foram necessárias para não alterar o princípio de funcionamento do simulador original, uma no que se refere a criação de rotas entre as aplicações e outra na utilização de "pacotes" na comunicação entre o componente AAL2 e as subconexões. Para entender as modificações deve-se levar em conta que a camada AAL2 em primeiro lugar, utiliza um VPC estabelecido para transportar vários usuários das camadas superiores, e em segundo, que recebe destes usuários pacotes de tamanhos variados e os trata para gerar o fluxo de células ATM.

A primeira alteração é, simplesmente um cuidado a mais na forma de criar a rota, e deste modo não foi necessário realizar mudanças na programação já existente, a qual realiza esta tarefa. Assim, cria-se a rota de um componente AAL2 a outro AAL2 da mesma forma como se realizavam as rotas anteriormente, pois o AAL2 é da classe conexão, ou seja, a mesma das aplicações originais do simulador. Pronto, está criado o VPC, mas para que uma das novas aplicações possa estabelecer seu destino é necessário colocar no parâmetro "cn\_user\_id", um identificador único e igual nas duas extremidades (bidirecional), é o equivalente ao identificador de canal do AAL2. Veja a figura 5.4 com um exemplo de rota correta entre o VBR2 e o VBR3, como também um caso errado entre o VBR1 e o VBR4.

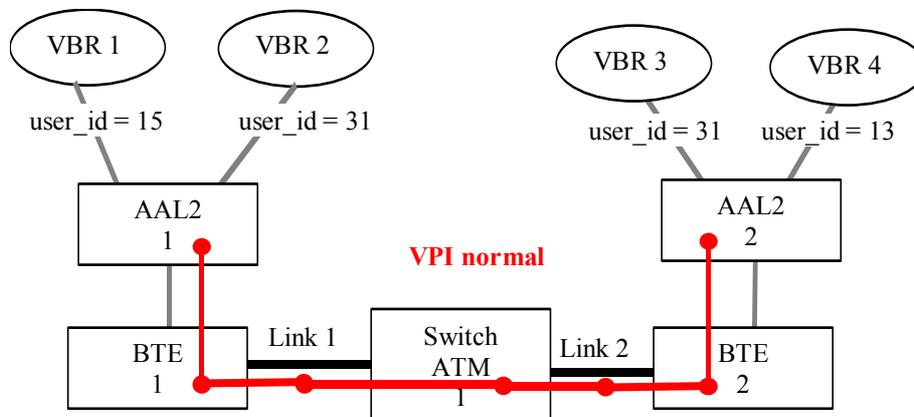


FIGURA 5.4 - Adição de rota entre as aplicações

A segunda alteração foi mais complexa, pois exigiu a criação de algumas estruturas de dados novas para montar o fluxo de comunicação entre as subconexões e o componente AAL2. A estrutura normal de mensagem utilizada é a "célula", e é definida num módulo interno do simulador. Mas pouca coisa se parece com a célula do ATM, aliás, é uma estrutura complexa que serve de maneira genérica às várias aplicações existentes.

Seguindo alguns casos semelhantes, acrescentou-se um apontador na estrutura "célula" para que seu "payload" carregue uma estrutura chamada de "PDU", possibilitando desta forma, que a informação específica do nosso modelo possa trafegar pelo simulador sem interferir na lógica dos demais componentes do simulador. O componente AAL2 inicial gera uma célula e a carrega com um apontador para a estrutura que realmente nos interessa, e ao final do percurso outro componente AAL2 retira o apontador e trata nosso "PDU".

A estrutura "PDU" não pode ser criada na forma tradicional com campos fixos para o cabeçalho e deixando de fora a carga do pacote, porque diferente de outros protocolos, o AAL2 tem cabeçalhos que podem estar em qualquer ponto da carga da célula. Desta forma, foi necessário criar-se a estrutura ocupando 48 octetos, mesmo que não sejam utilizados para a carga de dados reais, ocupando-se mais memória do que normalmente seria necessário durante a simulação. Mas, por outro lado, aproximou a programação do módulo AAL2 do que seriam os algoritmos reais propostos na norma.

Na comunicação do AAL2 com as subconexões foram criadas outras duas estruturas chamadas "pacote" e "SDU". O "pacote" tem formato semelhante a "célula", porém, bem mais simples, e o "SDU" é idêntico ao "PDU", mas com 64 octetos (tamanho máximo usado no AAL2). As aplicações enviam diversos "pacotes" que carregam "SDUs", que após a recepção na camada AAL2 são agrupados ou particionados antes de serem colocados nos "PDUs". Assim, eles trafegam pela rede, sendo recebidos do outro lado, e novamente tratados para formar os novos "SDUs", e serem entregues às aplicações através de novos "pacotes". A estrutura "SDU" é a única estrutura realmente desnecessária (normalmente vazia), porém, manteve-se para a clareza e igualdade com as especificações da norma e talvez um possível uso futuro.

É na estrutura "pacote" que existem os campos fixos para carregar o parâmetro "cn\_user\_id" necessário na identificação do canal AAL2 e realização do roteamento nesta camada. Caso o identificador seja inexistente o pacote é descartado conforme a norma.

### 5.5.3 *O Algoritmo do AAL2*

A programação do módulo que simula o protocolo AAL2 foi implementado o mais próximo possível do algoritmo real, ou seja, é totalmente baseado na norma do ITU-T [ITU 97], principalmente nos diagramas SDL fornecidos (veja o anexo 2). Por exemplo, as subrotinas de transmissão seguem a variável de estado que indica em qual dos quatro estados o componente se encontra, e desta forma, todas as outras variáveis se assemelham às originais inclusive nos nomes.

Existem mais subrotinas do que normalmente há nos módulos pré-definidos do simulador, mas o intuito é de simplificar a compreensão mais do que permitir a

reutilização das partes do programa. Também existem muitas variáveis temporárias, pois é necessário manter os dados dos pacotes que estão sendo remontados na recepção ou particionados na transmissão.

Além do algoritmo normal do AAL2, foi implementada uma subrotina chamada "cn\_manager" que simula a transmissão dos pacotes de permissão (conhecidos na norma como *MAAL-SEND.request*) da camada de gerenciamento e que fazem o controle do tráfego da conexão ATM. Essa subrotina envia esses pacotes na velocidade determinada pelo usuário no parâmetro de configuração *cn\_bit\_rate*.

Outras subrotinas estranhas ao diagrama SDL, mas necessárias, são as que controlam a parte referente ao relógio de uso combinado, que é o responsável por garantir o tempo mínimo desejado nas aplicações de tempo real do protocolo AAL2. O parâmetro de usuário *Timer\_CU* determina o tempo desejado, e cada vez que este expira ocorre uma mudança de estado na rotina de transmissão; levando a célula a ser transmitida mesmo que não esteja totalmente preenchida. Para realizar este relógio, foi utilizado o gerenciador de eventos programando um novo evento para avisar quando o contador *Timer\_CU* deve expirar, e neste momento comparar as variáveis do relógio para saber se o aviso é correto ou foi descartado (quando o pacote já foi enviado).

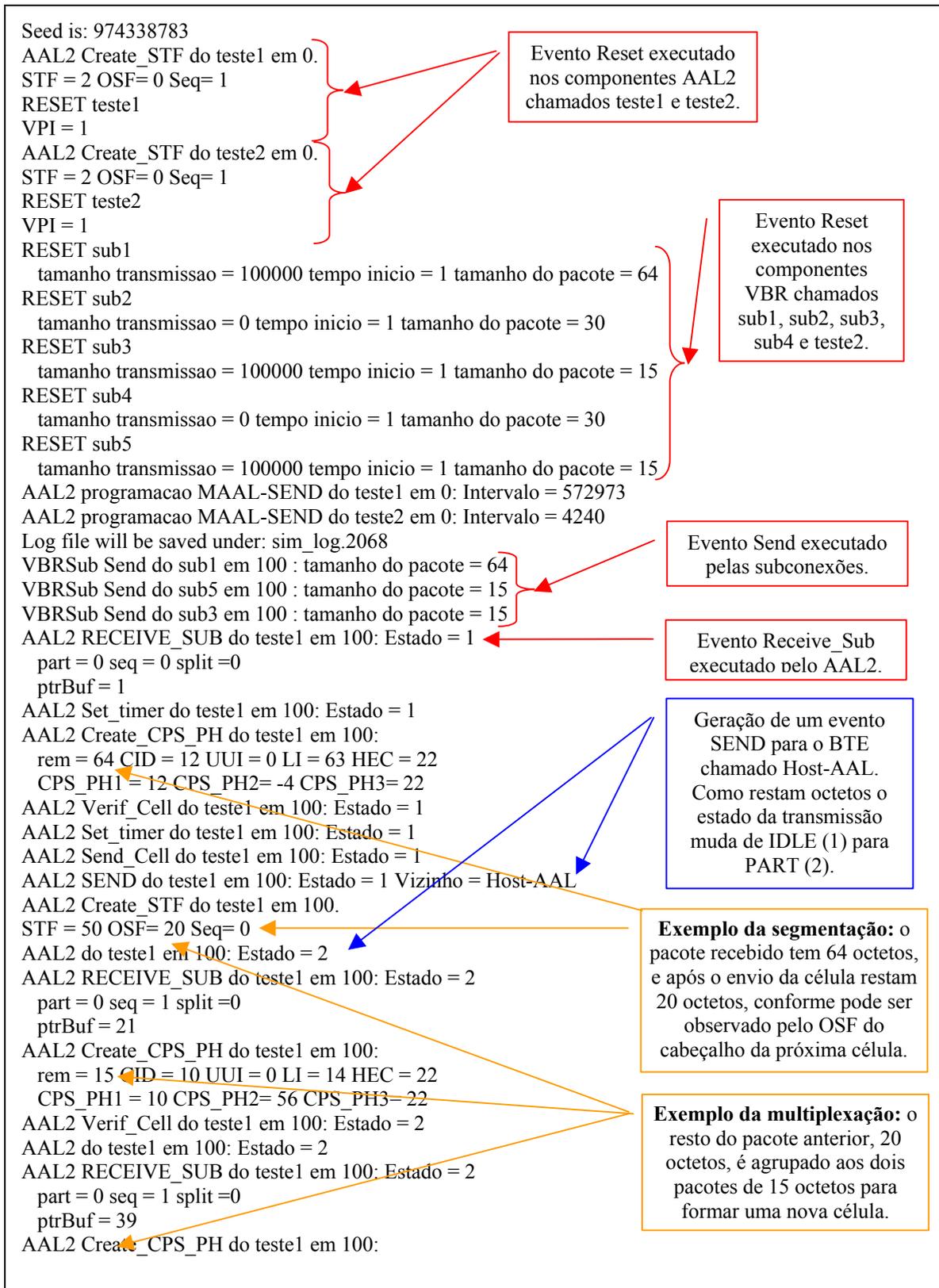
#### 5.5.4 Verificação do Módulo

Algumas dificuldades são encontradas em toda nova programação e nesta não foi diferente, tais como, os problemas de gerenciamento dinâmico da memória e a depuração de erros. Mas, sem dúvida, a pior parte foi a compreensão do código fonte, porque foi feito em linguagem C e sem o uso correto da orientação a objetos. Isto resulta no uso de um grande número de ponteiros e outras estruturas complexas, que foram criadas para facilitar a passagem de parâmetros, e também do controle de execução entre o gerenciador de eventos e os componentes.

Como comentado, a gerência da memória teve de ser realizada de forma cuidadosa, por falta de uso de uma ferramenta de programação mais atual. Mas, isso não deve-se tanto aos novos componentes que são criados no início da simulação e destruídos ao final, e sim, principalmente por causa da criação da estrutura "pacote". Esta estrutura, assim como a da "célula", deve ser gerada durante todo o processo sempre que houver o envio de tráfego das aplicações, para desaparecer ao chegar no seu destino final. Um descuido, e pode-se ter alguma rotina ocupando ou liberando memória indevidamente, mas o pior é que este erro pode ser de difícil localização se esta rotina estiver sendo chamada entre tantos comandos e estruturas de controle de laços.

Por isso, além do uso da opção "DEBUG" para compilação, foi criada outra opção específica para os novos módulos de programação que foi denominada "LOGAAL2". A principal razão foi separar as mensagens enviadas pelo depurador normal das enviadas pelo módulo do componente AAL2, tornando mais fácil a leitura do arquivo de saída. Na verdade, foi utilizado este tipo de mensagem em grande quantidade, para além de depurar, servir também como uma forma de realimentação e conhecimento do funcionamento do algoritmo do AAL2. Logicamente, não é aconselhado manter esta opção de compilação ativa quando se necessita de velocidade, devido ao consumo extra de CPU dos comandos incluídos.

Na figura abaixo, pode-se ver um exemplo da utilidade do "debug" e como é possível acompanhar a passagem pelas subrotinas internas do módulo; verificar a mudança de estados e conhecer o valor das principais variáveis em cada local onde normalmente elas são necessárias. Também pode ser visto o valor do relógio interno do simulador (em ticks) e desta forma saber se os tempos estão dentro do esperado para as taxas e parâmetros escolhidos no arquivo de configuração.



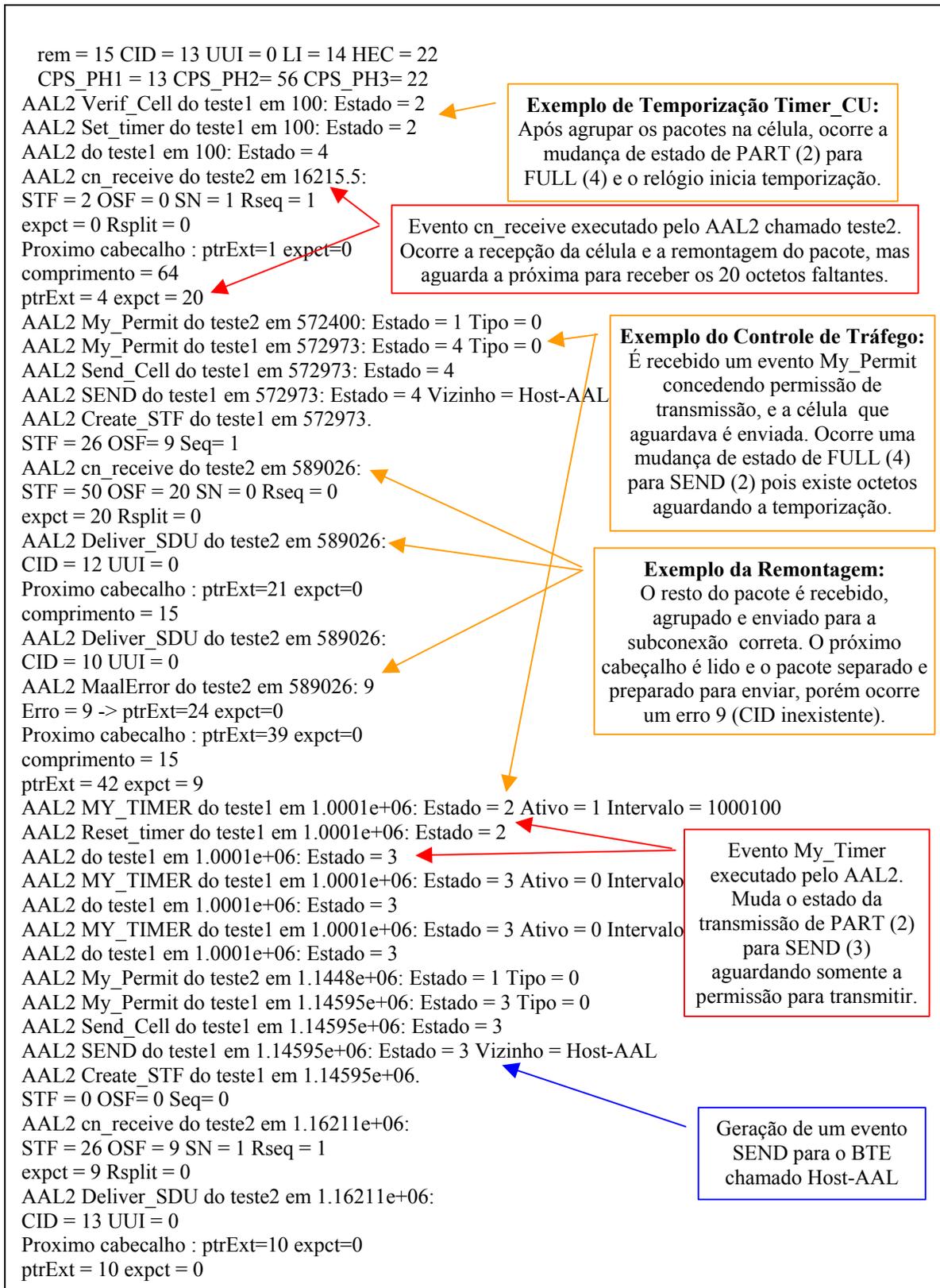


FIGURA 5.5 - Exemplo de saída do depurador LOGAAL2

## 6 *Teste e Validação*

Este capítulo é o centro de todo trabalho, tendo em vista o estudo e a pesquisa até então comentados, somados ao esforço de programação realizado, que serão aqui, discutidos. Pretende-se, conhecer algumas técnicas empregadas para verificação e validação de programas, depois demonstrar que o simulador está com a programação correta. A partir daí, será exposta uma série de comentários sobre artigos com o intuito de conhecer outros estudos sobre o AAL2, para então comparar com os dados do simulador e tentar validá-lo. No final, serão discutidos alguns problemas da simulação e qual sua influência nos dados.

### 6.1 *Técnicas de Verificação e Validação*

O motivo principal destas técnicas é garantir a confiabilidade dos modelos desenvolvidos, para que engenheiros e analistas possam fazer uso dos resultados obtidos e a partir deles projetem ou gerenciem algum recurso. É necessário, portanto, que o modelo tenha credibilidade. Quando possível, é importante que os futuros usuários participem das várias fases de desenvolvimento a fim de reduzir o grau de ceticismo sobre a validade do modelo.

Primeiro deve-se diferenciar conceitualmente a verificação da validação [SOR 90]. A verificação diz respeito a construção do modelo corretamente, ou seja, se a representação computacional representa as idéias concebidas. É a observação na busca de erros de implementação, tais como parâmetros de entrada errados ou lógicas de estrutura incorreta. Já a validação refere-se a escolha do modelo correto, ou seja, observa-se se está representando o sistema real com a precisão desejada. A aceitabilidade do modelo passa por esse processo iterativo de comparação, onde são observadas as discrepâncias no comportamento dos resultados.

Normalmente, a validação pode ser alcançada através da calibração, que são ajustes realizados durante todo o processo de desenvolvimento com o intuito de aproximar os resultados dos valores esperados. É importante observar que sempre refere-se ao modelo e não ao simulador, isto porque estes conceitos são abrangentes e podem ser usados inclusive para os modelos analíticos e suas representações matemáticas.

Como o objetivo não é discutir em profundidade cada técnica, serão relacionadas algumas de forma a obter-se o conhecimento necessário para a próxima seção. Deve-se notar que não é necessário usar todas as técnicas, elas são sugestões empregadas na literatura, e qualquer combinação delas pode ser aceitável. Primeiro, são listadas as técnicas para verificação [BAN 95]:

- Ter a representação computacional analisada por outro desenvolvedor ou analista.
- Fazer um diagrama de fluxo que inclua toda lógica possível do sistema, mostrando cada ação realizada quando ocorre um evento.
- O modelo deve mostrar os valores de entrada ao final do processo, a fim de garantir que não ocorreram mudanças inadvertidas.

- Manter o máximo possível de comentários na representação computadorizada, de forma a garantir a precisa definição de cada variável e trecho de código.
- Se o modelo tiver uma representação animada, verificar visualmente se ela imita o sistema real.
- As interfaces gráficas, quando presentes, são uma boa forma de documentação implícita, pois simplificam a compreensão do modelo; ajudando na verificação.
- Ter um sistema de controle interativo (IRC) ou um depurador (*debugger*) implementado no simulador. Esse tipo de ferramenta de apoio pode ajudar de diversas formas ou níveis de acordo com a complexidade implementada. Assim, podem-se monitorar os passos do processo, verificar mais atentamente somente um bloco ou uma variável; permitir dar pausas na execução e observar alguns valores selecionados; ou ainda num nível mais avançado, modificar esses valores durante as pausas para observação. Deve-se notar que estes sistemas acarretam em mais tempo de processamento e por consequência, tempo de simulação.
- Fazer um exame minucioso dos resultados do modelo a partir de uma razoável variedade de conjuntos de valores de entrada. É importante o modelo ter um grande número de resultados estatísticos.

Mesmo os mais experientes analistas cometem enganos na construção do modelo, por isso, não se pode deixar de verificar a programação mesmo que se tenha passado para a fase de validação. É importante estar sempre atento a aspectos estranhos ou discrepantes nos resultados das simulações, pois através deles pode-se detectar um ponto de falha na implementação. Para continuar a explanação das técnicas, a próxima lista refere-se a validação do modelo [BAN 95]:

- A validação das suposições estruturais deve ocorrer para dar credibilidade ao modelo. Por suposições estruturais entende-se as abstrações e simplificações da realidade, que devem ser verificadas através de observações durante um período ou confirmadas por usuários conhecedores do sistema real.
- As suposições de dados devem também ser validadas, veja o capítulo de modelos de tráfego [cap. 4]. Muitas vezes, os parâmetros de entrada de um sistema real não são bem conhecidos e devem-se realizar estudos a fim de se obter uma análise do padrão das variáveis. Esta análise pode ser dividida em três passos: a identificação da apropriada distribuição de probabilidade, a estimação dos parâmetros que definem esta distribuição proposta, e, por fim, a validação deste modelo estatístico através de testes ou gráficos.
- Podem-se obter dados de saídas sobre o sistema real e compará-los diretamente com os resultados do modelo. Este é o teste mais evidente que se pode idealizar, porém, muitas vezes não é possível realizá-lo, como no caso de sistemas propostos.
- Realizar testes estatísticos com os resultados do modelo e compará-los aos do sistema real. A diferença deste método para o anterior é a manipulação dos dados. Sempre lembrando que existem os problemas referentes a aquisição dos dados no simulador conforme visto na seção [5.2]. Em todo processo estocástico deve-se ter cuidados para manter a fidelidade [BIH 99].

- A análise de sensibilidade do modelo, que é descrita como a avaliação do comportamento do modelo quando muda-se uma variável de entrada específica. A comparação é baseada na experiência ou observação do sistema real, apesar de nem sempre se ter a variação de um único parâmetro isolado. Deve-se notar também que modelos grandes podem levar a testes exaustivos, então, deve-se escolher as variáveis de entrada mais críticas ao processo.
- Verificar a correta "predição do passado", e desta forma, pode-se considerar que o modelo seja capaz de prever o futuro, e assim supor sua validade. É uma técnica semelhante às anteriores onde comparam-se as saídas do modelo ao sistema real, porém as entradas usadas pelo modelo não são geradas de forma randômica; seguindo qualquer distribuição estatística, elas devem ser obtidas de dados reais. É necessário neste caso, não somente um histórico das saídas, mas também das entradas do sistema real.
- Testes de condição extrema, em que o modelo é submetido a uma combinação de fatores atípicos a fim de verificar se não apresenta um comportamento estranho. Pode-se, desta forma, estipular os limites de utilização do modelo.
- A estimulação através de valores fixos ou de cálculo fácil, que podem ser usados a fim de se obter certos resultados esperados, independente da ocorrência em sistemas reais. Pode-se desta forma compreender o modelo como um bloco do qual espera-se um determinado resultado.
- A calibração, que é uma técnica de ajuste e não de validação do modelo, porém, quando se passa por esta fase adquire-se maior credibilidade no modelo. De qualquer forma, é usual manter no mínimo dois conjuntos de dados de entrada, um para a calibração e outro utilizado ao final desta para validar o modelo.
- A validação de eventos, onde não são importantes os dados de saída, e sim que os eventos ocorridos no modelo de simulação, quando comparados aos eventos do sistema real, tenham uma mesma seqüência ou comportamento.
- Submeter o modelo ao julgamento de pessoas especialistas em algum aspecto do sistema real, ou em todo ele, a fim de observar o comportamento da simulação e dizer se é razoavelmente semelhante.
- Um fato que aumenta a credibilidade do modelo e conseqüentemente sua validade, é o acompanhamento ou participação de usuários do sistema real e especialistas no assunto durante a construção do simulador.
- A realização de um teste de Turing, ou seja, obter um conjunto de saídas do sistema real e outro conjunto de saídas do modelo para submetê-las aos especialistas. O objetivo é que eles identifiquem quais são as saídas do modelo e as do sistema real, e se o especialista não puder distinguir os conjuntos de dados pode-se assumir que não existem evidências de que o modelo é inadequado.
- Pode-se fazer comparações com outros modelos, e desta forma analisar as saídas do novo modelo em estudo a partir de vários resultados que se assemelhem a valores já validados anteriormente nos outros modelos conhecidos. Essa comparação é necessária em casos de sistemas propostos, podendo ser realizada com outros

sistemas que tenham comportamento similar, porém, não existem garantias de uma completa validação em sistemas não existentes, apesar de incrementar a confiança no modelo.

- A transferência de confiança é empregada quando um novo modelo é criado a partir de um modelo antigo. Neste caso, deve-se avaliar se as modificações são consideradas "menores", ou seja, se existe suficiente similaridade entre os modelos e não se realizou nenhuma mudança estrutural ou lógica que comprometa a validação anterior. Deve-se complementar o processo através da validação dos novos subsistemas criados.
- No caso de não existirem dados reais para comparar, tais como em sistemas na fase de planejamento, pode-se obter uma validação parcial utilizando as técnicas aqui descritas em alguns subsistemas do modelo.

É importante notar que a validação pode ser muito mais complicada de se garantir do que a verificação. Além disso, muitas das técnicas apresentadas têm um certo grau de subjetividade e dependem da experiência ou enfoque do analista, desta forma, é importante ter os objetivos bem definidos durante todo o processo. É bom lembrar neste caso, que a validação é feita de acordo com o propósito original do modelo, e a utilização para propósitos diferentes devem ser revalidados.

Algumas das técnicas são bem semelhantes entre si, isto porque quase todas se baseiam nos dados de saída, porém, a visão do problema que se pretende localizar é diferente. Um outro ponto de difícil decisão é quando julgar que o modelo é suficientemente validado, e para isso, normalmente deve-se predefinir um nível aceitável de discrepância e restrições. Desta forma, cada novo conjunto de entradas deve gerar resultados dentro do esperado, senão deve-se levar a novas calibrações, mesmo de modelos já considerados válidos.

## ***6.2 A Verificação do Simulador***

Conforme comentamos na seção anterior, a verificação é feita com o intuito de garantir que não existam erros de implementação. São na sua maioria técnicas usuais para grande parte dos programadores, apesar de normalmente não serem empregadas por excesso de confiança ou displicência. Nem todas são necessárias para se considerar o sistema verificado, mas quanto mais técnicas forem utilizadas mais credibilidade pode-se ter. De outra forma, tais erros de implementação podem causar sérios problemas e modificar o resultado de forma a tornar impraticável o uso do simulador para estudo ou previsão do comportamento da rede e seus protocolos.

Quando optou-se por utilizar o simulador NIST, algumas facilidades já faziam parte da programação original e não houve a necessidade de desenvolvê-las. Algumas dessas características do simulador NIST podem ser consideradas como ferramentas de auxílio a verificação, que são a interface gráfica e o arquivo de saída denominado de SNAP. Na interface gráfica pode-se ter idéia precisa de como estão todas as conexões da rede através do diagrama ali apresentado, apesar da parte de animação das rotas não ser muito boa e normalmente pouco útil. O arquivo de SNAP é semelhante ao arquivo de configuração, com a diferença que apresenta os valores dos parâmetros do momento em que foi salvo em disco, e pode ser criado através de um botão durante qualquer

pausa no uso da interface gráfica ou automaticamente ao final da simulação. No anexo 3 são apresentados exemplos dessas ferramentas, que foram empregadas durante as simulações.

Toda programação do novo módulo baseia-se nos diagramas SDL das normas do ITU-T, e procuramos manter inclusive os mesmos nomes identificadores para os parâmetros e variáveis, de forma a tornar simples qualquer comparação. Além disso, tenta-se manter o máximo de comentários no código fonte para tornar a lógica mais compreensível. Mas, o que realmente auxilia a visão da seqüência de eventos e da mudança dos valores das variáveis é um depurador simples que foi criado justamente com a finalidade de localizar erros de programação durante o desenvolvimento. A partir da opção de compilação "LOGAAL2", pode-se obter uma saída conforme a vista na seção 5.5.4.

Mesmo que o novo módulo não contenha erros, poderiam ocorrer problemas nas alterações efetuadas no simulador para que suportasse as novas funções e testes. Para tentar comprovar a integridade da programação original, foi feita a verificação através da comparação com simulações sem a presença do novo componente AAL2, realizadas entre o arquivo fonte inalterado e o novo arquivo fonte. Por se considerar que os exemplos de arquivos de configuração presentes no pacote do simulador NIST sejam o conjunto mais completo das combinações de rede por ele utilizada, e também por já estarem devidamente testados, foram empregados como dados de entrada nas simulações.

TABELA 6.4 - Teste para verificar a integridade do simulador original

Arquivo de entrada	números de sementes	número de simulações	tempo de simulação	tamanho da amostra		diferenças entre as saídas
				linhas	Mbytes	
configabr	3	3	40 s	38.517	5,6	0
configatcp	3	3	40 s	47.351	6,8	0
configcblan	3	3	4 s	12.127	1,7	0
configcbwan	3	3	0,68 s	16.781	2,4	0
configerica	3	3	4 s	56.092	11	0
configmixerica phantom	3	3	4 s	56.977	8,2	0
confignist	3	3	4 s	69.394	10	0
configphantom k1	3	3	4 s	43.204	6,2	0
configphantom k5	3	3	4 s	47.250	6,8	0
configqfclan	3	3	4 s	714.032	28,9	0
configqfcwan	3	3	0,63 s	14	0,0005	0

Como pode ser observado pela coluna diferenças, os arquivos de saída são idênticos. Tanto na simulação realizada com o fonte original do NIST, como na que utiliza o fonte modificado para suportar a camada AAL, os dados, o tempo de simulação e inclusive as dificuldades foram as mesmas. Neste teste, alguns arquivos de configuração são extremamente grandes e complexos, exigindo grande quantidade de

memória e CPU, o que dificultou longas replicações. Apesar disso, sempre que se usa a mesma semente nos dois fontes obtém-se um arquivo de saída exatamente igual.

Existe ainda a integração do novo módulo com os demais módulos do simulador, mas isso é verificado pelo perfeito funcionamento observado através do depurador e simulações do protocolo AAL2 que serão vistos mais adiante.

### **6.3 *Artigos sobre o AAL2***

Como foi dito anteriormente, o protocolo AAL 2 foi padronizado recentemente, sendo desta forma um campo ainda não totalmente avaliado. Apesar da pouca informação disponível, tentaremos, aqui, descrever algumas pesquisas e testes publicados para conseguir uma base sólida de estudo. Pretendemos, através deste conhecimento, comparar com as simulações que seguem nas seções posteriores.

#### **6.3.1 *Comentários sobre a Telefonia Celular***

Os engenheiros da comunicação celular é que buscaram, neste padrão, uma solução para seus tipos de redes, onde normalmente é usada voz comprimida em tempo real. Eles tiveram dois objetivos : baixo atraso de empacotamento e alta eficiência de largura de banda. Esta influência fez com que esta padronização do AAL2, que havia ficado no esquecimento desde que criaram-se os outros padrões, finalmente pudesse ressurgir e com grande força na direção da integração de redes de voz e dados.

Como a telefonia celular tem apresentado um crescimento acelerado nestes últimos anos, pode-se esperar uma evolução rápida do AAL2. Já pode-se inclusive contar com os primeiros estudos nesta área de comunicação móvel, seja usando as técnicas de rádio difusão como o W-CDMA ou mesmo com aquelas popularizadas como o GSM. Em [NOG 98], Juan Nogueira da Ericsson Nippon conclui que o AAL 2 proporciona um bom mecanismo para ajuste fino do atraso de empacotamento e utilização eficiente da largura de banda para serviços de baixa taxa de bits com aplicações VBR. Ele verifica que um equipamento GSM gera 260 bits a cada 20 ms; podendo ser usado diretamente na subcamada CPS sem necessidade de tratamento extra.

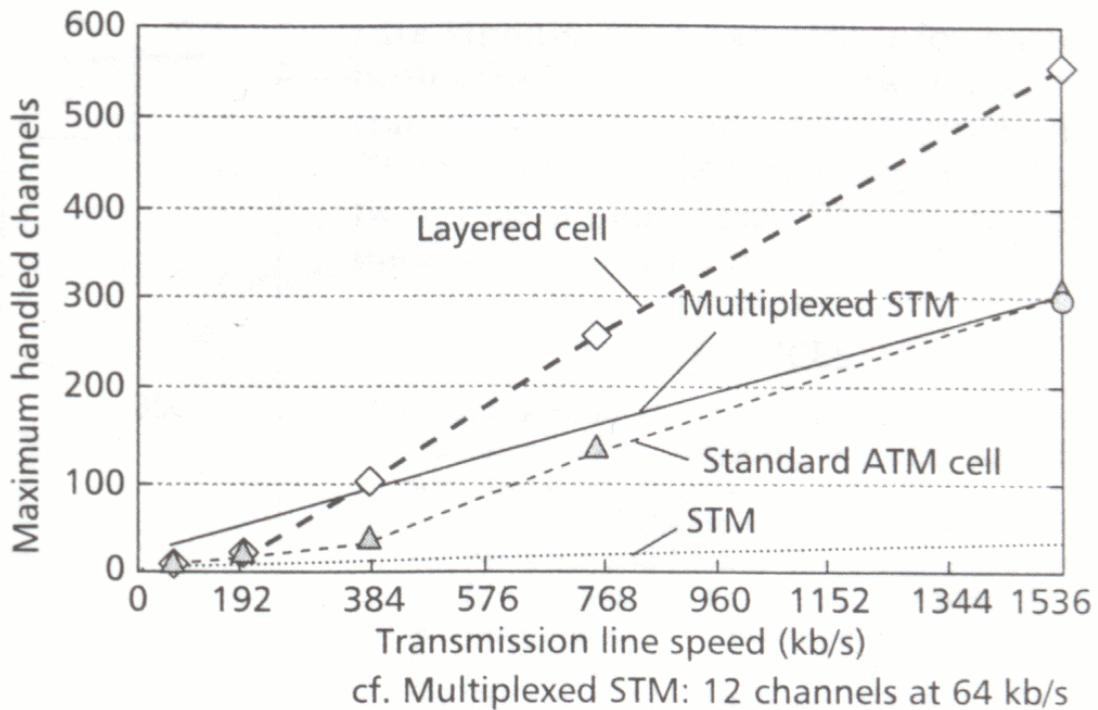


FIGURA 6.1 - Comparação da eficiência de transmissão

Na avaliação feita por simulação em [NAK 98], onde o AAL 2 é usado para transmissão das estações receptoras para a estação de controle de redes de celulares, ficou demonstrado que ele é mais eficiente do que os métodos tradicionais. Foi observado que, quanto maior a velocidade da linha de transmissão, cresce muito o número de canais que podem ser multiplexados. Para realizar esta simulação, foi usado o modelo de tráfego da tabela 6.1 abaixo, e obteve-se o gráfico da figura 6.1.

TABELA 6.1 - Modelo de tráfego para a simulação

Parâmetros	Período de voz	Período de silêncio
Intervalo do quadro (ms)	40	
Janela de chegadas (ms)	4	
Taxa de bits (Kb/s)	4	0
Tamanho do pacote (octetos)	20	0
Probabilidade de silêncio (%)	50	50
Tempo expirado de t1 (ms)	1	
Tempo expirado de t2 (ms)	4	
Taxa de células perdidas	$10^{-3}$	

Também fica evidente que para uma melhor economia das baterias de terminais móveis e uma maior eficiência na utilização da banda de frequência de rádio, a transmissão operada por voz (VOX) é cada vez mais empregada. No VOX só ocorre transmissão enquanto o usuário fala, quando existe silêncio o canal fica livre. Neste caso, pode-se verificar a vantagem do ATM sobre o STM [NAK 98], pois no segundo é necessário reservar um "time slot" para cada canal nas estações receptoras para comunicação com a estação de controle.

### 6.3.2 A Eficiência do Protocolo AAL2

Os dados que serão apresentados nesta subseção foram obtidos dos artigos publicados pela General DataComm [MCL 97a]. Devido ao fato do protocolo ter tamanho variável de pacote (de 1 a 45 octetos), ele permite um controle fácil do atraso de empacotamento. Desta forma, possibilita que a aplicação escolha o tamanho de acordo com o atraso desejado, ou seja, quanto menor o tamanho, menor será o atraso. No entanto, o atraso no preenchimento das células pode influenciar nas características de atraso da voz, desta forma, deve-se escolher bem o tempo de montagem e segmentação, conforme será visto na seção [6.3.4]. Diferentes circuitos de voz podem ter diferentes requisitos mínimos de atraso, sendo importante saber balancear o atraso com a eficiência desejada.

TABELA 6.2 - Eficiência do protocolo AAL2

	Cabeçalho	Carga	Eficiência
32K ADPCM (2 ms)	3 bytes	8 bytes	73 %
32K ADPCM (4 ms)	3 bytes	16 bytes	84 %
32K ADPCM (8 ms)	3 bytes	32 bytes	91 %
PCM (2 ms)	3 bytes	16 bytes	84 %
PCM (4 ms)	3 bytes	32 bytes	91 %
PCM (8 ms)	3 bytes	64 bytes	96 %
Default LI	3 bytes	45 bytes	94 %
Máximo LI	3 bytes	64 bytes	96 %

A primeira tabela [6.1] mostra algumas relações entre o atraso no empacotamento e o número de octetos que podem ser carregados antes do envio do pacote para a rede ATM. Pode-se notar que a eficiência puramente medida a partir da carga do pacote e do "overhead" do AAL2 parecem piorar quando o tempo de atraso é menor, mas esta visão é limitada e não vislumbra os benefícios da multiplexação.

TABELA 6.3 - Largura de Banda requerida usando AAL2 para 6 canais de voz

Número de canais	Taxa do Canal	AAL	Atraso no empacotamento (ms)	Supressão de silêncio	Largura de Banda (Kbps)
6	64K	2	6	não	495
6	32K	2	4	não	318
6	32K	2	6	não	283
6	32K	2	8	não	265
6	64K	2	6	sim	198
6	32K	2	4	sim	128
6	32K	2	6	sim	113
6	32K	2	8	sim	106

O ganho na largura de banda é realizado pela multiplexação de várias conexões ativas de AAL2 sobre o mesmo VCC. A tabela 6.3 mostra o uso de seis canais de voz dando o fator de compressão (nenhum ou 32K ADPCM), o atraso de codificação, o atraso de empacotamento, o suporte para supressão de silêncio (assumindo 50% de silêncio) e na última coluna a largura de banda requerida. Agora, pode-se verificar a real performance.

Ficaram claros os benefícios do AAL2 sobre o tráfego de voz. Porém, devido a dificuldades, tais como calcular o ganho estatístico na compressão do canal de voz (por exemplo, ADPCM e supressão de silêncio) e possíveis complicações de empacotamento, torna-se complicado obter uma fórmula teórica para calcular a largura de banda necessária para a rede ATM.

TABELA 6.4 - Número de canais de voz em um tronco DS1 usando AAL2

Atraso no empacotamento (ms)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
32K ADPCM	22	28	31	33	34	34	35	35	36	36	36	37	37	37	37	37
64K PCM	14	16	17	17	18	18	18	18	18	18	19	19	19	19	19	19
32K ADPCM *	55	70	77	82	85	85	87	87	90	90	90	92	92	92	92	92
64K PCM *	35	40	42	42	45	45	45	45	45	45	47	47	47	47	47	47

Outra maneira de verificar a eficiência do protocolo AAL2 é dada pela identificação de quantos canais de voz podem ser colocados em um tronco ATM com largura de banda fixa. Na tabela 6.4 verifica-se o acréscimo de canais em um tronco T1 ATM quando aumenta-se o atraso no preenchimento do pacote, e também como já era de se esperar, quando utiliza-se compressão nos canais de voz. As duas últimas linhas (\*) representam o significativo ganho quando assume-se que cada circuito de voz tenha 50% de silêncio e 20% de canais em estado de espera (idle).

### 6.3.3 Comparação entre o AAL1 e o AAL2

Este artigo também foi publicado pela General DataComm [MCL 97b], e propõe-se a comparar os protocolos AAL2 e AAL1 supondo que sejam utilizados para emular um circuito convencional de telefonia. Hoje, para o tráfego de voz, existem dois tipos de emulação usando o protocolo AAL1, sobre os quais podem-se comentar alguns detalhes :

#### A) Emulação de Circuito não Estruturado (Circuit Emulation)

- Esta aproximação baseia-se no uso de uma conexão ATM CBR AAL1 para carregar um circuito E1 ou T1 inteiro entre dois pontos da rede. O objetivo deste método é transportar transparentemente dados numa conexão entre Multiplexadores por Divisão de Tempo (TDM) ou PBXs Digitais. Este método é bastante simples, porém, inflexível e ineficiente, devido ao fato de não fazer uso do ganho estatístico, que é o fator chave da migração para o ATM.
- Tipicamente, esta emulação é configurada como um circuito virtual permanente, e como tal, está sempre ativo. Outras aplicações não podem fazer uso dos recursos de rede, mesmo que existam vários canais de voz que fiquem em estado de espera (idle). A exemplo dos circuitos de telefonia, a largura de banda fica sempre alocada, porém com uma desvantagem adicional, pois para suportar uma emulação de circuito, a rede ATM necessita significativamente mais banda. Isto deve-se ao "overhead" das células (12%), ou seja, um circuito DS1 com 1,536 Mbps irá requerer 1,73 Mbps de largura de banda no ATM. O "overhead" é a medida do quanto de cabeçalho está sendo transportado junto da informação e, logicamente,

utiliza parte da largura de banda, onde o valor normal neste caso é 10% para camada ATM (5 dos 53 octetos) adicionados aos 2% do AAL1.

### **B) Emulação de Circuito Estruturado (AAL1 CES/CAS)**

- Oposta a anterior, esta aproximação usa as conexões AAL1 ATM para estabelecer  $N \times 64$  Kbps circuitos. Infelizmente, desta forma perde-se a simplicidade e a transparência, pois deve-se reconhecer cada canal individual a fim de mapeá-los para os troncos PBXs. A vantagem é que se pode estabelecer mais circuitos granulares de  $N \times 64$  Kbps para adicionar largura de banda, quando necessário, ou reduzir se estiverem sem uso. Este reconhecimento de cada "timeslot" leva a rede a carregar a Sinalização Associada ao Canal (CAS) através de um canal base.
- Existem riscos devido ao contraste entre eficiência versus características de atraso. Isto ocorre, porque os serviços de voz exigem um mínimo de atraso, caso contrário podem surgir problemas como o eco. A emulação de circuito estruturado deve pegar de cada amostragem somente a informação do canal desejado, sendo assim necessário aguardar algum tempo para preencher toda a célula. No caso mais eficiente, as células são totalmente preenchidas (47 octetos), mas é necessária uma taxa de 47 amostras no tempo. Para diminuir o atraso, muitas vezes é necessário preencher parcialmente as células. Enquanto um E1 tradicional usa 384 Kbps para 6 canais, o preenchimento total usa 443 Kbps (devido ao overhead), e para o caso do preenchimento parcial pode usar 866 Kbps a uma taxa de amostragem de 24 vezes (23 octetos seguem vazios).

Como pode ser visto, são usadas excessivas larguras de banda nestas emulações, além da falta de flexibilidade e de controle dos atrasos. Mesmo a emulação de circuito estruturada, se comparada ao E1 ou T1 tradicional ocupa cerca de 15% a mais de largura de banda. Neste caso, pode-se concluir que as propriedades do AAL2 podem ajudar nas futuras integrações entre redes de telefonia e dados, tendo em vista seu eficiente controle da largura de banda através do tráfego VBR e o uso de múltiplos canais de voz sobre uma única conexão ATM. Além disso, apresenta características que permitem o uso de voz comprimida, detecção e supressão de silêncio, ou detecção de canal de voz inativo, fazendo com que ele seja tão vantajoso.

Algumas comparações diretas podem ser feitas como por exemplo, o AAL1 apresenta 98% de eficiência (apenas 2% de overhead), enquanto o AAL2 pode variar entre 25% a 96% dependendo do tamanho do pacote. Mas na seção anterior, mostrou-se que para os mesmos 6 canais de voz no AAL2, usam-se 495 Kbps sem a utilização de outra técnica auxiliar, o que se aproxima do melhor caso obtido com AAL1, 443 Kbps. Quando utilizando o AAL2 no seu melhor desempenho, reduz-se para 113Kbps, o que seria um ganho de 74% menos largura de banda do que o AAL1 CES/CAS, e cerca de 70% menos que o tradicional E1.

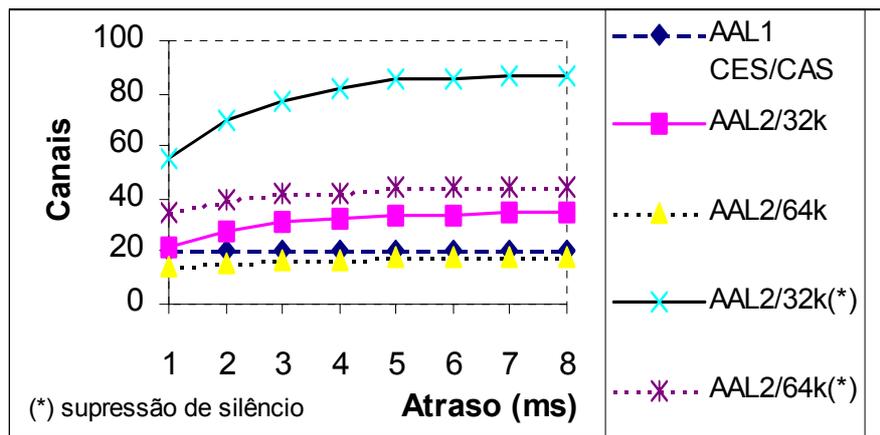


FIGURA 6.2 - Número de canais em um tronco DS1 variando o atraso de empacotamento

O gráfico da figura 6.2 mostra novamente a comparação entre os protocolos, usando os dados expostos na subseção anterior onde verifica-se o número de canais em um tronco T1 ATM, sendo que o AAL1 consegue transportar um máximo de 20 canais (com 47 amostras). Pode-se verificar o evidente salto de quatro vezes mais canais quando usando supressão de silêncio.

Mas isto não significa que não exista demanda para o protocolo AAL1, pelo contrário, estamos falando aqui especificamente de transmissão de voz e por isso, sobre tráfegos que têm certas características particulares. Existem diversas outras aplicações onde somente o AAL1 pode ser usado, por exemplo, a transmissão de vídeo. Mesmo no caso do tráfego de voz, o AAL1 ainda é melhor empregado em casos onde necessita-se um "Digital Cross-Connect System" sobre a rede ATM [MCL 97b]. Neste caso é necessário utilizar um diferente circuito virtual para conectar a cada diferente equipamento distribuído na rede, desta forma, o AAL2 não é indicado, pois utiliza um único circuito virtual para todos os usuários.

#### 6.3.4 O Relógio de Uso Combinado (*Timer\_CU*)

Esta seção é baseada nas conclusões do artigo "Packing Density of Voice Trunking Using AAL2" escrito por Raj Jain do ATM Fórum [JAI 99], que verifica a influência do parâmetro *Timer\_CU*, pois na definição do novo protocolo a norma não especifica qual deveria ser o seu valor. Este artigo chega a conclusão de que o parâmetro tem significativo impacto na eficiência das conexões, e verifica que o apropriado valor é dependente do número de fontes de voz e do atraso desejado.

De acordo com a Recomendação ITU-T G.114 [ITU 96a], um atraso fim a fim de 0 a 150 ms é aceitável pela maioria das aplicações. Atrasos superiores a 400 ms são inaceitáveis, mas valores entre 150 e 400 ms ficam a encargo dos administradores das redes assumirem a redução de qualidade nas aplicações dos usuários.

Como já foi dito antes, o problema do atraso ocorre durante o empacotamento do sinal de voz na célula, e fica mais grave quando utilizam-se métodos de compressão. Por exemplo, para preencher os 48 octetos de carga da célula, um "codec" seguindo a norma G.711 a 64 Kbps necessita 6 ms, mas se utilizar a norma G.723.1 a 5,3 Kbps (voz

comprimida) necessitaria de 72 ms. Estes tempos não incluem as outras perdas inerentes da rede, tais como o atraso de propagação, tempo de atendimento em filas, etc.

O princípio de multiplexação do AAL2 garante o preenchimento mais rápido da célula com pacotes de outros usuários e, desta forma, reduz o problema do atraso. Porém, este método tem o problema de não poder aguardar indefinidamente por outros pacotes, pois prolongaria muito o tempo de atraso. É por isso que a norma propôs o parâmetro `Timer_CU`, e o artigo descrito fornece um algoritmo para calcular um valor de referência para este parâmetro baseado na densidade do pacote.

É utilizada uma simulação para comprovar os valores obtidos para o cálculo do `Timer_CU`. Baseado em estudos sobre fontes de voz e considerando o G.723.1 como o padrão mais largamente usado, foi definido o modelo de tráfego da tabela 6.5. A partir deste modelo foi calculada a expressão da densidade e realizada a simulação. Por densidade compreende-se a taxa de utilização do pacote, ou seja, o número de octetos de dados usados em relação ao tamanho do pacote.

TABELA 6.5 - Modelo de tráfego para a simulação do `Timer_CU`

Parâmetros	Período de voz	Período de silêncio
Tamanho do pacote (octetos)	20	0
Comprimento médio (ms)	352	650
Intervalo entre pacotes (ms)	30	0

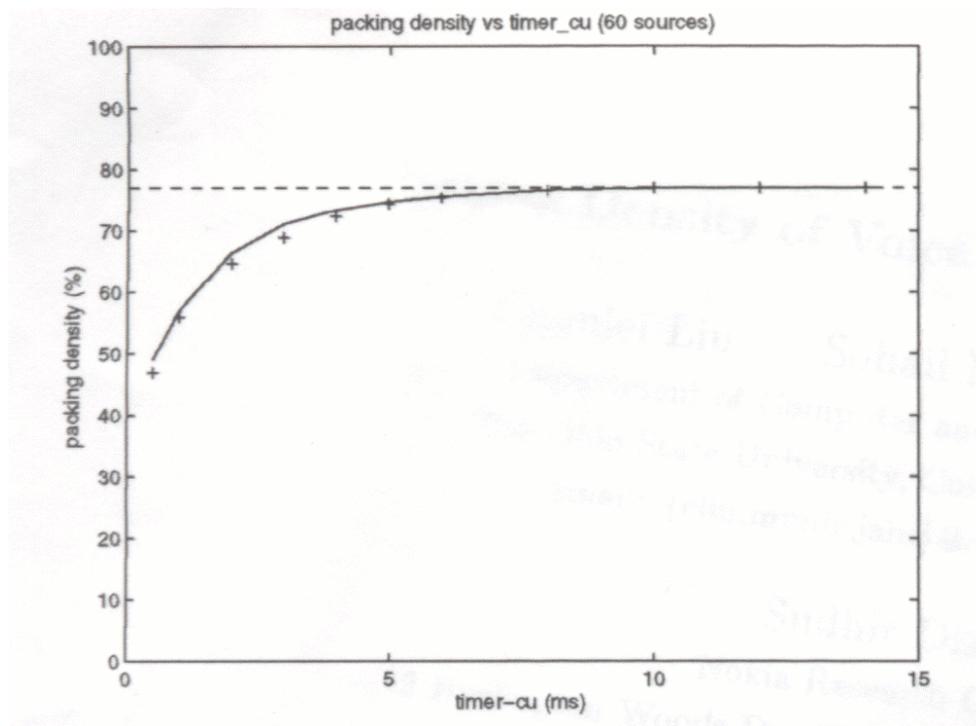


FIGURA 6.3 - Crescimento da densidade do pacote com o aumento do `Timer_CU`

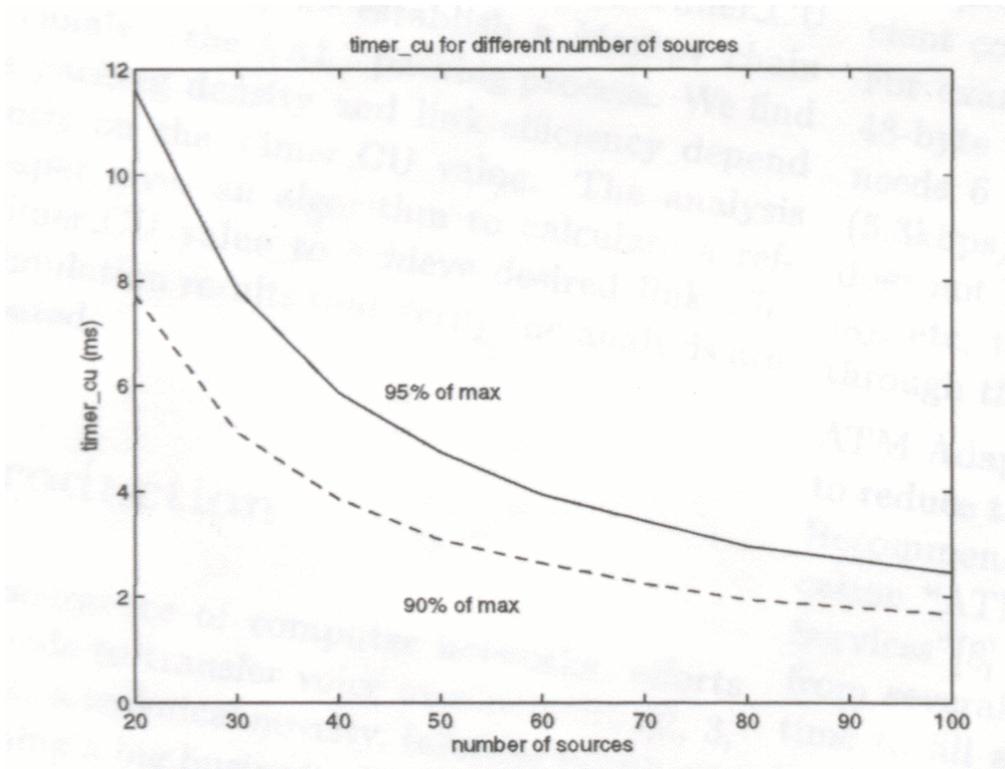


FIGURA 6.4 - Valores de Timer\_CU para alcançar 90% e 95% da máx. densidade do pacote

O gráfico acima é o resultado mais importante para a continuidade deste trabalho, pois define os valores do Timer\_CU em relação à utilização da conexão, tendo grande utilidade para garantir a escolha de um bom valor, e assim uma boa eficiência e qualidade. Observe que os valores muito pequenos de Timer\_CU podem fazer com que aumente o número de células parcialmente vazias; reduzindo a eficiência da conexão, e valores muito grandes podem causar um prolongamento do atraso e prejudicar a qualidade do serviço. É citado neste artigo um outro estudo que avalia o impacto do valor do Timer\_CU sobre a variação do atraso do pacote (PDV), e nele conclui-se que o parâmetro não tem peso relevante no valor total deste atraso.

## 6.4 A Comparação para Validação

A parte mais complexa depois do trabalho de criar o novo componente que deveria simular o protocolo AAL2, é validá-lo. A partir do ambiente gráfico que o simulador NIST dispõe, foram criados alguns diagramas de redes com a finalidade de testar o funcionamento correto do protocolo de acordo com a norma ITU-T. Mas, posteriormente foram realizadas simulações com o modelo de tráfego de voz escolhido, a fim de obter informações do comportamento destas redes e compará-los aos resultados descritos nos artigos referidos nas seções anteriores.

Dentre as técnicas de validação descritas na seção 6.1, a maioria não pode ser aplicada a este trabalho, por vários motivos diferentes, seja pelo tempo curto para realizar certas análises, seja pela dificuldade de encontrar especialistas no assunto ou ainda por não existir uma rede com as características necessárias disponível para obter dados reais. Assim, optou-se por comparar os resultados das simulações com resultados

de outros autores, sempre se preocupando com os problemas relacionados à obtenção dos valores estatísticos necessários que serão vistos na seção 5.2 .

Quanto a validação das suposições dos dados usados para simular, pode-se dizer que sendo o protocolo AAL2 voltado para tráfego de voz e devido ao exposto no capítulo 4 que o modelo ON/OFF exponencial é bastante adequado para representar este tipo de tráfego, então conclui-se que as fontes geradoras dos dados de entrada devem se aproximar bem da realidade reduzindo as chances de haver resultados incorretos.

Outro ponto importante é que o NIST conta com uma experiência de 13 anos no desenvolvimento de simuladores, o que é a garantia de uma ferramenta já validada e verificada, portanto, que cumpra os parâmetros de padronização emitidos pelo próprio instituto. Pode-se então neste caso, concluir que exista a transferência de confiança do simulador original para o modificado, tendo em vista que todas as mudanças realizadas sempre tiveram a preocupação de não alterar as funções essenciais do simulador, conforme exposto em toda a seção 5.5 . Porém, resta validar o novo componente para verificar se representa o real funcionamento do protocolo AAL2.

Para validação de eventos do novo módulo pode-se usar o depurador criado para o AAL2, que foi discutido na seção 5.5.4. Através dele pode-se verificar a seqüência de eventos e compará-la com o esperado no diagrama SDL da norma ITU-T, e após diversos testes, pode-se dizer que o módulo representa perfeitamente o protocolo AAL 2.

TABELA 6.6 - Tipos de testes executados para validação

Arquivo de entrada	timer_CU	fonte de tráfego	número de fontes	taxa de transmissão das fontes	tempo ativo	tempo inativo	tamanho do pacote
teste1	10 ms	vbr	2	13 kbps	100 ms	100 ms	32
teste2	10 ms	vbr	4	13 kbps	100 ms	100 ms	15 e 32
teste2b	10 ms	on-off	4	32 kbps	100 ms	100 ms	15 e 64
teste2c*	10 ms	on-off	4	32 kbps	100 ms	100 ms	15 e 64
teste3	10 ms	vbr	6	13 kbps	100 ms	100 ms	15 e 64
teste3b	10 ms	on-off	6	64 kbps	100 ms	100 ms	15 e 64
teste4	10 ms	vbr	60	32 kbps	100 ms	100 ms	32
teste4a	1 a 12ms	on-off	variável	32 kbps	352 ms	352 ms	20
teste4b	1 a 12ms	on-off	variável	64 kbps	352 ms	352 ms	20
teste4c	1 a 12ms	on-off	variável	32 kbps	352 ms	352 ms	20
teste4d	1 a 12ms	on-off	variável	64 kbps	352 ms	352 ms	20
teste5	4 ms	on-off	60	4 kbps	40 ms	40 ms	32
teste5a	4 ms	on-off	100	4 kbps	40 ms	40 ms	20
teste5b	4 ms	on-off	50	4 kbps	352 ms	352 ms	20
teste5c	4 ms	on-off	100	4 kbps	352 ms	352 ms	20
teste5d	4 ms	on-off	200	4 kbps	352 ms	352 ms	20
teste5e	4 ms	on-off	300	4 kbps	352 ms	352 ms	20
teste5f	4 ms	on-off	400	4 kbps	352 ms	352 ms	20
teste5g	4 ms	on-off	500	4 kbps	352 ms	352 ms	20
teste6	0,5 a 12ms	on-off	60	5,3 kbps	352 ms	650 ms	20

\* taxa de transmissão do componente AAL2 é reduzida de 10 Mbps para 74 kbps.

As comparações dos artigos com o resultado das simulações podem ser resumidas nos gráficos e tabelas desta seção, e o diagrama da rede de teste e seus arquivos de configuração podem ser vistos em alguns exemplos no anexo 3. Os testes 1, 2 e 3, foram realizados para verificar o protocolo através do debug.

TABELA 6.7 - Uso do arquivo de configuração “teste5” para validação do simulador

Arquivo de entrada	número de replicações	tempo de simulação (P200Mhz)	tamanho da amostra (40 s)		média	desvio padrão	Intervalo de confiança (95%)
			linhas	Mbytes			
teste5b	10	~ 1:20 h	~ 40	~ 0.9	159.51	1.91	+ -3.46
teste5c	29	~ 40 min	~ 80	~ 1.7	308.57	1.70	+ -2.89
teste5d	24	~ 40 min	~ 140	~ 3.0	619.43	3.40	+ -5.82
teste5e	24	~ 40 min	~ 200	~ 4.3	928.98	5.50	+ -9.40
teste5f	24	~ 40 min	~ 290	~ 5.7	1236.11	5.40	+ -9.23
teste5g	26	~ 40 min	~ 350	~ 7.0	1547.38	5.84	+ -9.98

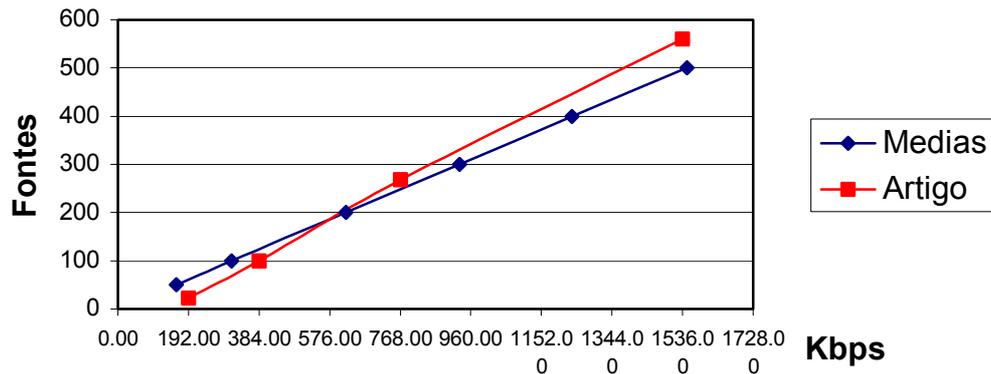


FIGURA 6.5 - Comparação do número de fontes obtidas pelo simulador com a figura 6.1

Este gráfico acima, obtido pelos arquivos do tipo "teste5", serve de comparação com o artigo de "Aplicando ATM para Infraestrutura de Redes Móveis" [NAK 98]. Pode-se observar que as curvas são bem próximas, porém, a medida que cresce o número de fontes de tráfego a diferença aumenta. Isto pode ser devido ao fato da simulação do artigo usar o total de fontes divididos em quatro grupos e cada qual ativo por 10 ms em cada 40 ms, o que não é possível realizar com as fontes exponenciais ON-OFF, pois seu período é aleatório. Assim, optou-se por manter intervalos iguais (50%) com o tempo de 352 ms proposto em outros estudos.

Existe também, na simulação do artigo, um buffer de entrada onde é armazenado todo pacote recebido pelas fontes de tráfego. Isto faz com que estes pacotes não sejam descartados, caso a célula esteja completa e ainda não exista permissão de envio pela camada de gerenciamento. No caso da modificação realizada no NIST, isto não ocorre, pois foi seguido a risca a norma I.363.2, que não especifica que atitude tomar nestes momentos e somente declara ficar para estudos futuros do ITU. Logicamente, esta perda será tanto maior quanto for o número de canais em uso, o que pode ser uma das diferenças presente no gráfico.

TABELA 6.8 - Uso do arquivo de configuração “teste6” para validação do simulador

TimerCU (ms)	número de replicações	tempo de simulação (P200Mhz)	tamanho da amostra		média (%)	desvio padrão	Intervalo de confiança (95%)
			linhas	Kbytes			
0.5	14	~ 1h	~ 21050	~ 396	49.61	0.0623	+ - 0.1096
1	12	~ 1h	~ 18750	~ 353	56.76	0.0731	+ - 0.1301
1.5	12	~ 1h	~ 18500	~ 329	60.67	0.1098	+ - 0.1954
2	11	~ 1h	~ 16750	~ 316	62.87	0.0716	+ - 0.1289
3	11	~ 1h	~ 16500	~ 311	64.90	0.1017	+ - 0.1831
4	11	~ 1h	~ 16200	~ 305	65.70	0.1151	+ - 0.2071
5	11	~ 1h	~ 15750	~ 297	66.17	0.1097	+ - 0.1975
6	11	~ 1h	~ 15750	~ 297	66.60	0.1095	+ - 0.1971
7	11	~ 1h	~ 15800	~ 301	67.60	0.1134	+ - 0.2041
8	11	~ 1h	~ 15750	~ 297	66.95	0.1258	+ - 0.2265
9	11	~ 1h	~ 15800	~ 301	66.93	0.0928	+ - 0.1671
10	11	~ 1h	~ 15700	~ 297	66.82	0.1476	+ - 0.2657
11	11	~ 1h	~ 15800	~ 297	67.00	0.0737	+ - 0.1327
12	11	~ 1h	~ 15600	~ 286	66.74	0.1034	+ - 0.1862

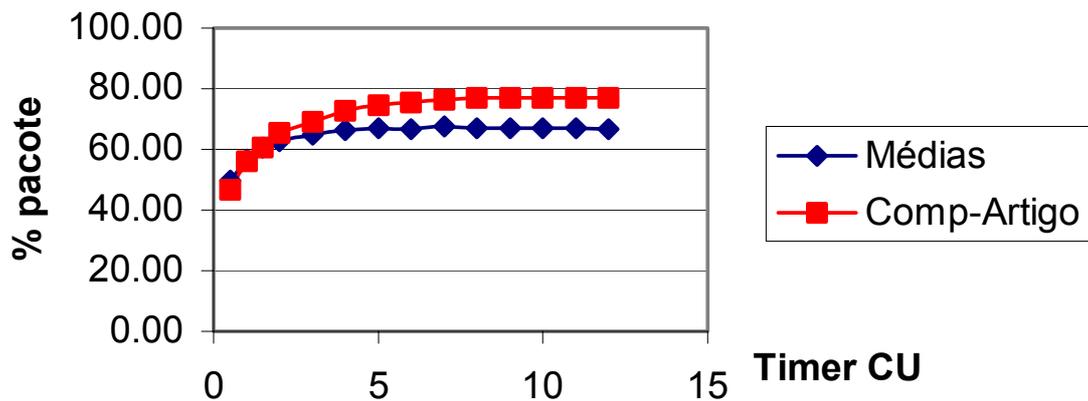


FIGURA 6.6 - Comparação da densidade do pacote obtida pelo simulador

Por este outro gráfico, percebe-se que a tendência de crescimento da densidade com o aumento do Timer\_CU ocorre normalmente no simulador. Porém, existe uma baixa de aproveitamento que pode ser atribuída a diferenças no modelo de tráfego, como descrito no exemplo anterior. Durante todo o arquivo de log existem diversos pacotes com a densidade máxima, que neste caso é de 77%, porém para se levantar o aproveitamento do protocolo é necessário se obter a média que visivelmente ficou abaixo do obtido pelo artigo [JAI 99].

Semelhante ao caso anterior existe um afastamento do valor desejado, mas o comportamento ou tendência do modelo é o mesmo. A medida que cresce o tempo de espera, cresce a utilização da célula pois pode-se aguardar por mais pacotes. O valor de pico em simulação, normalmente é difícil de se determinar devido a variabilidade estatística, mas neste caso, existe um limite máximo que é a ocupação total do pacote; e pode-se observar nos logs um número de amostras iguais a esse valor crescente com o aumento do TimerCU.

A última comparação é realizada com os artigos da General DataComm [MCL 97b] que na verdade não é informado se os dados foram obtidos por simulação ou não. Mas isso não é essencial, pois mesmo que os dados sejam obtidos de seus equipamentos ou calculados por algum modelo analítico, pode-se utiliza-los para comparar com a simulação. No entanto, surge um problema, como descobrir quais são as condições dos testes que geraram esses dados, tendo em vista que existem diversas variáveis que podem influenciar no resultado, conforme pode ser visto até agora.

Alguns parâmetros podem ser obtidos do texto, tais como o número de fontes e sua velocidade de transmissão (32Kbps ou 64 Kbps), mas do modelo de tráfego só é dito que existe supressão de silêncio com 50 % de período ativo e 20 % dos canais sempre em espera. Para traduzir em termos da configuração do NIST foi adotado o tempo do período ativo proposto por Jain [JAI 99], por ser um pesquisador que já tem trabalhos divulgados no ATM Forum. Assim, nosso modelo ficou com rajadas de 352 ms e intervalos também do mesmo tamanho para realizar os 50 % de silêncio. Mas no caso dos canais em espera, não existe como realizar isto na configuração do simulador, a não ser que se reduza manualmente o número de fontes de tráfego.

Assim, o teste ficou dividido em 4 : “a” com 32 Kbps, “b” com 64 Kbps, “c” com 32 Kbps menos 20% dos canais, e “d” com 64 Kbps menos 20 % também. Foi utilizada a tabela que mostra o número de canais em um tronco DS1 para realizar a comparação. Porém, como não mudou-se a quantia de fontes, foi observado se a largura de banda utilizada se aproximava do valor esperado, ou seja, 1536 Kbps.

Como pode ser observado na tabela 6.9, os valores são bem mais próximos nos casos “c” e “d” que retira parte dos canais como declarado no artigo. Porém, existe uma diferença onde o relógio Timer\_CU é menor, pois obteve-se valores abaixo do esperado. Este fato pode ser devido ao desconhecimento de outros fatores que possam modificar a simulação, tais como tempos associados ao empacotamento e que tem sua influência maior quando o tempo de espera por pacotes é reduzido. Apesar das diferenças, o aproveitamento da banda é bem semelhante, e pode-se dizer que o modelo é apropriado.

TABELA 6.9 - Comparação do numeros de canais utilizando um tronco DS1.

(Arquivo) replicações	(teste4a) 10				(teste4b) 10			
timer_CU	Total de Canais	Média	Desvio Padrão	Intervalo de Confiança (+ - valor)	Total de Canais	Média	Desvio Padrão	Intervalo de Confiança (+ - valor)
1 ms	55	1206.95	13.72	25.11	35	1558.98	63.83	116.81
2 ms	70	1512.12	35.96	65.80	40	1725.21	55.13	100.88
3 ms	77	1644.65	25.90	46.87	42	1748.74	50.79	91.93
4 ms	82	1790.71	30.20	54.65	42	1777.24	23.66	42.82
5 ms	85	1864.56	24.55	44.44	45	1951.28	43.90	79.46
6 ms	85	1820.69	31.59	57.17	45	1844.66	56.80	102.81
7 ms	87	1905.03	46.44	84.05	45	2100.96	40.49	73.29
8 ms	87	1899.66	34.63	62.68	45	1949.28	47.90	86.70
9 ms	90	1928.79	31.48	56.98	45	1950.04	42.90	77.65
10 ms	90	1972.53	34.68	62.76	45	1942.81	38.88	70.38
11ms	90	1966.50	28.23	51.09	47	2079.10	61.11	110.61
12 ms	92	1943.90	51.44	93.11	47	2010.76	51.26	92.77
13 ms	92	1996.71	37.29	67.50	47	2002.06	44.99	81.44
(Arquivo) replicações	(teste4c) 10				(teste4d) 10			
timer_CU	Total de Canais	Média	Desvio Padrão	Intervalo de Confiança (+ - valor)	Total de Cana is	Média	Desvio Padrão	Intervalo de Confiança (+ - valor)
1 ms	44	1001.27	13.62	24.93	28	1214.98	27.68	50.65
2 ms	56	1196.32	20.25	37.06	32	1420.14	41.91	76.69
3 ms	62	1377.87	21.77	39.40	34	1520.38	33.39	60.44
4 ms	66	1434.47	23.60	42.72	34	1434.52	32.24	58.35
5 ms	68	1553.21	34.44	62.33	36	1617.82	42.92	77.69
6 ms	68	1480.72	26.24	47.50	36	1546.35	50.59	91.56
7 ms	70	1526.69	27.68	50.10	36	1557.31	33.49	60.62
8 ms	70	1552.99	26.87	48.64	36	1493.06	36.54	66.13
9 ms	72	1568.73	33.90	61.36	36	1559.00	51.31	92.88
10 ms	72	1515.55	34.01	61.55	36	1612.87	54.02	97.78
11ms	72	1495.55	29.49	53.37	38	1639.53	32.15	58.18
12 ms	74	1611.08	36.92	66.82	38	1629.38	51.39	93.01
13 ms	74	1589.15	20.12	36.42	38	1609.04	51.73	93.64

## 7 Conclusão

Ao dividirmos os objetivos em dois, estudar a funcionalidade do protocolo de adaptação AAL2, e criar um ambiente para simular e comprovar a performance deste protocolo, eles acabaram se misturando ao final quando se foram realizados diversos testes no intuito de demonstrar a correta implementação do AAL2 no simulador NIST. Quando os testes estavam sendo realizados, ficou mais clara a utilização deste protocolo e suas vantagens no emprego do tráfego de VBR; mostrando facilmente a diminuição da largura de banda necessária.

Conforme visto no capítulo 6 , o novo módulo do simulador funciona perfeitamente e pode ser utilizado normalmente como parte do original. Porém, existiram diferenças nos valores esperados, na maioria dos casos devido a comparação ter sido realizada com artigos em que as condições empregadas não foram idênticas. E algumas vezes não eram divulgadas, levando a adotar um valor escolhido por algum critério de teste, mas que pode não ter sido próximo do utilizado no modelo comparado. Isto pode ter se refletido em todos os estágios do modelo, inclusive na determinação da fonte de tráfego, que apesar de todos os cuidados aqui utilizados pode ainda ser mais testada, ou ainda testar outras fontes de tráfego diferentes. Logicamente, existem também as influências estatísticas e diferenças de implementação do modelo que sempre causam pequenas diferenças de resultados.

Comparações mais precisas só surgirão quando pudermos contar com equipamentos usando esse protocolo, mas baseado na comparação com artigos publicados pôde-se verificar e validar as simulações. No entanto, fica claro que a precisão do modelo pode não ser boa o suficiente para realizar projetos ou pesquisas baseadas nos valores da simulação, mas pode-se utilizar para estudar o comportamento dinâmico do protocolo frente a alterações na rede. E futuramente, pode-se passar por uma etapa de calibração a fim de tornar os cálculos mais exatos.

A grande vantagem da utilização de um simulador de código aberto, é poder contar com a possibilidade de alteração de qualquer parâmetro indesejado, ou criar variáveis para medir valores até então não utilizados. Além disso, o NIST tem vários anos de experiência em simulação e esta ferramenta já foi bastante utilizada. Depois das modificações, pode-se contar com a possibilidade de simular mais um protocolo, além de poder valer-se de um depurador (debug) específico para este protocolo.

A verificação foi realizada sem problemas, isto porque houve sempre a preocupação de não corromper o funcionamento padrão do simulador, e desta forma, foram criadas estruturas semelhantes às existentes e realizadas mudanças sutis que foram reconhecidas normalmente; criando um ambiente de testes propício para o conhecimento e avaliação da camada AAL.

A partir destas modificações, já se pode pensar em desenvolver os módulos de programação para realizar a simulação de outros AAL, pois tornou-se mais fácil a criação no simulador dos protocolos de adaptação que faltam nas mesmas bases do AAL2. Aliás, ele pode ser empregado como ferramenta de auxílio no aprendizado do protocolo da adaptação, tendo em vista que segue perfeitamente a norma.

Uma proposta de trabalho futuro um pouco audaciosa, seria trabalhar sobre o simulador NIST para rever suas estruturas e provavelmente refazê-las, principalmente na parte gráfica que apesar de ajudar, ainda deixa muito a desejar. Um grande avanço seria criar um módulo para inferência estatística, ou seja, proporcionar dados como média, pico, variância, desvio-padrão e intervalo de confiança. Outra implementação que pode auxiliar é uma interface geradora de saídas para a ferramenta *gnuplot*; auxiliando na coleta dos dados e visualização dos resultados. As universidades brasileiras estão usando este simulador como ferramenta auxiliar para pesquisas de simulação em redes ATM [JIR 2000], o que é mais um ponto forte para adotá-lo e incrementá-lo.

O ATM é uma das opções fortes para construção de redes WANs de alta velocidade, mas quase sempre se conhecem e se utilizam os protocolos AAL1 e AAL5, entretanto é necessário divulgar este novo protocolo de adaptação, que como pode ser visto, não substitui os demais e sim complementa.

O maior custo da utilização deste protocolo é o "overhead" devido a utilização de mais cabeçalhos, porém, isto não é tão importante quando comparado às técnicas de preenchimento parcial da célula empregadas no AAL1. Aliás, torna-se cada vez menos significativo a medida que aumenta o número de usuários do AAL devido ao fato deste protocolo apresentar uma boa eficiência na multiplexação estatística destes serviços. Como o novo padrão denominado AAL 2 trata de serviços do tipo VBR (taxa de bits variável) com vinculação temporal, futuramente podem-se estudar novas aplicações. Mas para isso, precisa-se conhecer melhor os limites de utilização deste protocolo, e para este fim o simulador modificado pode ser uma ferramenta bastante útil.

Sabendo-se do crescimento da telefonia celular e da busca de tecnologias como do voz sobre IP, é importante observar que as vantagens do AAL2 possibilitam mais uma forma de controle sobre o tráfego de rede, que pode ser essencial em aplicações em tempo real, principalmente, quando deseja-se uma integração dos meios de comunicação. Hoje, já se buscam alternativas de migração dos antigos equipamentos PBXs para centrais modernas conectadas a rede, ou estudos para atualizar as empresas de telecomunicação para suportarem o ATM [GDC 2000].

Em um futuro próximo, deverão ser estabelecidas redes ATM em diversas cidades brasileiras, tais como no projeto METROPOA e o da Rede Metropolitana de Florianópolis, que a princípio serão basicamente de dados, mas podem vir a integrar mais serviços necessários às comunidades que fazem parte destes projetos, visto que muitos dos integrantes são instituições de ensino superior que, normalmente, têm diversos campos de pesquisa, tais como o ensino à distância.

## Anexo 1 Exemplos da Recomendação I.363.2 para a Segmentação e Multiplexação do AAL2

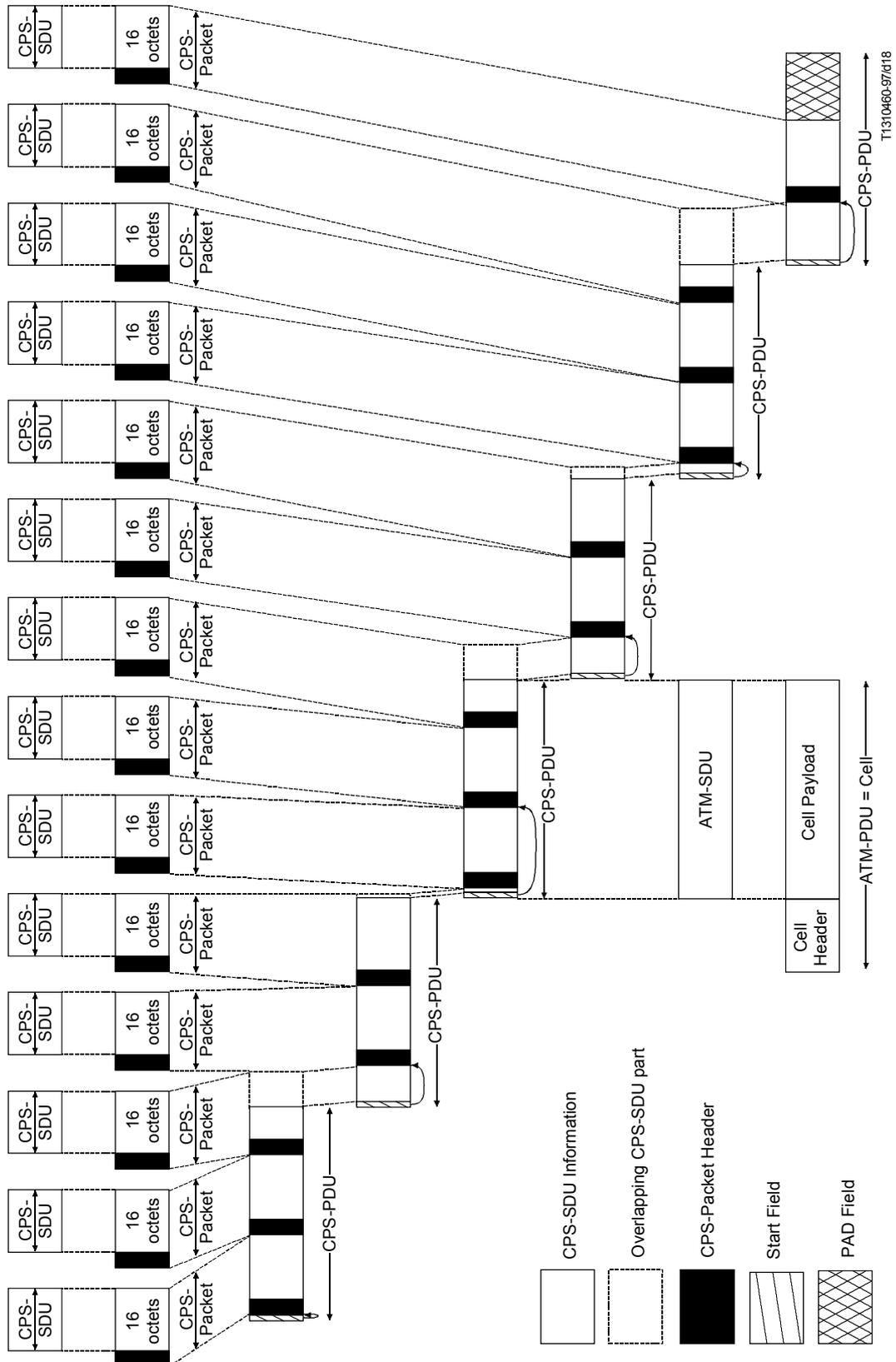


Figure I.1/I.363.2 – Multiplexing and packing CPS-Packets into CPS-PDUs (ATM cells)

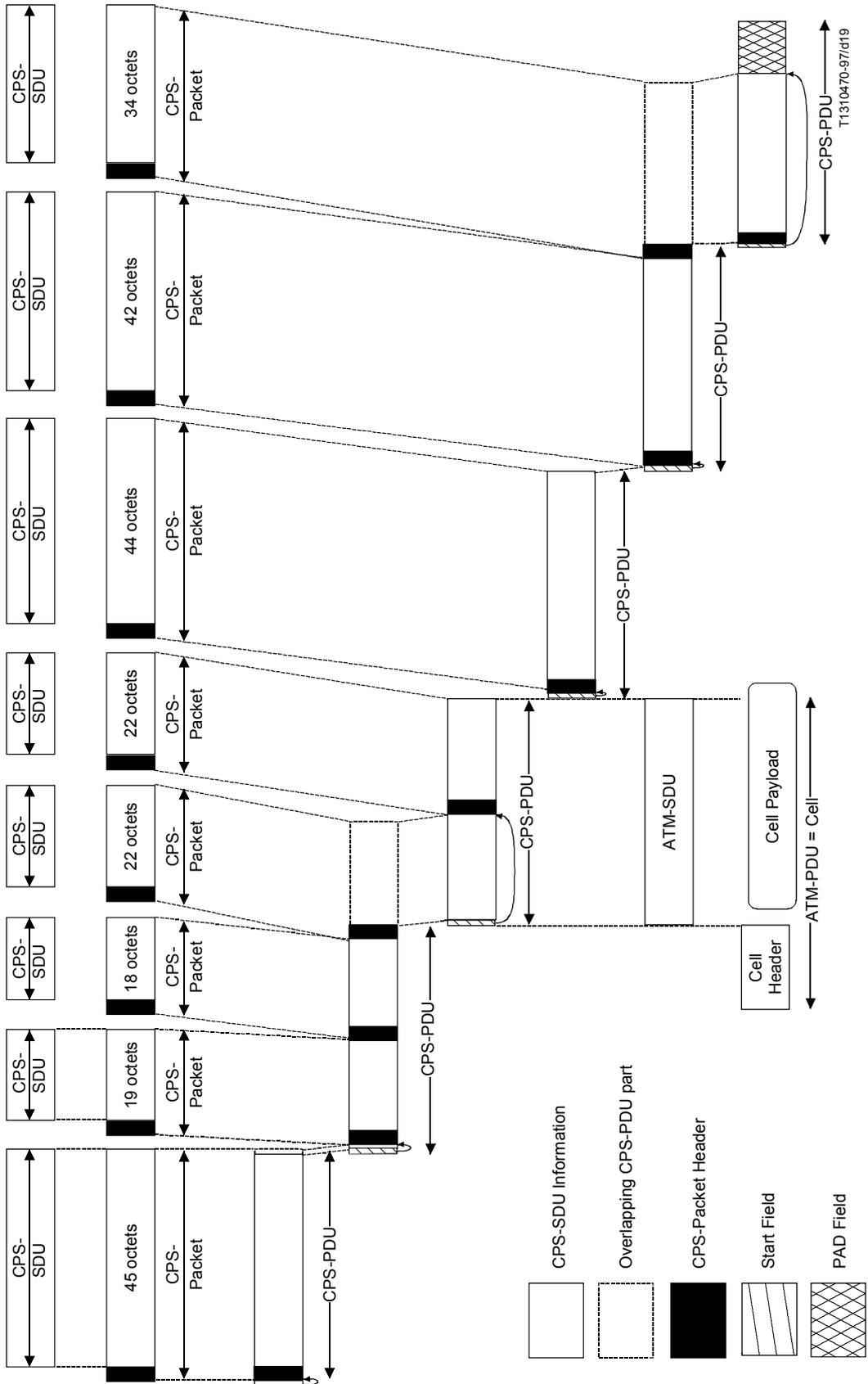


Figure I.2/I.363.2 – Multiplexing and packing of special case CPS-Packets into CPS-PDUs (ATM cells)

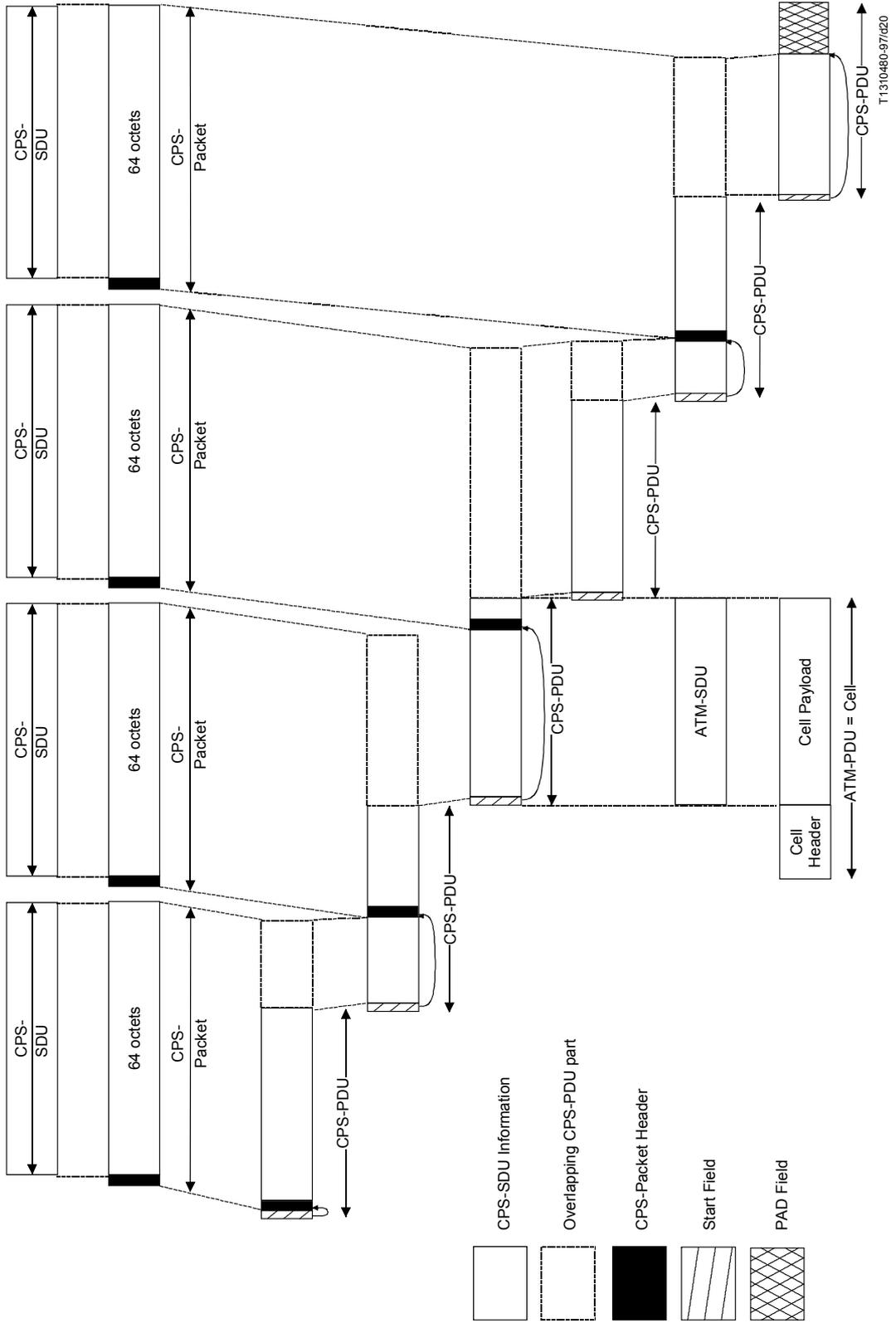
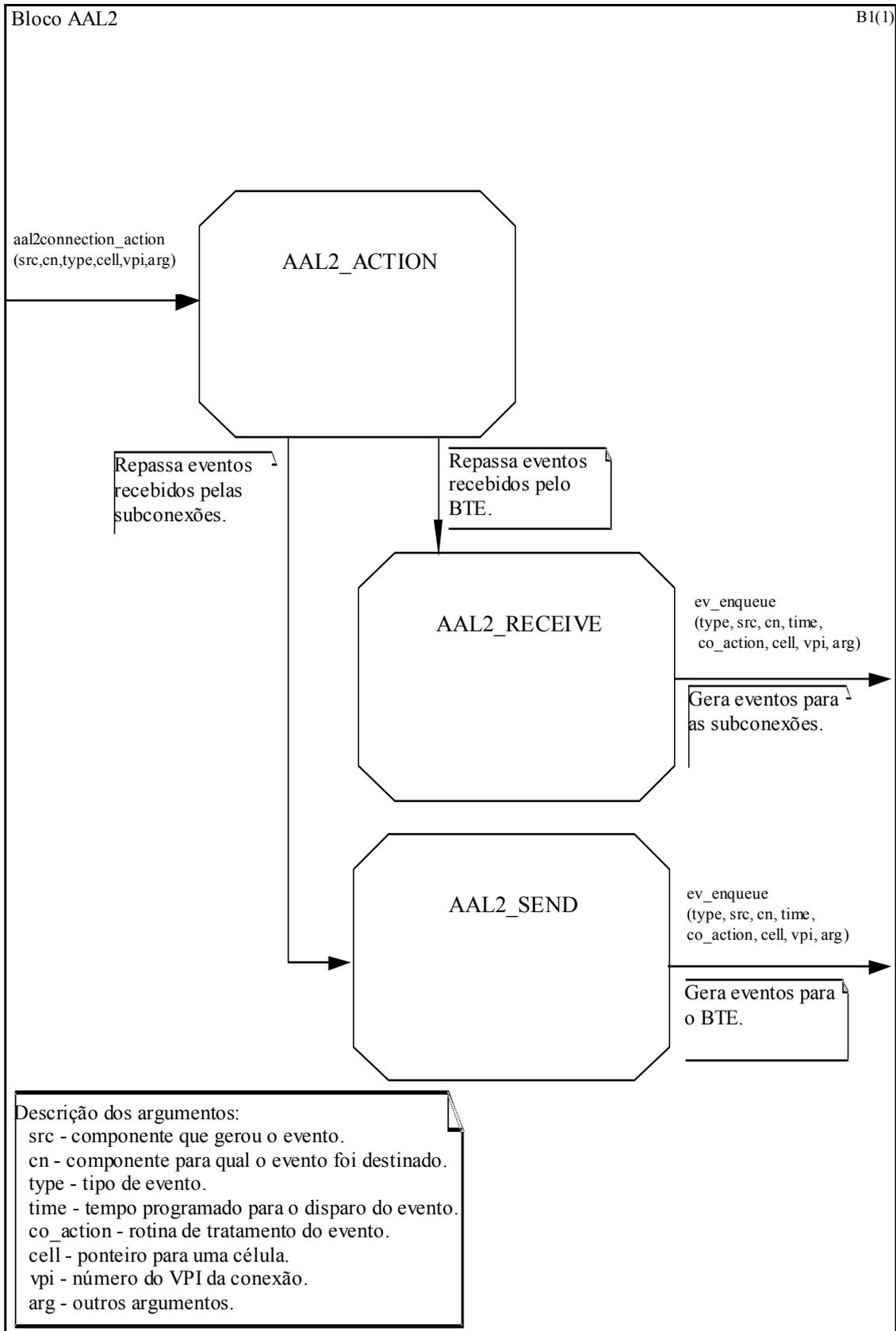
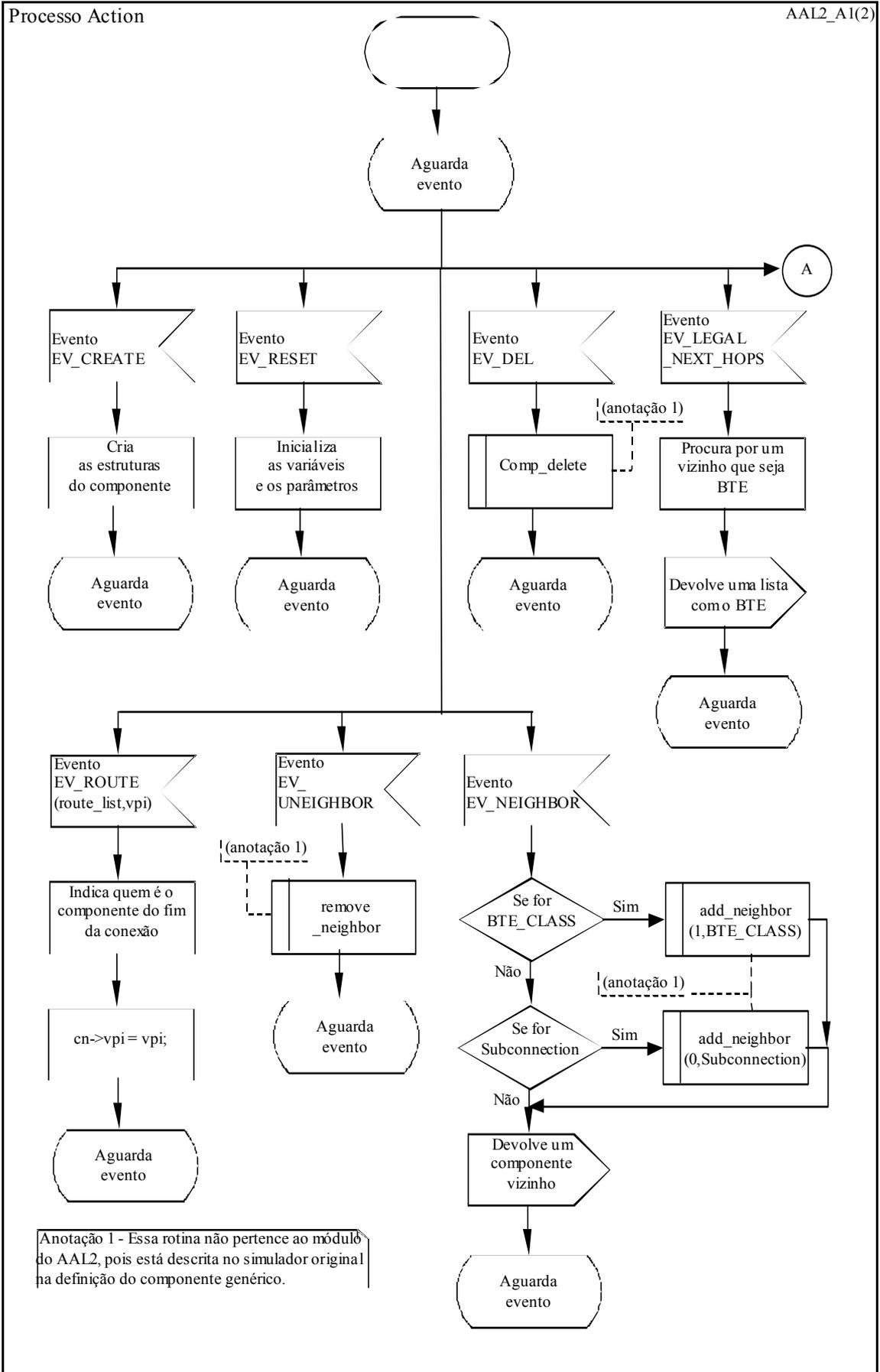
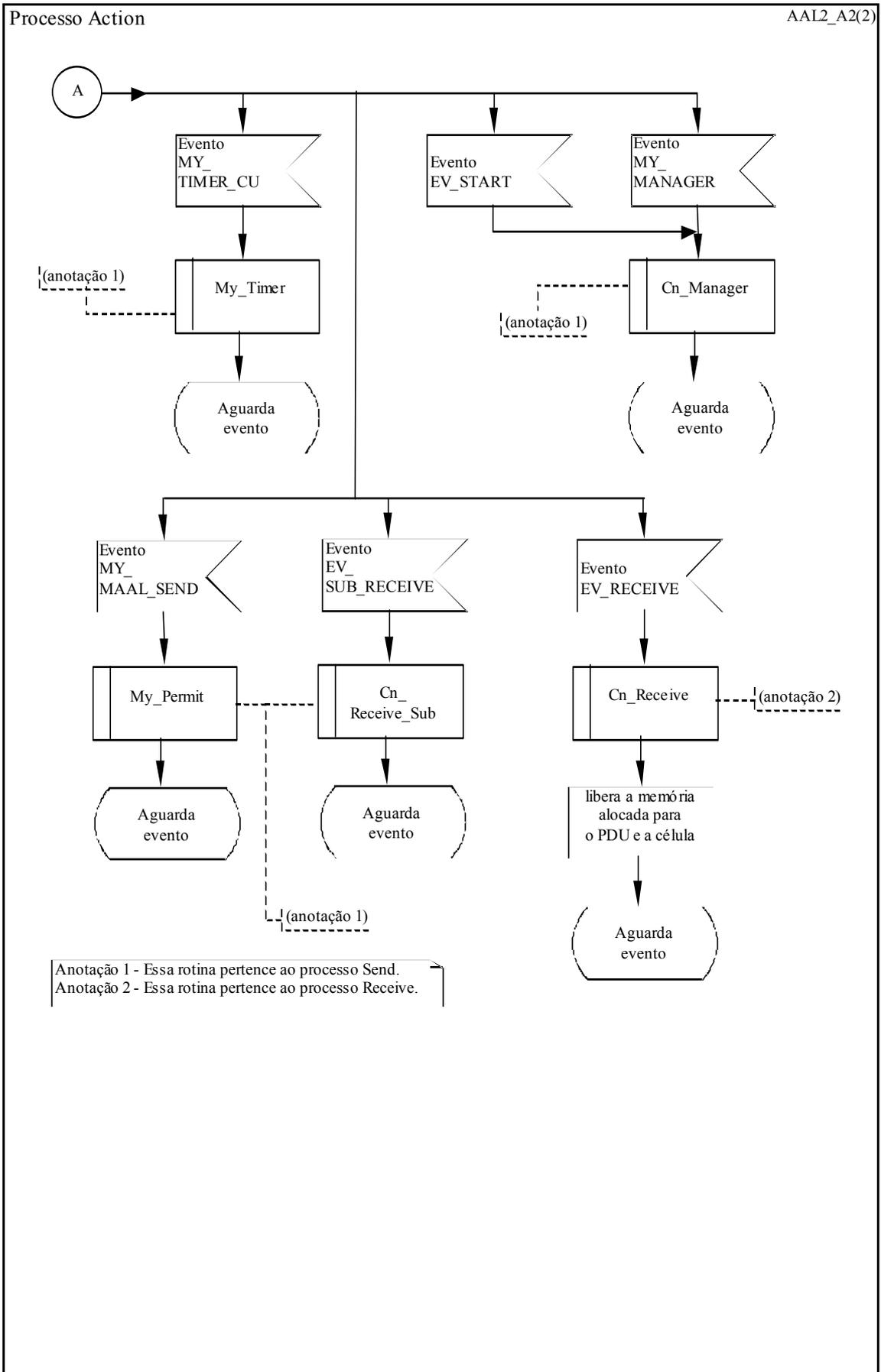


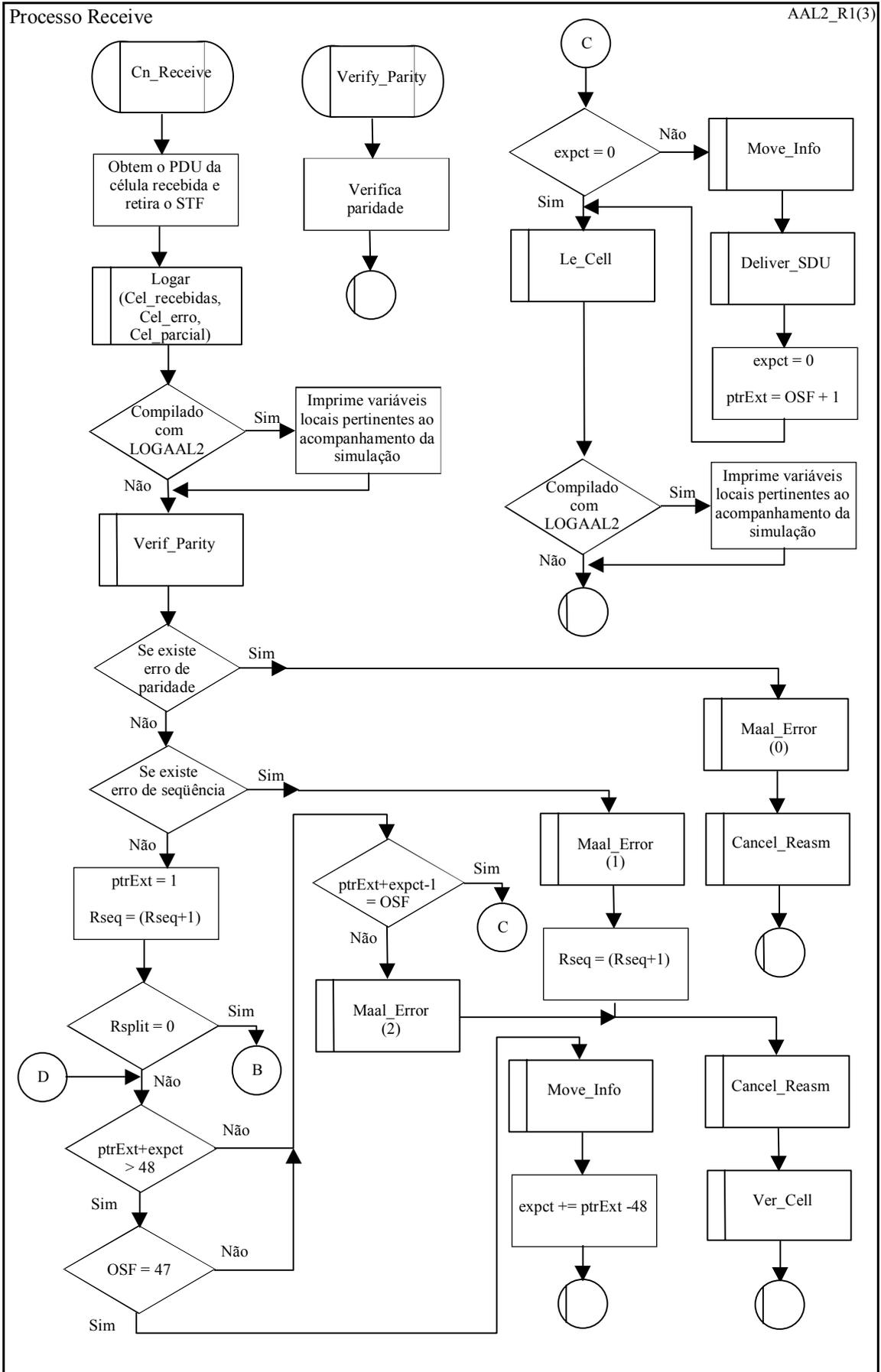
Figure I.3/I.363.2 – Multiplexing and packing of maximum size CPS-Packets into CPS-PDUs (ATM cells)

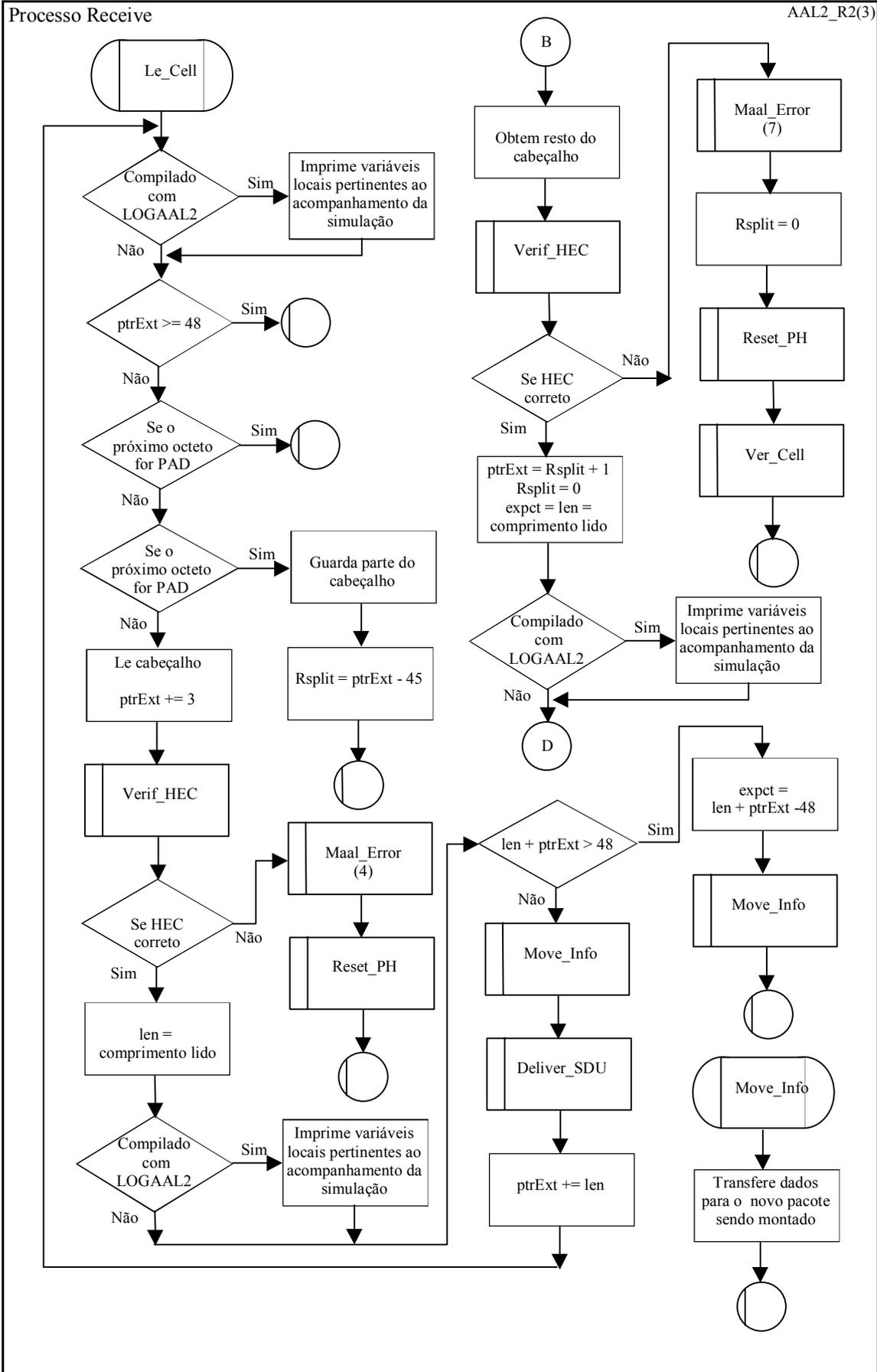
## Anexo 2 Diagramas SDL do Módulo AAL2 criado para o Simulador NIST

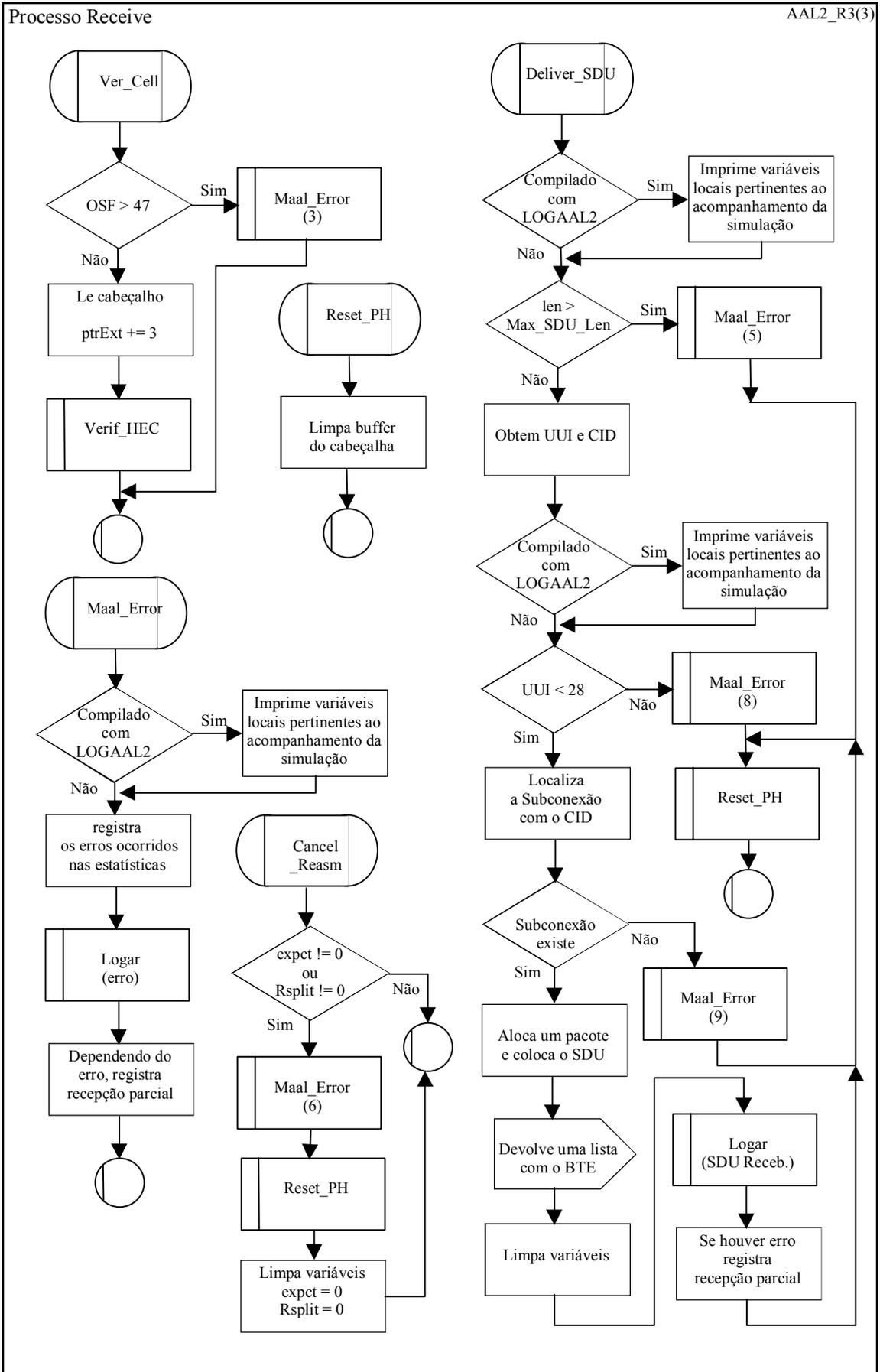


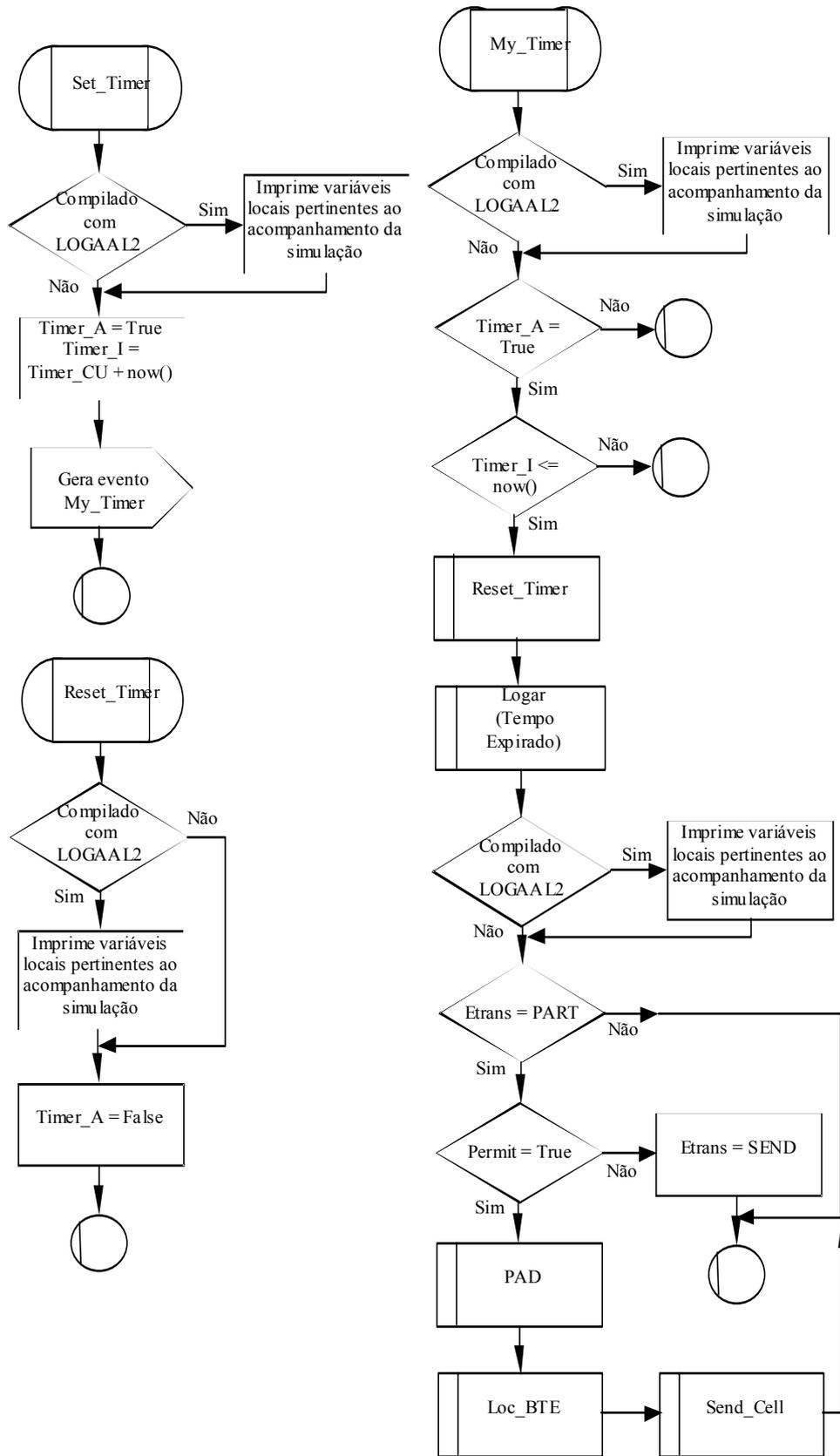


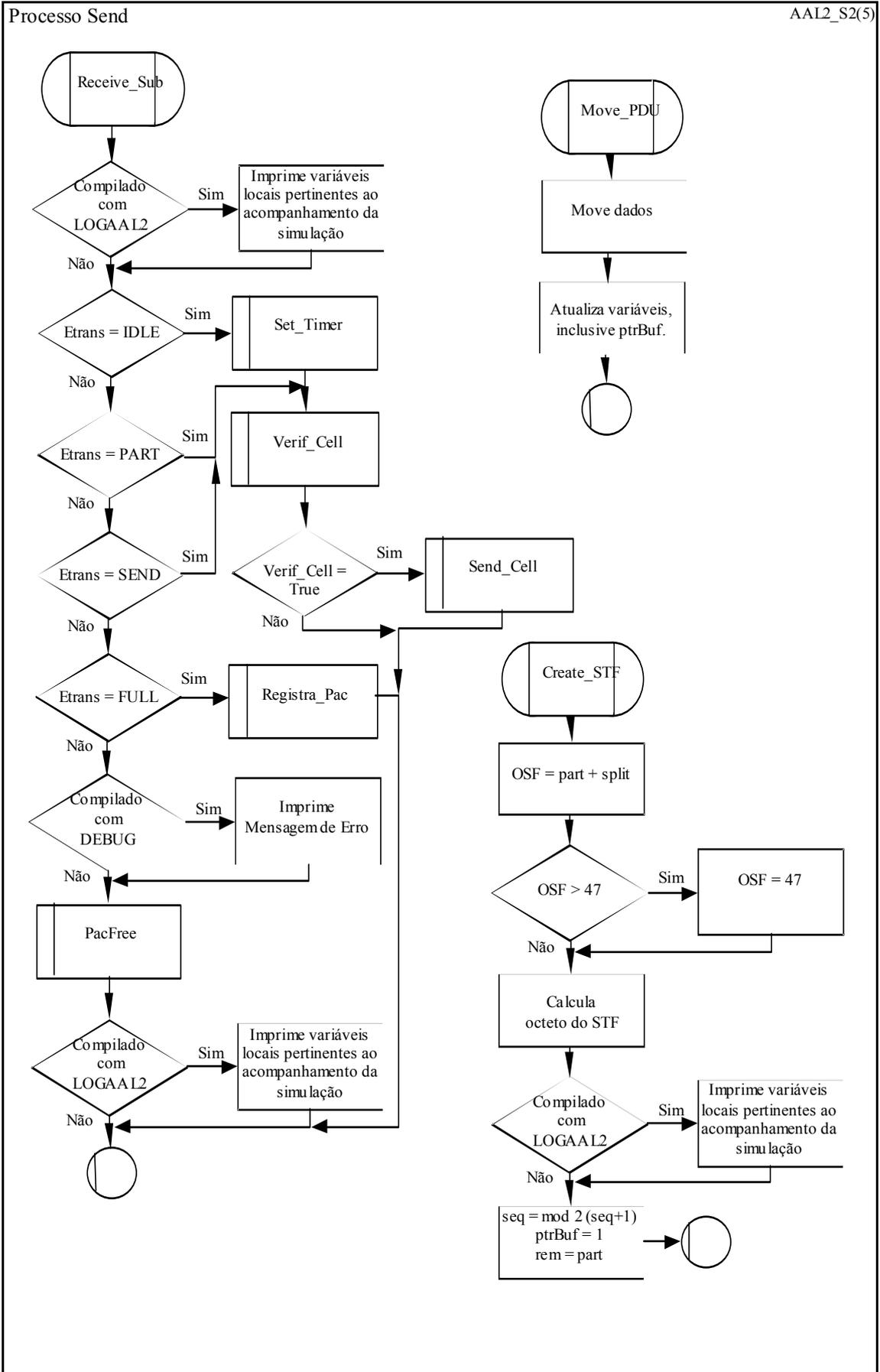


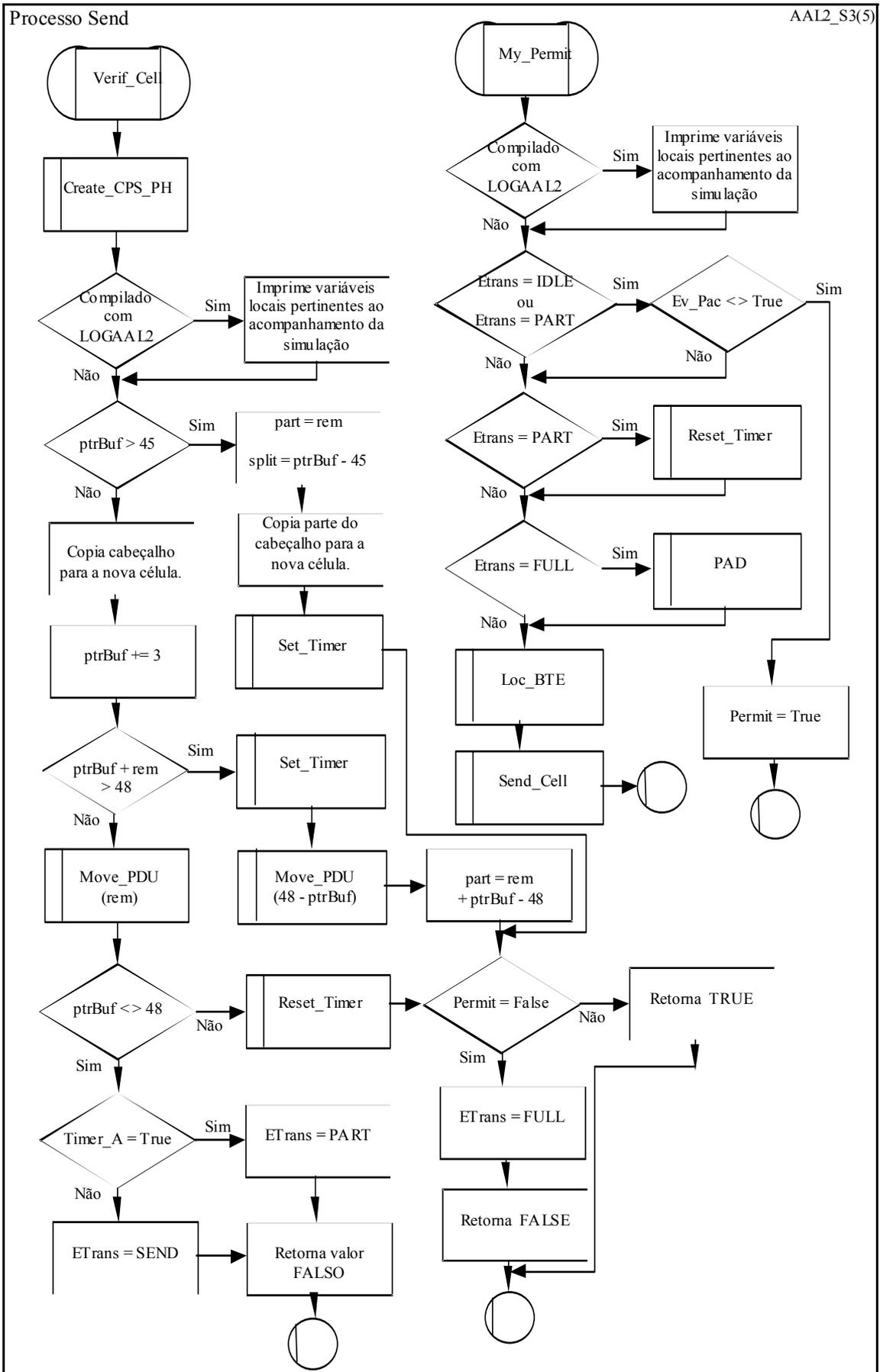


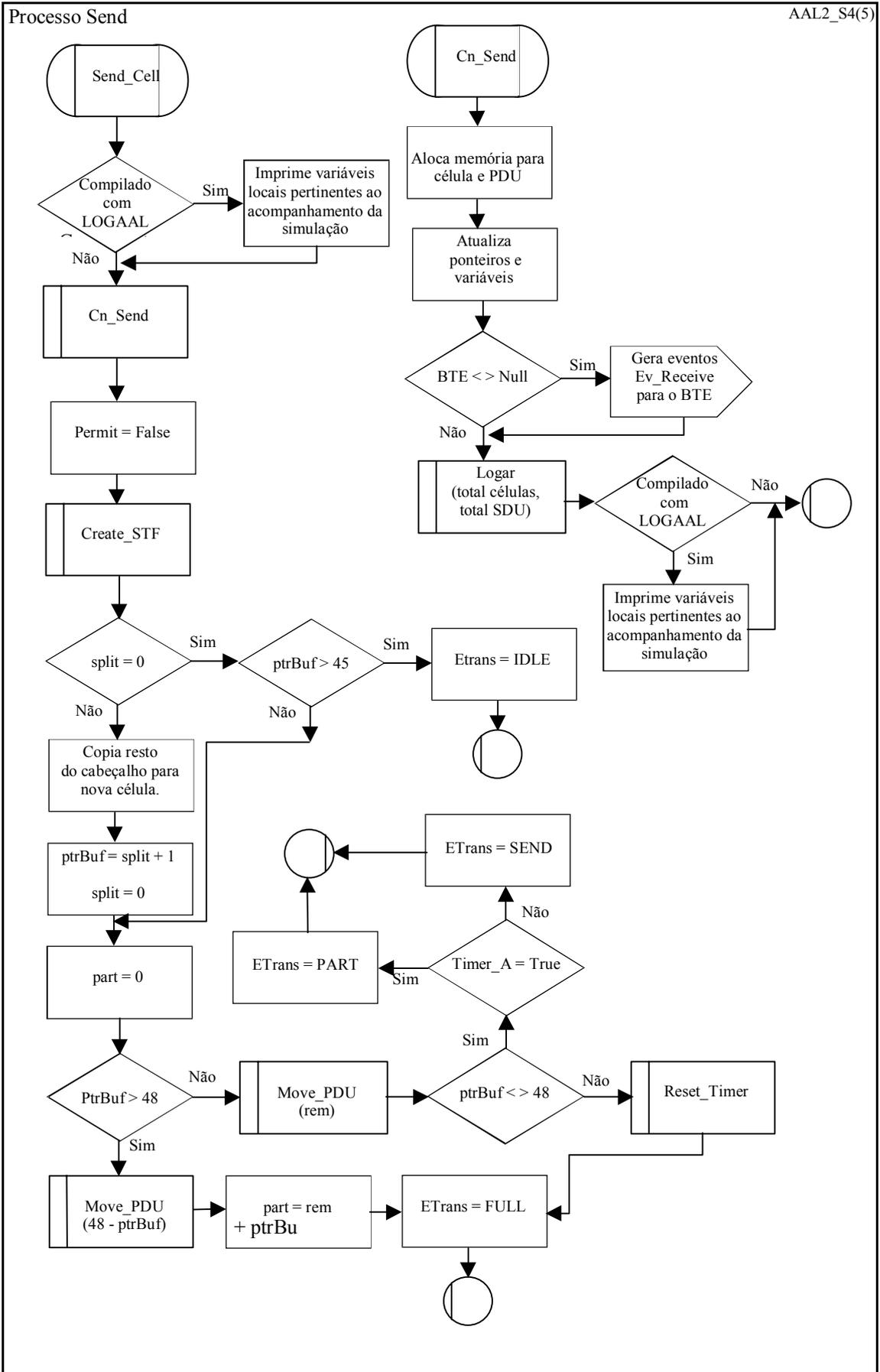






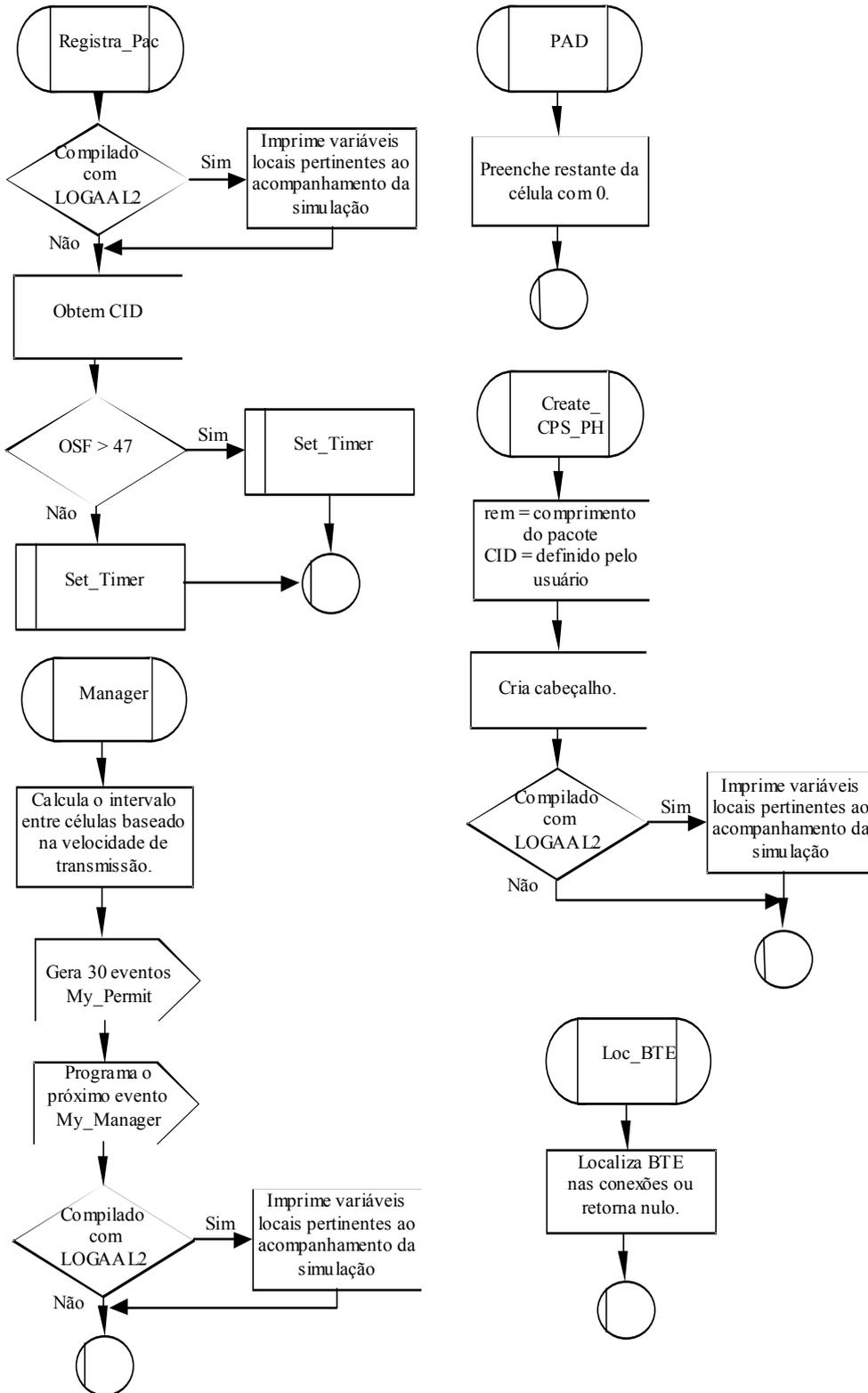






Processo Send

AA12\_S5(5)



## Anexo 3 Exemplos de Arquivos do Simulador

### A) Arquivo de configuração do simulador NIST

```

component 'Switch-1' SWITCH 80 280
param 'Switch-1' 32 0 # Switch-1
param 0 12 0 # Delay to process a cell (uSec): 0
param 155 12 0 # Slot time (Mbit/s): 155
param 300 12 0 # Output q_size (cells, -1=inf): 300
param 200 12 0 # High Threshold (Cells): 200
param 100 12 0 # Low Threshold (Cells): 100
param 2 12 0 # buf mgmt: EPD (0), FBA(1), RED(2), FQ(3): 2
param 100 12 0 # EPD Threshold (Cells): 100
...../ /...../ /.....
pflags 2a 4 #Sigma Pos for Q toLink-A1: 0
pflags 2a 4 #Ratio for Q toLink-A1: 0

```

Cada componente é descrito em seqüência num pequeno bloco.

```

component 'Host-AAL' BTE 220 280
param 'Host-AAL' 32 0 # Host-AAL
param 300 12 0 # Max Output Queue Size(-1=inf): 300
param 300 12 0 # Max Input Queue Size(-1=inf): 300
pflags 2a 4 #Cells Received: 0
param 1 12 0 # Logging every (ticks) (e.g. 1, 100): 1
param 5000000 12 0 # Stop sending at (usecs): 5000000
pflags 2a 4 #Cells in VBR Q to Link-A1: 0
pflags 2a 4 #Cells dropped in VBR Q toLink-A1: 0
pflags 2a 4 #Cells in ABR Q to Link-A1: 0
pflags 2a 4 #Cells Dropped in ABR Q to Link-A1: 0
pflags 2a 4 #Cells in UBR Q to Link-A1: 0
pflags 2a 4 #Cells Dropped in UBR Q to Link-A1: 0

```

A primeira parte identifica o componente e o localiza no plano da interface gráfica.

A segunda parte traz todos os parâmetros do componente. Aqueles que podem ser configurados pelo usuário são definidos pela partícula “param” seguida de seu valor. Os demais são opcionais, e servem para visualizar os dados através do log. Neste segundo caso, após a partícula “pflags” é especificado por uma flag o tipo de gráfico, se deve gerar logs, e outros detalhes deste variável.

```

component 'Host-AAL-1' BTE 379 361
param 'Host-AAL-1' 32 0 # Host-AAL-1
...../ /...../ /.....
pflags 2a 4 #Cells Dropped in UBR Q to Link-A11: 0

```

```

component 'Link-A1' LINK 150 280
param 'Link-A1' 32 0 # Link-A1
param 155 12 0 # Link Speed (Mbits/sec): 155
...../ /...../ /.....
param 10 12 0 # Distance (Km): 10
pflags 2a 4 #Link rate (Mbit/s) to Host-AAL: 0
pflags 2a 4 #Link rate (Mbit/s) to Switch-1: 0

```

```

component 'BBT' LINK 205 208
param 'BBT' 32 0 # BBT
param 155 12 0 # Link Speed (Mbits/sec): 155
pflags 2 0
param 10 12 0 # Distance (Km): 10
pflags 2a 4 #Link rate (Mbit/s) to Switch-1: 0
pflags 2a 4 #Link rate (Mbit/s) to Switch-2: 0

```

```

component 'teste1' AAL2CONNECTION 193 344
param 'teste1' 32 0      # teste1
param 10 12 0  # Bit Rate (Mbits/s): 10
param 10000 12 0      # Timer_CU em Us: 10000
param 0 12 0  # Envia pacotes sempre q recebe MAAL_SEND: 0
param 64 12 0  # Maximo SDU Comprimento: 64
param 0 12 0  # Atraso no processo de recepção em Us: 0
param 0 12 0  # Atraso no processo de Transmissão em Us: 0
param 100 12 0 # Qtde de ticks entre cada log: 100
pflags 2a 4  #Total de SDU recebidos sem erro: 0
pflags 2a 4  #Total de células recebidas: 0
pflags 2a 4  #Total de células recebidas erradas: 0
pflags 2a 4  #Total de PDU parcial corretos recebidos: 0
pflags 2a 4  #Total de erros de paridade (Erro 0): 0
pflags 2a 4  #Total de erros de sequencia (Erro 1): 0
pflags 2a 4  #Erros qtde de octetos experado (Erro 2): 0
pflags 2a 4  #Total de PDU discartado (Erro 3): 0
pflags 2a 4  #Total de erros de CRC (Erro 4): 0
pflags 2a 4  #Erros de comprimento excedido (Erro 5): 0
pflags 2a 4  #Discarte de SDU sendo montados (Erro 6): 0
pflags 2a 4  #Erros de CRC em cab. montados (Erro 7): 0
pflags 2a 4  #Total erros de CID reservado (Erro 8): 0
pflags 2a 4  #Erros de CID não esperado (Erro 9): 0
pflags 2a 4  #Total de vezes que o Timer_CU expirou: 0
pflags 2a 4  #Total de pac que ultrapassa o Timer_CU: 0
pflags 2a 4  #Total de SDU enviados: 0
pflags 2a 4  #Total de células enviados: 0
pflags 2a 4  #SDU dados perdidos por passar o ATC: 0
pflags 2a 4  #SDU Gerencia perdidos por passar o ATC: 0
pflags 22 0

```

Esta é a descrição do novo componente AAL2, e como pode ser observado, mantém a mesma forma padrão do NIST. Abaixo começam a surgir os parâmetros que são em sua maioria comentados no

```

component 'sub1' VBRSUBCONNECTION 208 411
param 'sub1' 32 0# sub1
param 0.032 12 0# Bit Rate (Mbits/s): 0.013
param 100000 12 0  # Mean Burst Length (usecs): 100000
param 100000 12 0  # Mean Interval Between Bursts (usecs): 20000
param 1 12 0  # Start time (usecs): 1
param 100000 12 0  # Number of Mbits to be sent: 100000
param 64 12 0  # Length of packet: 32
param 2 12 0  # Numero identificador da conexão: 2
param 100 12 0  # Qtde de ticks entre cada log: 100
pflags 6e 4  #Total de pacotes enviados: 0
pflags 6e 4  #Vel. de transmissão em Kb/s: 0

```

Esta é a descrição do novo componente SubVBR, que também mantém a mesma forma padrão do NIST. São os mesmos parâmetros da aplicação VBR original.

```

neighbor1 'Switch-1' 'BBT'
neighbor1 'Switch-1' 'Link-A1'
neighbor1 'Host-AAL' 'teste1'
neighbor1 'Link-A1' 'Host-AAL'
neighbor1 'Link-A1' 'Switch-1'
neighbor1 'BBT' 'Switch-1'
neighbor1 'BBT' 'Switch-2'
...../ /...../ /.....
neighbor1 'sub4' 'teste2'

```

Após os blocos que descrevem os componentes, vem um bloco indicando quem é vizinho de quem. Isto serve para criar as ligações.

A última parte são as rotas da rede criada. Cada rota é uma seqüência de componentes na ordem em que eles estão ligados.

```

route1 'teste1' 'Host-AAL' 'Link-A1' 'Switch-1' 'BBT' 'Switch-2' 'Link-A11' 'Host-AAL-1' 'teste2'

```

## B) Arquivo de SNAP do simulador NIST

```

# Seed 821240736
# Time of snapshot (ticks) 1662567594
component 'Switch-1' SWITCH 80 280
param 'Switch-1' 32 0 # Switch-1
param 0 12 0 # Delay to process a cell (uSec): 0
param 155 12 0 # Slot time (Mbit/s): 155
...../ /...../ /.....

component 'Link-A11' LINK 0 0
param 'Link-A11' 32 0 # Link-A11
param 155 12 0 # Link Speed (Mbits/sec): 155
...../ /...../ /.....
pflags 2e 4 #Link rate (Mbit/s) to Switch-2: 0
pflags 2e 4 #Link rate (Mbit/s) to Host-AAL-1: 0.327488

component 'teste1' AAL2CONNECTION 0 0
param 'teste1' 32 0 # teste1
...../ /...../ /.....
pflags 2e 4 #Erros de CID não esperado (Erro 9): 0
pflags 2e 4 #Total de vezes que o Timer_CU expirou: 11851
pflags 2e 4 #Total de pac que ultrapassa o Timer_CU: 0
pflags 2e 4 #Total de SDU enviados: 56971
pflags 2e 4 #Total de células enviados: 31849
pflags 2e 4 #SDU dados perdidos por passar o ATC: 29
pflags 2e 4 #SDU Gerencia perdidos por passar o ATC: 66
pflags 26 4 #% do total da célula: 37.7358
pflags 22 0

component 'teste2' AAL2CONNECTION 0 0
param 'teste2' 32 0 # teste2
param 10 12 0 # Bit Rate (Mbits/s): 10
param 4000 12 0 # Timer_CU em Us: 4000
param 0 12 0 # Envia pacotes sempre q recebe MAAL_SEND: 0
param 64 12 0 # Maximo SDU Comprimento: 64
param 0 12 0 # Atraso no processo de recepção em Us: 0
param 0 12 0 # Atraso no processo de Transmissão em Us: 0
param 100 12 0 # Qtde de ticks entre cada log: 100
pflags 2e 4 #Total de SDU recebidos sem erro: 6984
pflags 2e 4 #Total de células recebidas: 3436
pflags 2e 4 #Total de células recebidas erradas: 0
pflags 2e 4 #Total de PDU parcial corretos recebidos: 4
pflags 2e 4 #Total de erros de paridade (Erro 0): 0
pflags 2e 4 #Total de erros de sequencia (Erro 1): 0
pflags 2e 4 #Erros qtde de octetos esperado (Erro 2): 4
...../ /...../ /.....

component 'sub1-1' VBRSUBCONNECTION 0 0
param 'sub1-1' 32 0 # sub1-1
param 0.004 12 0 # Bit Rate (Mbits/s): 0.004
param 352000 12 0 # Mean Burst Length (usecs): 352000
param 352000 12 0 # Mean Interval Between Bursts (usecs): 352000
param 1 12 0 # Start time (usecs): 1
param 100000 12 0 # Number of Mbits to be sent: 100000
param 20 12 0 # Length of packet: 20
param 1 12 0 # Numero identificador da conexão: 1
param 100 12 0 # Qtde de ticks entre cada log: 100

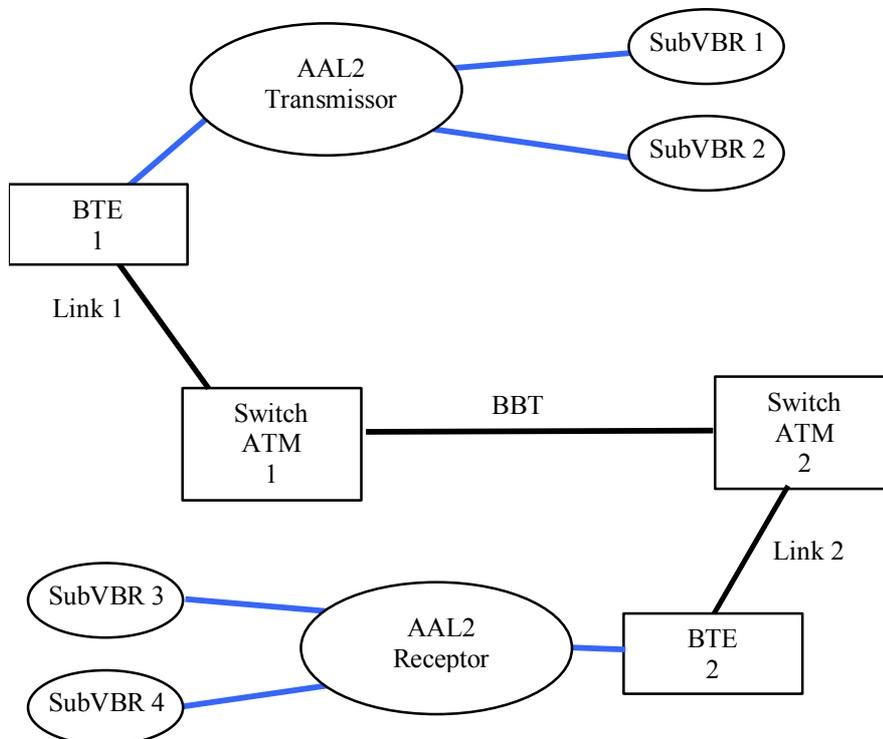
```

A principal diferença entre o arquivo de SNAP e um de configuração, que pode ser vista por todo este texto, é a presença de valores instantâneos no fim de cada linha. Somente os parâmetros "pflags" tem os valores alterados, pois os outros são fixados pelo usuário.

### C) Arquivo de log do simulador NIST

# 1 'Link-A1' 'Link rate (Mbit/s) to Switch-1'	<p>A primeira parte deste arquivo descreve qual os parâmetros que estão sendo gravados e o número que os identifica na segunda parte.</p>
# 2 'teste1' 'Total de SDU enviados'	
# 3 'teste1' 'Total de células enviados'	
# 4 'teste1' 'SDU dados perdidos por passar o ATC'	
# 5 'teste1' 'SDU Gerencia perdidos por passar o ATC'	
# 6 'teste1' '% do total da célula'	
3 400140178 2140	<p>A segunda parte é uma seqüência de linhas na ordem em que elas ocorrem. Cada linha tem um identificador, o tempo em ticks e o valor obtido do simulador.</p>
2 400140178 2865	
6 400140178 75.4717	
1 400140178 0.291714	
3 400210499 2141	
2 400210499 2866	
6 400210499 37.7358	
3 400367700 2142	
2 400367700 2867	
6 400367700 37.7358	
3 400761662 2143	
2 400761662 2868	
6 400761662 37.7358	
3 400842736 2144	
2 400842736 2870	
6 400842736 75.4717	

### D) Diagrama básico da rede de teste



## ***Anexo 4 Modelo do AAL2 com Buffer***

Devido ao fato de muitos dos artigos apresentados possuírem buffers em seus modelos, as comparações ficaram prejudicadas. Como foi dito, o nosso modelo é totalmente baseado na recomendação I.363.2 do ITU-T, a qual deixa a questão de buffers para futuras padronizações. Assim, durante todo o processo de programação e estudo, seguiu-se os diagramas SDL da norma, não ocorrendo uma preocupação com problemas da validação, mesmo porque alguns artigos foram obtidos posteriormente.

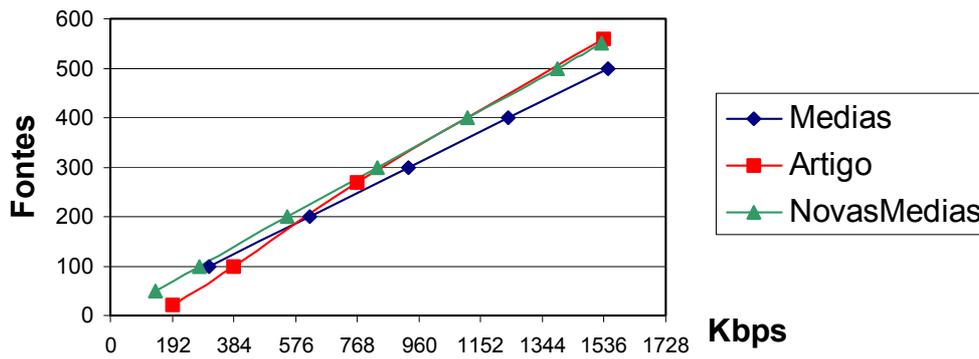
Apesar da tendência ser a esperada e o ganho de performance ser superior ao do AAL1 como desejava-se demonstrar, o modelo pode perder credibilidade por apresentar diferenças tão grandes em relação aos artigos apresentados. Claramente, não dispúnhamos de dados precisos sobre as outras simulações, mas também não se esperava tanta discrepância.

Verificando que as diferenças eram principalmente devido a perda de pacotes, então estudou-se uma maneira de acrescentar um buffer simples e estudar rapidamente sua influência. Esta perda de parte do tráfego vindo das fontes ON-OFF deve-se ao fato de usarmos modelos estatísticos, e desta forma, não se pode prever quando dois pacotes chegam ao mesmo tempo. Isto é tanto mais freqüente, quanto mais cresce a velocidade do tráfego ou o número de fontes. Este problema também ocorre nas redes reais, e é o principal motivo para acrescentar um buffer de entrada.

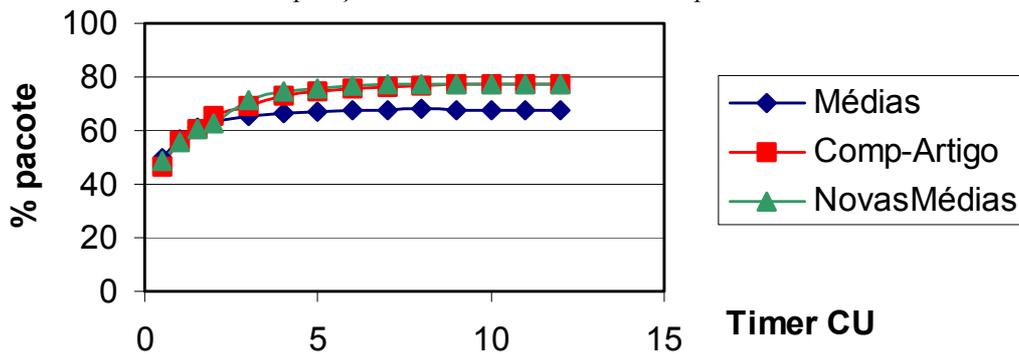
Como não fazia parte do diagrama SDL inicial, optou-se por algumas pequenas modificações no modelo anterior, conforme demonstrado no diagrama extra presente no fim deste anexo. A mudança ocorre em dois pontos principais: o primeiro quando recebe um pacote de uma fonte qualquer, mas a máquina de estados da transmissão se encontra em FULL, então ao invés de descartar o evento, ele é armazenado; o segundo ponto é logo após o envio de uma célula, antes de retornar a um estado de espera deve-se observar o contador do buffer para procurar pacotes em espera e tratá-los.

Esta alteração não seguiu nenhum estudo de filas ou documentação de outros simuladores, e partiu de uma modificação que fosse simples e rápida de ser executada, com o intuito de verificar seus resultados. Quanto ao arquivo de configuração, acrescentou-se mais alguns parâmetros ao módulo AAL2, o tamanho máximo do buffer e o tempo de permanência do pacote no buffer, que são fixos pelo usuário e um outro contador para leitura da utilização do buffer.

Apesar do pouco tempo de teste, o resultado apresentado abaixo comprova o esperado, as diferenças em relação aos artigos reduzem drasticamente nas partes altas dos gráficos, onde o tráfego é mais intenso. Estudos futuros podem estudar um buffer mais adequado e realizar simulações mais específicas, a fim de obter eventos relacionados a esta implementação, e estudar a capacidade, método de trabalho e outros parâmetros necessários. Como uma primeira observação, pode-se destacar a tabela A-1 na qual restringiu-se a largura de banda disponível a fim de forçar o preenchimento do buffer.



Nova comparação do número de fontes obtidas pelo simulador



Nova comparação da densidade do pacote obtida pelo simulador

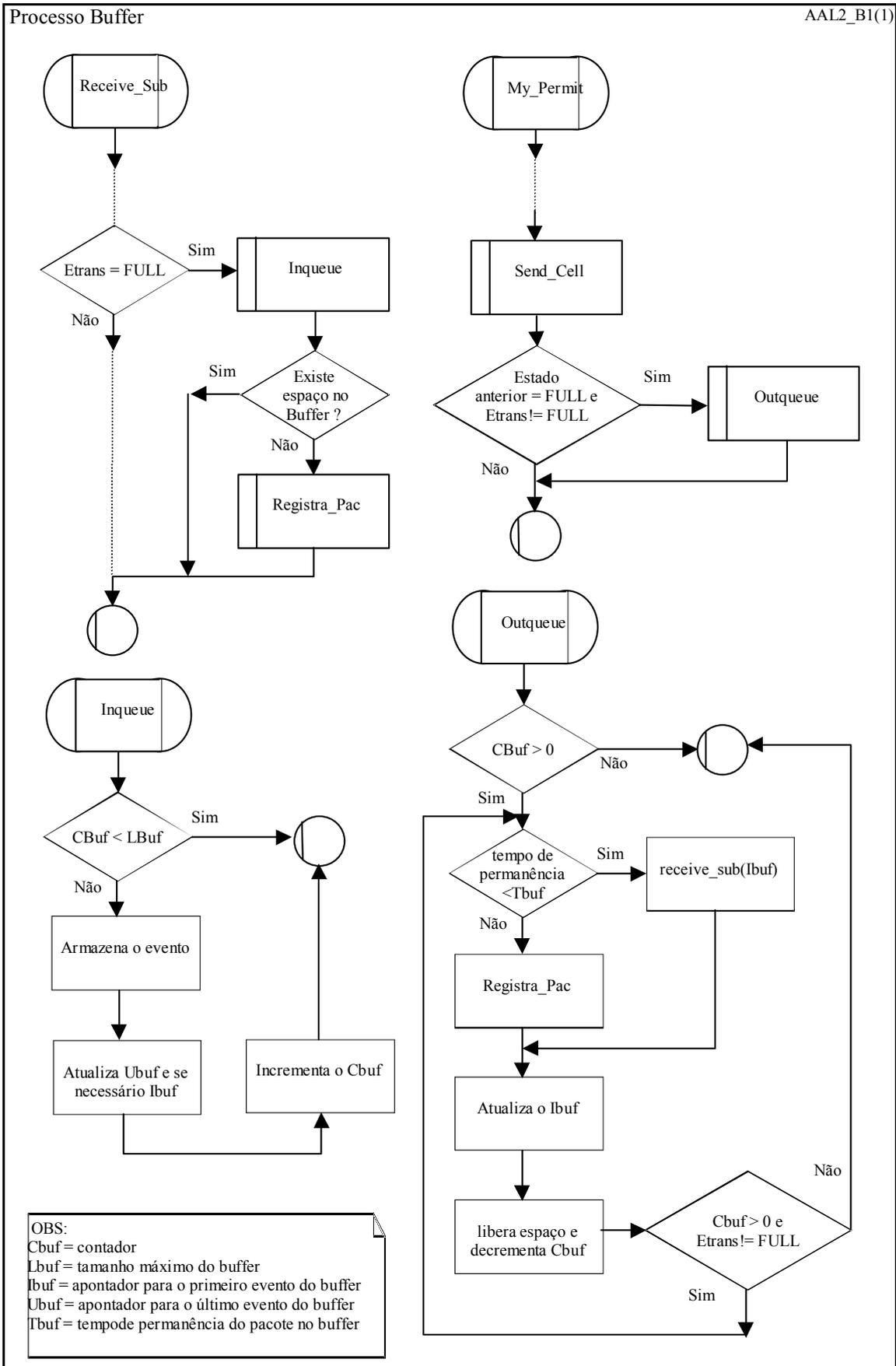
A tabela A-1 apresenta os resultados obtidos por Mcloughin (coluna cinza) da General DataComm. Mcloughin conforme já apresentado anteriormente. Na primeira parte da tabela, calculou-se qual a largura de banda seria utilizada pelo número de canais proposto por Mcloughin, e comparando-a ao valor esperado de 1536 Kbit/s. Na segunda parte da tabela, reduziu-se o número de fontes em 20%, o que aproximou os valores do desejado.

Como estamos tratando de tráfego VBR, quando dizemos que a utilização média é próxima de 1536 Kbit/s, significa que parte do tempo este valor é ultrapassado. Se, no entanto, realmente restringirmos a saída máxima de células a 1536 Kbit/s, a taxa de utilização cai um pouco devido a perda de pacotes que ultrapassarem esta taxa. Na última parte da tabela, calculamos um número de canais baseado nesta saída restrita e outra de 1843 Kbit/s (20% acima do DS1), mas consideramos uma perda máxima admissível de 5% de pacotes. Pode-se concluir que, para um determinado número de canais VBR, além da utilização média, é importante ter-se uma pequena largura de banda excedente e um *buffer* de entrada para reduzir as perdas por excesso de pacotes. Futuramente, pode-se realizar estudos para determinar uma relação entre o tamanho do *buffer*, o número de canais e a largura de banda necessária.

**Tabela A-1.** Utilização de um tronco DS1 com o protocolo AAL2

Timer_CU	Número de canais a 32Kbs	Largura de Banda em Kbit/s	Número de canais a 32Kbs (20% IDLE)	Largura de Banda em Kbit/s	Número de canais a 32 Kbit/s			
					Máx: 1536Kbit/s	Máx: 1843Kbit/s	Máx: 1536Kbit/s com buffer	Máx: 1843Kbit/s com buffer
1 ms	55	1207	44	1001	57	69	72	82
2 ms	70	1512	56	1196	57	69	72	82
3 ms	77	1645	62	1378	57	69	72	82
4 ms	82	1791	66	1434	57	69	72	82
5 ms	85	1865	68	1553	57	69	72	82
6 ms	85	1821	68	1481	57	69	72	82
7 ms	87	1905	70	1527	57	69	72	82
8 ms	87	1900	70	1553	57	69	72	82
9 ms	90	1929	72	1569	57	69	72	82
10 ms	90	1973	72	1516	57	69	72	82
11ms	90	1967	72	1496	57	69	72	82
12 ms	92	1944	74	1611	57	69	72	82
13 ms	92	1997	74	1589	57	69	72	82

E) Diagrama SDL das mudanças realizadas



## Anexo 5 Módulo de Inferência Estatística

Durante todo o processo de simulações realizadas na dissertação, os cálculos da média, variância e outras variáveis estatísticas eram feitos de forma quase manual ou com a utilização de outros programas. Por exemplo, o arquivo de log era filtrado para obtenção do parâmetro de interesse, depois era levado a uma planilha de cálculo, onde as fórmulas eram digitadas e os resultados finais usados para gerar os gráficos.

Foi sugerido como trabalho futuro a realização de um módulo de inferência estatística, mas através da troca de idéias com outros membros do LABCOM da UFRGS sobre a utilização do próprio simulador NIST para realizar este trabalho, desenvolveu-se um primeiro módulo de programação. Feito a princípio para gerar os cálculos básicos e expor os resultados de forma textual ao final da simulação, permite melhor aproveitamento do tempo para estudos das saídas. A geração de gráficos ainda é a encargo de outros programas, mas a partir deste módulo pode-se pensar em gerar saídas próprias e específicas para este fim.

Esta mudança no simulador seguiu a mesma tendência inicial de não alterar toda estrutura já existente desenvolvida pelo NIST. Isto só foi possível porque no byte utilizado para guardar os flags de cada parâmetro existia um bit sobrando, que foi utilizado para indicar um tratamento especial. Os demais flags dos componentes indicam se aquele parâmetro deve ser logado, se terá um medidor na parte gráfica, se será visível, etc. Este flag extra que foi criado, só é usado quando a variável em questão está sendo logada, então ocorre a verificação e a chamada ao novo módulo se necessário.

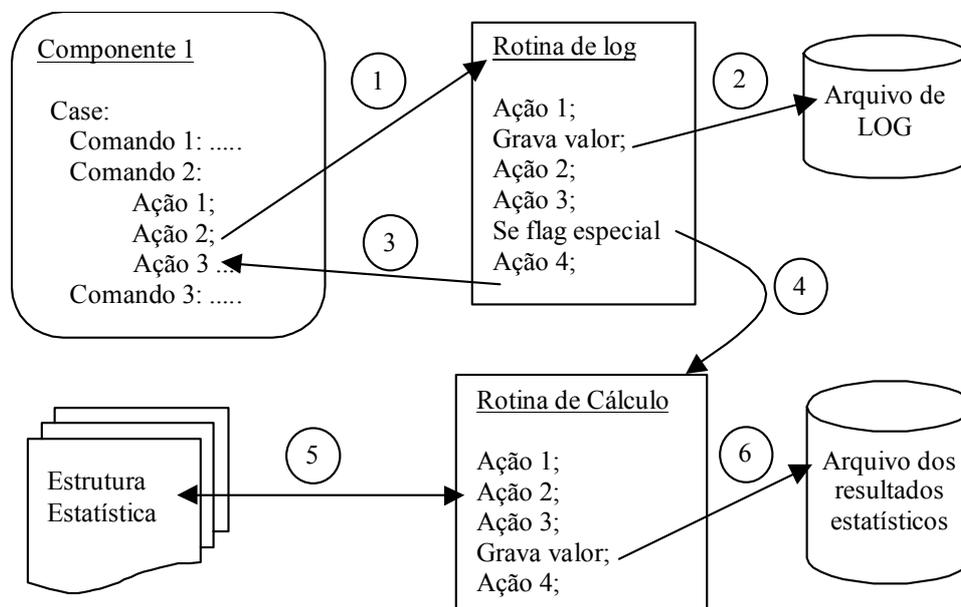


Diagrama representando o funcionamento do módulo estatístico

Para guardar todas as variáveis estatísticas, existe uma função de inicialização que armazena espaço para uma estrutura específica destinada a este fim. No programa principal do simulador existe a chamada para esta função e para uma outra que testa o fim da simulação, caso a precisão desejada tenha sido atingida. Para evitar alterações bruscas, o término ocorre pela alteração da variável de ambiente "stoptime" (a mesma

usada na linha de comando) que irá causar a parada do controlador de eventos após algum tempo.

Todo processo estatístico necessita de informações específicas que são lidas de um arquivo com a extensão ".stc" e o mesmo nome do arquivo de configuração. Caso este arquivo não exista, nada é realizado. Os resultados também ficam num arquivo separado cuja extensão é ".cal", de forma que não existe alteração no arquivo de log original. Nas informações necessárias estão : número de intervalos, precisão desejada do resultado, nível de confiança do resultado, tamanho do intervalo em microsegundos, tamanho do intervalo em número de amostras e o salto inicial em microsegundos para evitar os transitórios.

É utilizado o método de subintervalos descrito no item 6.5, no qual cada replicação é subdividida em intervalos nos quais são calculadas as médias. Durante o processo é reavaliado o intervalo de confiança para verificar se atingiu o nível de confiança especificado e a precisão desejada. O tamanho dos intervalos são definidos pelo tempo de duração ou número de amostras logadas (note que se a variável é logada a cada 10 ocorrências esse valor será diferente do número de amostras reais). Além do diagrama anterior pode-se entender a lógica do módulo através do diagrama SDL a seguir, e observar os exemplos dos arquivos ".stc" e ".cal".

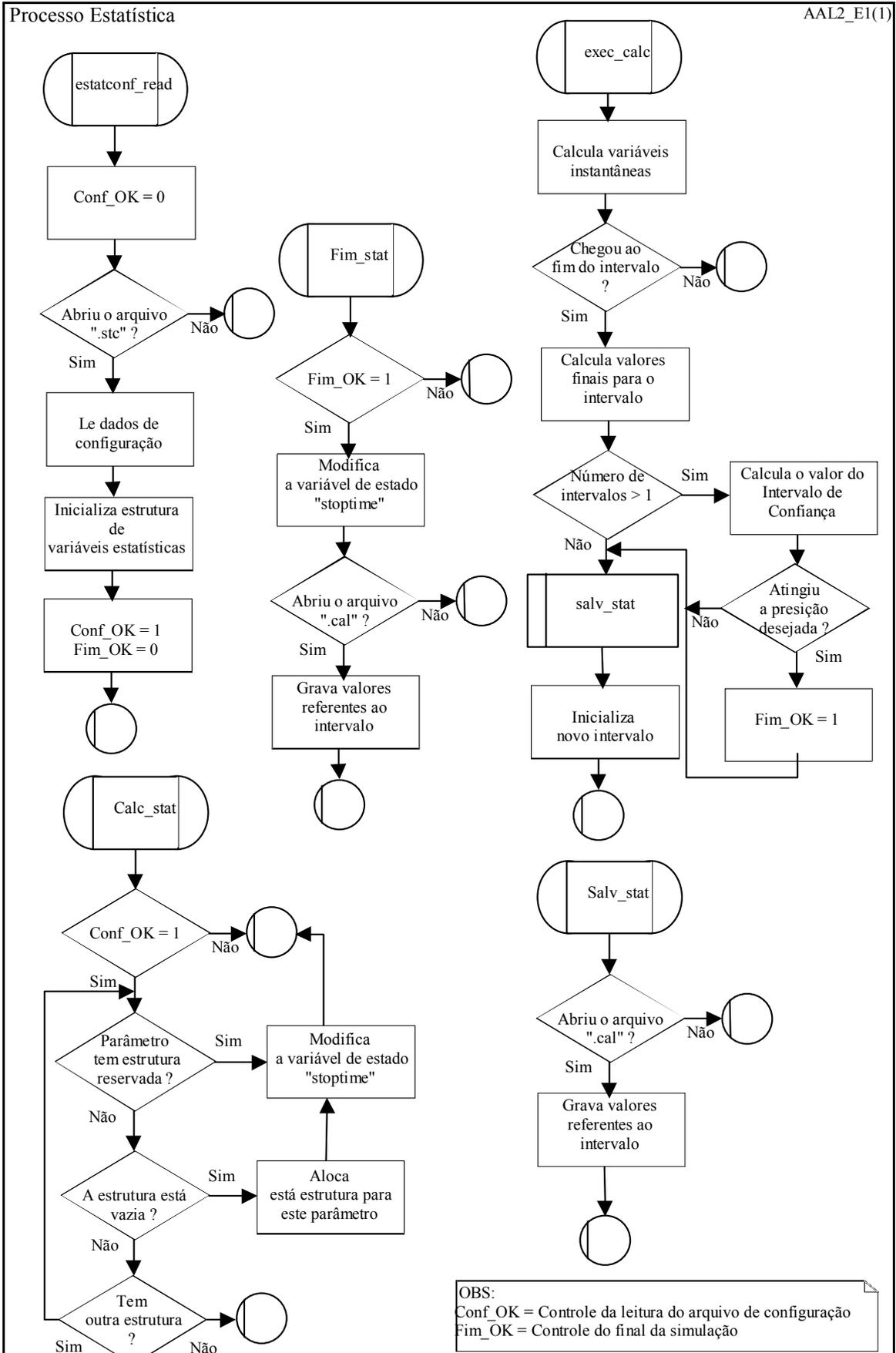
#### A) Arquivo de configuração estatística (".stc")

```
30 1.0 1.70 2000000 1000 2000000
30 1.0 1.70 2000000 1000 2000000
30 5.0 1.70 400000 100 400000
30 5.0 1.70 400000 100 400000
```

#### B) Arquivo de resultado estatístico (".cal")

```
# Inicio da Simulacao do arquivo ./di/conf_temp
Lido do Arquivo de Conf: 30 1 1.7 2e+06 1000 2e+06
Lido do Arquivo de Conf: 30 1 1.7 2e+06 1000 2e+06
Lido do Arquivo de Conf: 30 5 1.7 400000 100 400000
Lido do Arquivo de Conf: 30 5 1.7 400000 100 400000
# Componente Parametro Intervalo Amostra Tempo Media Int_Confianca Presicao
'teste1' '% do total da célula' 0 1000 1.8833e+08 48.1886 48.1886 0 1
'teste1' '% do total da célula' 1 1000 1.54685e+08 47.0566 47.6226 0.962265 1
'teste1' '% do total da célula' 2 1000 1.76126e+08 49.9245 48.3899 1.41779 1
'teste1' '% do total da célula' 3 1000 1.82681e+08 51.6981 49.2169 1.7268 1
'teste1' '% do total da célula' 4 1000 1.72309e+08 48.9811 49.1698 1.33997 1
'teste1' '% do total da célula' 5 1000 1.78445e+08 48.7547 49.1006 1.10039 1
'teste1' '% do total da célula' 6 1000 1.8679e+08 50.2641 49.2668 0.971977 1
'teste1' '% do total da célula' 7 1000 1.72478e+08 50.3773 49.4056 0.87421 1
'teste1' '% do total da célula' 8 1000 1.65948e+08 47.8113 49.2285 0.827711 1
'teste1' '% do total da célula' 9 1000 1.78634e+08 45.5094 48.8566 0.973559 1
'teste1' '% do total da célula' 10 1000 1.66751e+08 47.9622 48.7753 0.891398 1
'teste1' '% do total da célula' 11 1000 1.768e+08 47.5849 48.6761 0.831022 1
'teste1' '% do total da célula' 12 1000 1.84681e+08 48.8679 48.6908 0.764841 1
'teste1' '% do total da célula' 13 1000 1.70009e+08 48.2641 48.6603 0.709998 1
'teste1' '% do total da célula' 14 1000 1.59598e+08 47.5094 48.5836 0.67372 1
```

C) Diagrama SDL do módulo estatístico



## *Bibliografia*

- [ATM 96] ATM Forum/95-0013R10. **ATM Traffic Management Specification**. v.4, 1996. Disponível em : <<ftp://ftp.atmforum.com/pub/approved-specs/af-tm-0056.000.os>>. Acesso em: fev. 1999.
- [ATM 99] ATM Forum. **ATM Trunking using AAL2 for Narrowband Services**. 1999. Disponível em : <<ftp://ftp.atmforum.com/pub/approved-specs/af-vtoa-0113.000.pdf>>. Acesso em: fev. 1999.
- [BAN 95] BANKS, Jerry; CARSON, John S.; NELSON, Barry L. **Discrete-Event System Simulation**. 2nd ed. New Jersey: Prentice Hall, 1995. 548p.
- [BIH 99] BIHRE, Marcelo. **Uma Biblioteca para Simulação e Avaliação de Modelos de Filas em Redes de Comunicação de Dados**. Porto Alegre: Instituto de Informática da UFRGS, 1999.
- [CER 97] CEREDA, Ronaldo L. Dias et al. **ATM O Futuro das Redes**. São Paulo: Makron Books do Brasil, 1997. 178p.
- [CYC 97] CYCLADES CORPORATION. **Guia Internet de Conectividade**. 4th ed. São Paulo: Cyclades do Brasil, 1997. 162p.
- [DOR 96] DORLING, Brian et al. **Internetworking Over ATM: An Introduction**. New Jersey: Prentice Hall, 1996. 206p.
- [FRO 94] FROST, Victor S.; MELAMED, Benjamin. Traffic Modeling for Telecommunications Networks. **IEEE Communications Magazine**, New York, v.32, n.3, p.70-81, Mar. 1994.
- [GDC 2000] GENERAL DATACOMM. **TDM to ATM Migration by GDC**. Disponível em : <<http://www.gdc.com/home/atm/preface.html>>. Acesso em: 10 nov. 2000.
- [GOL 98] GOLMIE, Nada et al. **The NIST ATM/HFC Network Simulator : Operation and Programming Guide Version 4.1** Disponível em : <<http://isdn.ncsl.nist.gov>>. Acesso em: 1998.
- [HES 98] HESSELBACH, X. On the Quality of Service Enhancement in Variable Bit Rate Networks. **The IEEE Computer Society's Students Newsletter**, Barcelona, v.6, n.1, Spring 1998. Disponível em : <<http://computer.org/students/looking/spring1998/index.html>>. Acesso em: 1999.
- [HEU 90] HEUSER, Carlos A. **Modelagem Conceitual de Sistemas: Redes de Petri**. Nova Friburgo: EBAI, 1990. 150p.
- [ITU 93] ITU-T. **B-ISDN ATM adaptation layer (AAL) specification**, Recommendation I.363. [S.l.], 1993.

- [ITU 96a] ITU-T. **One-Way Transmission Time**, Recommendation G.114. [S.I.], 1996.
- [ITU 96b] ITU-T. **Traffic Control and Congestion Control in B-ISDN**, Recommendation I.371. [S.I.], 1996.
- [ITU 97] ITU-T. **B-ISDN ATM Adaptation layer specification : Type 2 AAL**, Recommendation I.363.2 . [S.I.], 1997.
- [ITU 98] ITU-T. **Segmentation and Reassembly Service Specific Convergence Sublayer for the AAL type 2**, Recommendation I.366.1. [S.I.], 1998.
- [ITU 99a] ITU-T. **B-ISDN ATM layer specification**, Recommendation I.361. [S.I.], 1999.
- [ITU 99b] ITU-T. **AAL Type 2 SSCS for trunking**, Recommendation I.366.2. [S.I.], 1999.
- [JAI 91] JAIN, R. **The Art of Computer Systems Performance Analysis, technique for Experimental Design, Measurement, Simulation and Modelaling**. Massachusetts: John Wiley & Sons, 1991.
- [JAI 96] JAIN, R. **Source Behavior for ATM ABR Traffic Management: Na Explanation**. 1996. Disponível em : <<http://www.cis.ohio.state.edu/~jain/papers.html>>. Acesso em: jul. 2000.
- [JAI 99] JAIN, Raj et al. **Packing Density of Voice Trunking using AAL2**. Disponível em: <<http://www.cis.ohio-state.edu/~jain/papers/density.htm>>. Acesso em: jul. 2000.
- [JIR 2000] JIRON, Diana Maria Blandon. **Simulação de Videoconferência Através de Técnicas de Simulação por Eventos Discretos**. Porto Alegre: Instituto de Informática da UFRGS, 2000.
- [MCL 97a] MCLOUGHLIN, Mike; O'NEIL, John. **A Management Briefing on Adapting Voice for ATM Networks, Na AAL2 Tutorial**. Connecticut: General DataComm, 1997. Disponível em : <<http://www.gdc.com/inotes/pdf/aal2tut.pdf>>. Acesso em: out. 1999.
- [MCL 97b] MCLOUGHLIN, Mike; MUMFORD, Keith. **A Management Briefing on Adapting Voice for ATM Networks, A Comparison of AAL1 Versus AAL2**. Connecticut: General DataComm, 1997. Disponível em : <<http://www.gdc.com/inotes/pdf/aal1vs2.pdf>>. Acesso em: out. 1999.
- [NAK 98] NAKAMURA, Hiroshi et al. Applying ATM to Mobile Infrastructure Networks. **IEEE Communications Magazine**, New York, v.38, n.1, p.66-73, Jan. 1998. Disponível em : <<http://www.comsoc.org/ci/private/1998/jan/ Nakamura.html>>. Acesso em : 2000.

- [NOG 98] NOGUEIRA, Juan. ATM Adaptation Layer 2. **The IEEE Computer Society's Students Newsletter**, Barcelona, v. 6, n. 1, Spring 1998. Disponível em : <<http://computer.org/students/looking/spring1998/index.html>>. Acesso em : 1999.
- [ONV 95] ONVURAL, Raif O. **Asynchronous Transfer Mode Networks, Performance Issues**. 2nd ed. Boston: Artech House, 1995. 536p.
- [PIT 96] SCHORMANS, Pitts. **Introdução to Assynchronous Transfer Mode, Design and Performance**. New York: Wiley, 1996.
- [PRY 95] PRYCKER, Martin de. **Asynchronous Transfer Mode : Solution for Broadband ISDN**. 3rd ed. London: Prentice Hall, 1995.
- [ROC 98] ROCHOL, Juergen. **Redes de Computadores**. Porto Alegre: Instituto de Informática UFRGS, 1998. Apostila .
- [SIL 2000] SILVA, Fernanda Mendes Ascensão. **Multiplexação Estatística de Fontes Heterogêneas em ATM usando o modelo On/Off Pareto**. Porto Alegre: Instituto de Informática da UFRGS, 2000. Dissertação de mestrado.
- [SOR 90] SORAES, Luiz Fernando G. **Modelagem e Simulação Discreta de Sistemas**. São Paulo: IME/USP, 1990. 254p. Trabalho apresentado na Escola de Computação, 7., 1990, São Paulo.
- [STA 96] STALLINGS, Willian. **Data and Computer Communications**. 5th ed. Upper Saddle River: Prentice Hall, 1996.
- [TAN 96] TANENBAUM, A. S. **Computer Networks**. 3rd ed. Upper Saddle River: Prentice Hall PTR, 1996.