Serviços de Transferência de Dados Ponto a Multiponto

Marcos Tadeu von Lutzow Vidal [1]
Universidade Federal Fluminense

Fernando Mascarenhas Cavalcanti de Barros
Companhia Vale do Rio Doce

Otto Carlos Muniz Bandeira Duarte [2]
Universidade Federal do Rio de Janeiro

Universidade Federal do Rio de Janeiro
COPPE - Programa de Engenharia Elétrica
P.O. Box 68504 CEP 21945 - Rio de Janeiro - RJ - Brasil
FAX: +55 21 290.6626 Email: otto@coe.ufrj.br

Resumo

Uma parcela considerável do tráfego das novas redes de comunicação será constituída de aplicações multi-mídia distribuidas que se servirão de comunicações ponto a multiponto. O objetivo deste trabalho é a apresentação de três serviços de transferência de dados ponto a multiponto e a análise dos resultados de desempenho obtidos com os protocolos implementados. Dois ambientes de comunicação diferentes foram considerados: um canal de satélite simulado em um computador pessoal compatível IBM-PC inclusive com geração de erros e uma rede local do tipo Ethernet. As implementações efetuadas seguem um modelo de arquitetura em camadas e oferecem um serviço de Enlace ponto a multiponto usando os serviços oferecidos pela camada Física do satélite (simulação) ou da camada MAC de uma rede local. Os resultados de desempenho para o protocolo mais complexo indicam uma vazão de até 3.2 Mbps para o ambiente de satélite simulado em um computador pessoal compatível IBM-PC (80386 DX-33) e de 1.3 Mbps para o ambiente de rede local com computadores pessoais compatíveis IBM-PC (80286 8 MHz).

Abstract

A significant part of the ATM network traffic will probably due to the distributed multimidia applications supported by one to many communication type. This work aims at presenting three point-to-multipoint services and analysing the performance results obtained by the protocol implementations. Two different environments was considered: one simulates a satellite channel with errors characteristics and the other is an Ethernet LAN. The implementations consider a layered model and offer point-to-multipoint Data Link Services using the services provided by a satellite Physical layer (simulated) or by the MAC layer of a LAN. The performance results of the most complex protocol show a throughput up to 3.2 Mbps for the simulated satellite environment (IBM-PC 80386-DX 33 MHz) and 1.3 Mbps for the Ethernet LAN interconnecting personal computers IBM-PC (80286 8 MHz).

[1] atualmente em tese de M. Sc. na COPPE/UFRJ Email: marcos@coe.ufrj.br

[2] atualmente em pós-doutorado no Laboratório MASI
Laboratoire MASI - Equipe Réseau et Performance
Université Pierre et Marie Curie
4 Place Jussieu - 75252 Paris Cedex 05 - France
Fax: +33 1 44.27.62.86 Email: duarte@masi.ibp.fr

1 Introdução

Uma comunicação pode ser classificada em: ponto a ponto, multiponto a ponto, ponto a multiponto e conferência. A grande maioria das aplicações existentes são baseadas em comunicações ponto a ponto através da rede telefônica ou da Rede Nacional de Comutação de Pacotes (RENPAC). As aplicações que se servem dos outros tipos de comunicação são relativamente recentes, uma vez que a necessidade de um canal de difusão limitava seu uso aos ambientes de redes locais e de redes via satélite. No entanto, este cenário deverá se modificar nos próximos anos, uma vez que as comunicações ponto a multiponto também serão possíveis nas futuras Redes Digitais de Serviços Integrados em Banda Larga baseada no Modo de Transferência Assíncrono. Esta próxima geração de redes de computadores se beneficiará dos avanços tecnológicos obtidos nas últimas décadas e oferecerá velocidades de transmissão da ordem de centenas de Mbps. Esta larga banda passante associada ao modo de transferência empregado permitirá aplicações do tipo multi-mídia que integram voz, dados e vídeo. Assim sendo, uma série de novas aplicações poderão ser oferecidas, como por exemplo: correio eletrônico multi-mídia, tele-conferência digital, edição cooperativa de textos multi-mídia, controle cooperativo de experiências remotas, projeto distribuido, etc.

O Grupo de Teleinformática e Automação (GTA) da UFRJ realiza pesquisas relacionadas a estes temas e desenvolve atualmente um ambiente de comunicação ponto a multiponto. Neste trabalho são apresentados três serviços de transferência de dados ponto a multiponto e a análise dos resultados de desempenho obtidos com os protocolos implementados. Estes serviços foram originalmente concebidos para utilização em um canal de difusão de satélite e adaptados para sua utilização em rede local (LAN).

Uma primeira versão dos protocolos descritos neste artigo foi implementada [1] em um computador pessoal compatível IBM-PC, onde o transmissor, os receptores e o canal de satélite (com possibilidades de geração de erros) são implementados em uma única máquina. Atualmente, uma segunda versão destes protocolos considera também o ambiente de rede local onde o transmissor e os receptores são computadores PCs independentes interconectados por uma rede Ethernet. Estas implementações seguem uma arquitetura particular que privilegia o alto desempenho e servirão de teste para as aplicações multi-mídias. Uma nova versão destes protocolos deverá rodar em máquinas SUN com sistema operacional UNIX.

Este trabalho descreve em linguagem natural os serviços e protocolos ponto a multiponto. Procurou-se usar o vocabulário e os níveis de abstração (arquitetura em camadas, definição de serviços e especificação dos protocolos) definidos no Modelo de Referência de Interconexão de Sistemas Abertos [2]. Em seguida mostra-se as diferenças entre as implementações para o ambiente de satélite simulado e de redes locais. A arquitetura de implementação usada é descrita de uma forma sucinta. Finalmente, as medidas de desempenho são apresentadas e discutidas.

2 Serviços Ponto a Multiponto

Os três serviços de transferência unilateral ponto a multiponto de informação que são objeto deste trabalho foram denominados: difusão simples, difusão com rotação e ponto a multiponto com canal de retorno.

2.1 Difusão Simples

Neste serviço a comunicação se dá em um único sentido, isto é, uma estação transmite e as demais só recebem. É um serviço extremamente simples sendo apropriado para a utilização do serviço DATASAT-UNI mas também pode ser usado em redes locais. Devido a sua característica unilateral não é possível uma recuperação de erros por pedido de retransmissão; assim como, pelo mesmo motivo, não é possível efetuar um controle de fluxo que exija informações enviadas pelos receptores. Assim sendo, os únicos métodos de recuperação de erros plausíveis de serem usados são os de correção automática de erro; onde, para garantir a integridade dos dados, acrescentam-se redundâncias e códigos corretores de erro na mensagem. Contudo, uma vez que esses métodos são complexos e de eficiência não muito alta (1/2, 3/4, 7/8, ...); muitas vezes sua utilização torna-se inviável.

Este serviço garante a integridade dos dados transferidos, utilizando-se unicamente do mecanismo de numeração dos quadros. Não é possível a recuperação de um erro, mas a quebra da seqüência indicará um erro ao usuário. Como a detecção de erros é feita por seqüenciamento, uma seqüência de quadros errados no fim de uma transmissão não seria possível de ser detectada. Portanto, há necessidade de uma mensagem marcando o fim da transmissão.

Quanto ao controle de fluxo existirá um acordo *a priori*, onde a cadência de transmissão e os recursos necessários estarão definidos e disponíveis no momento da comunicação. Um possível esquema para controlar a cadência de transmissão pode ser implementado da seguinte forma: o transmissor difunde no máximo N mensagens seguidas, dispara: um temporizador T e só reinicia a transmissão com o término desse temporizador. Na verdade o mecanismo descrito "frea" a transmissão diminuindo a sua velocidade ao introduzir "silêncios" a cada bloco de mensagens transmitidas. Dessa forma, estipulando-se os valores de N e T de acordo com a capacidade de processamento e armazenamento dos receptores, pode-se garantir que não haverá sobre-escrita nos *buffers* de recepção.

Sendo este serviço bastante simples, e não orientado à conexão, as únicas primitivas existentes são as de dados, e uma primitiva, usada nas estações receptoras, que avisa ao usuário se chegar um quadro fora de seqüência. A Tabela 1 mostra as primitivas e seus parâmetros.

A única PDU (Unidade de Dados de Protocolo) deste serviço é a de dados e tem o formato mostrado na Figura 1.

Tabela 1: Primitivas do Serviço de Difusão Simples.

Serviço	**Primitiva**	**Parâmetros**
Transferência de Dados	MDL_DATA.Request	Dados do Usuário
	MDL_DATA.Indication	Dados do Usuário
Erro	MDL_ERROR.Indication	-

MI	Delim.	End.Fonte	End.Global	Cód.	Ns	Dados	CRC	Delim
	1	1	1	1	4	N	2	1

$num.Octetos$

Figura 1: Codificação da PDU de informação

2.2 Difusão com Rotação de Permissão

Este serviço também possui característica unidirecional, isto é, a cada instante somente uma estação transmite e as demais somente recebem a transmissão da primeira. Contudo, a estação transmissora não é mais sempre a mesma, como no caso anterior. Neste serviço cada estação detém a permissão para transmitir durante um intervalo de tempo e há, portanto, uma rotação entre todas as estações. Todavia, deve ficar claro que quando uma estação transmite, as demais só podem receber. Dos diversos mecanismos possíveis para realizar a rotação da permissão de transmissão, o escolhido é baseado na multiplexação síncrona no tempo, devido a sua relativa simplicidade de implementação. Neste mecanismo, um período de tempo (por exemplo: 60 minutos) é dividido em intervalos fixos para cada estação (por exemplo: com três estações, 18 minutos para cada), existindo ainda um período de resguardo (no exemplo, 2 minutos) entre o fim da permissão de uma estação e o início de outra. Neste serviço o controle da permissão de transmissão é distribuido. Sendo assim, cada estação tem seu próprio relógio de tempo real e "sabe" o período correto em que pode transmitir. Estes períodos têm que ser estipulados previamente e colocados em uma tabela com os horários de permissão para cada estação, sendo que cada estação é possuidora de uma cópia desta tabela. O *gap* entre duas permissões serve para sanar eventuais diferenças entre os relógios (de tempo real) de cada estação, e sua duração deve ser estipulada de acordo com a precisão dos relógios.

Para oferecer este serviço, é necessário um meio de transmissão que seja capaz de difundir uma mensagem de uma estação qualquer para todas as outras. Uma vez garantida esta premissa, esse serviço provê a transferência transparente de dados entre os diversos usuários do serviço, liberando-os da obrigatoriedade de possuir conhecimento sobre o mecanismo de transmissão. Além disto, a transferência se caracteriza como transparente também pelo fato de não haver nenhuma ressalva quanto ao conteúdo das mensagens do usuário, que são integralmente preservadas.

Neste serviço já é possível implementar um esquema de recuperação de erros, uma vez que cada estação pode utilizar parte de seu tempo de transmissão para pedir retransmissão de quadros não recebidos (descartados por estarem com erro), para retransmitir quadros que tenham sido pedidos e, finalmente, para transmitir novas mensagens. A estratégia utilizada consiste em numerar os quadros de maneira absoluta (devido a simplicidade), identificando cada quadro unicamente. Sendo assim, qualquer quebra na seqüência de quadros recebidos indicará que houve erro. Enquanto transmissora, uma estação deve guardar todos os quadros transmitidos por ela. Somente quando receber confirmação de todas as demais estações para um determinado quadro, é que este pode ser descartado. Quando estiver atuando como receptora, uma estação deve fazer uma lista de quadros não recebidos (que foram detectados pela quebra de seqüenciamento) para poder enviá-la quando estiver na vez de transmitir. Ao receber uma lista deste tipo, deve-se verificar se algum dos quadros presentes na lista está armazenado, a espera de reconhecimento, e, em caso positivo, este quadro deve ser retransmitido, quando a estação tornar-se transmissora novamente. Se, ao tornar-se transmissora, uma estação não tiver nenhum pedido de retransmissão a fazer (não houve nenhum erro detectado), ela deve transmitir um reconhecimento positivo com o número do próximo quadro esperado, para indicar as demais estações que todos os quadros anteriores foram recebidos com sucesso e, no que lhe diz respeito, eles podem ser descartados.

Inserções e retiradas de estações do "ciclo" não estão previstas, de modo que a queda numa estação pode causar problemas de falta de memória nas demais, visto que deixarão de receber reconhecimento positivo da estação que saiu do ar e, deixarão de descartar os quadros transmitidos. Isto pode ser resolvido através do emprego de um contador de número de fatias de tempo que uma estação ficou em silêncio. Ao ultrapassar um valor limite previamente estipulado, a estação é dada como fora do ar e, todos os quadros que só dependiam do reconhecimento desta estação podem ser descartados.

Não haverá controle de fluxo explícito, a princípio. A cadência de transmissão e os recursos necessários serão definidos *a priori*, e estarão disponíveis no momento da comunicação. Mesmo assim, em caso de perda de quadros devido ao impedimento no tratamento, estes serão recuperados pelos mecanismos de recuperação de erros.

As primitivas e seus parâmetros são mostradas na Tabela 2. Este serviço também não é orientado a conexão, porém existem primitivas que que realizam funções de gerência da comunicação. Estas primitivas avisam ao usuário (ou a uma camada de gerência) de que ele pode transmitir (PODE_TRANSMITIR.Indication) ou não (NAO_PODE_TRANSMITIR.Indication), e uma outra avisa a todos os usuários quem vai transmitir, ou seja, de quem é a vez a partir do momento da chegada da primitiva (VEZ.Indication).

Tabela 2: Primitivas do Serviço Rotação de Permissão

Serviço	**Primitiva**	**Parâmetros**
Transferência de Dados	MDL_DATA.Request	Dados do Usuário
	MDL_DATA.Indication	Dados do Usuário
Gerência da comunicação	PODE_TRANSMITIR.Indication	-
	NAO_PODE_TRANSMITIR.Indication	-
	VEZ.Indication	-

A PDU de dados (informação) deste serviço é idêntica ao do serviço anterior (Figura 1). Além desta, existem mais duas PDUs, (Figura 2) que são utilizadas para enviar os números dos quadros com reconhecimento positivo (ACK) e reconhecimento negativo (NACK). Sendo a numeração de quadros independente para cada estação, cada quadro é designado por seu número, seguido do endereço (número) da estação que o transmitiu.

ACK

Delim.	End.Fonte	End.Global	Cód.ACK	Ns	End.Est.	...	Ns	End.Est.	CRC	Delim
1	1	1	1	4	1		4	1	2	1

num.Octetos

NACK

Delim	End.Fonte	End.Global	Cód.NACK	Ns	End.Est.	...	Ns	End.Est.	CRC	Delim
1	1	1	1	4	1		4	1	2	1

num.Octetos

Figura 2: Codificação de PDU - ACK e NACK

2.3 Ponto a Multiponto com Canal de Retorno

Assim como nos serviços precedentes, este serviço oferece uma transferência unilateral de informação de um ponto para muitos. A grande diferença está na possibilidade de se poder contar com um canal de retorno que permita transferir dados de controle no sentido dos receptores para o transmissor. O uso de uma rede local como provedor do meio de difusão é relativamente trivial e em rede de satélite é necessário a alocação do canal de difusão (sentido transmissor/receptores) e R canais de retorno no sentido receptor para emissor. As R bandas passantes alocadas como canais de retorno serão usadas para as mensagens de controle (basicamente reconhecimentos positivos, negativos e pedidos de desconexão).

O serviço permite: estabelecer uma conexão; transferir, de modo transparente, mensagens (Unidades de Dados de Serviço - SDUs) durante a conexão; liberar a conexão de modo incondicional, por parte do transmissor e liberar a conexão de modo negociado, por parte de um dos receptores.

O estabelecimento de uma conexão é iniciado pelo emissor que difunde um pedido de conexão e aguarda a aceitação (eventualmente uma recusa) por parte de cada um dos receptores. O serviço oferecido só considera a conexão como estabelecida se receber confirmação positiva de todos os receptores. Para sanar problemas com possíveis perdas do pedido de conexão ou da aceitação, um mecanismo baseado em um contador de tempo reenvia o pedido de conexão um certo número de vezes, caso não chegue resposta de todos os receptores, dentro de um período previamente estabelecido. É importante observar que ao se difundir um novo pedido de conexão após a temporização, os receptores que tivessem recebido corretamente o pedido de conexão e aberto esta conexão poderiam interpretar este reenvio como um pedido de nova conexão. Afim de resolver este problema e evitar algoritmos mais complexos, nesta primeira versão não está previsto multiplexagem e portanto, uma única conexão ponto à multiponto é possível. Ainda assim, existe o problema de um (ou mais) receptores terem recebido corretamente o pedido de conexão, tratado este pedido, aberto a conexão, enviado a resposta de aceitação de abertura de conexão ao transmissor e esta resposta por alguma falha do provedor do serviço utilizado (ex, perda desta Unidade de Dados de Protocolo - PDU) não chegou ao transmissor. Como no serviço proposto o transmissor emprega um endereço coletivo e os receptores não tem como saber se as suas respostas foram recebidas corretamente, estes últimos ao receberem um novo pedido de conexão fecham a conexão aberta e reiniciam o procedimento de abertura. O transmissor por sua vez, não considera as respostas de aceitação de conexão já recebidas anteriormente.

A transferência de informação se dá apenas no sentido emissor para receptores (unidirecional), sendo que o serviço garante a integridade e a seqüência dos dados transferidos. As transmissões no sentido receptor para emissor são apenas de dados de controle do serviço, a nível de provedor de serviço, sendo que o usuário não toma conhecimento destas transmissões.

O serviço de liberação de conexão é do tipo abrupto com possíveis perdas de informação, caso a iniciativa de liberação seja do usuário emissor ou do provedor do serviço. Os usuários receptores podem pedir a liberação, pedido este que será enviado ao usuário emissor, cabendo a este decidir se haverá ou não liberação da conexão. A decisão de ruptura da conexão pode ser tomada: pelo usuário emissor, por vontade própria ou atendendo a um pedido de liberação de um ou mais receptores; pela máquina de estados do emissor devido a: recusa da conexão ou não resposta por um ou mais receptores; perda de algum receptor (detectado pelo silêncio deste no canal de retorno); erro permanente (não reconhecimento positivo de um quadro após várias repetições).

O protocolo de comunicação utilizado é baseado no HDLC (*High-Level Data Link Control*[3]), com modificações para suportar vários usuários e uma importante modificação no mecanismo de controle de erros. Esta modificação consiste no aperfeiçoamento do mecanismo SREJ do HDLC, adaptado para ambientes de alta velocidade, longo tempo de propagação e altas taxas de erros. O quadro SREJ N(R) ao mesmo tempo que pede a retransmissão do quadro N(R) reconhece positivamente todos os quadros com número de seqüência inferior a N(R). Isto impede o envio de um outro SREJ enquanto existir uma "condição de exceção SREJ", ou seja, enquanto o quadro N(R) pedido em retransmissão não for recebido corretamente. Isto é extremamente prejudicial e se constitui em um dos principais fatores de limitação de desempenho dos protocolos tipo HDLC, principalmente em ambientes onde o produto da taxa de transmissão pelo tempo de propagação (caso de redes de alta velocidade e/ou de satélite) é alto. Para sanar este inconveniente, pode-se eliminar a característica de reconhecimento positivo do quadro SREJ N(R) [4, 5] onde este último assume unicamente o comportamento de um reconhecimento negativo individual (como um NACK). Desta forma vários SREJs poderão ser enviados sem esperar a recuperação correta de um deles. Assim sendo, um SREJ será enviado pelos receptores sempre que for detectado quebra na seqüência dos quadros que estão recebendo. O emissor retransmite um quadro para cada SREJ que chegar, mesmo que cheguem dois ou mais com o mesmo número de quadro. Em termos de eficiência do canal de comunicação esta estratégia não apresenta um bom desempenho em ambientes com alta taxa de erros. Em uma comunicação com R receptores, se ocorre erro na mensagem difundida (no caso do satélite um erro no canal de subida) isto provocará R pedidos de retransmissões e consequentemente R retransmissões. Estes tipos de erros não são muito frequentes, pois em um ambiente de satélite o canal de subida tem mais potência (menor probabilidade de erros) e em uma rede local os erros são raros. Em uma segunda versão deste protocolo está previsto um algoritmo baseado em temporização que desconsidera pedidos de retransmissão de uma mesma mensagem durante um determinado período de tempo. Este algoritmo exige a criação de temporizadores tanto no transmissor quanto nos receptores, sendo sua eficácia dependente dos diferentes tempos de transferência entre o emissor e os diversos receptores.

O reconhecimento positivo fica a cargo de PDUs RR N(R) que reconhecem positivamente todos os quadros com número de seqüência menores ou iguais a $N(R)-1$. Estas PDUs devem ser enviadas com uma determinada freqüência, de modo a não deixar muitos quadros na memória do emissor (fila de possíveis quadros a serem retransmitidos) a espera de reconhecimento positivo, e também quebrar o silêncio no canal de retorno (uma das causas da liberação da conexão).

As mensagens difundidas deverão ser reconhecidas por todos os receptores. Um grande número de receptores implica em um grande número de reconhecimentos. Portanto, deve-se evitar um reconhecimento positivo por mensagem corretamente recebida pois isto sobrecarregaria o transmissor com o processamento destes reconhecimentos positivos. Para solucionar este problema duas abordagens foram estudadas: reconhecimento por grupo e temporizado. O reconhecimento por grupo é um mecanismo que enviaria um reconhecimento positivo a cada G mensagens corretamente recebidas e o reconhecimento temporizado implica no envio de reconhecimento positivo a cada "estouro" de um temporizador (T_{ack}) criado para esta finalidade. Evidentemente, o desempenho do protocolo e os recursos de memória necessários serão funções dos parâmetros G e T_{ack} que deverão ser escolhidos considerando as características do sistema e da aplicação. É importante observar que o procedimento de reconhecimento grupado não elimina a necessidade de um temporizador afim de se resguardar de uma parada de transmissão temporária onde o transmissor não teria nada mais a enviar e os receptores não teriam recebido G quadros para formar um reconhecimento. Nesta primeira versão optou-se pelo mecanismo de temporização.

O mecanismo de temporização é exclusivo aos reconhecimentos positivos. O reconhecimento negativo é enviado imediatamente afim de se recuperar um erro o mais rápido possível.

Além disto, está previsto um mecanismo, baseado em contadores de tempo, que retransmite todos os quadros a espera de reconhecimento positivo, caso haja um silêncio por parte dos receptores (não chegar nem RR nem SREJ durante um período de tempo pré-determinado). A cada chegada de reconhecimento por parte de todos os receptores para um determinado quadro (o que permite seu descarte), ou a chegada de um SREJ causa o re-disparo deste contador.

Neste serviço está implementado um controle de fluxo baseado em janela de emissão. O tamanho desta janela é determinado *a priori*, de acordo com os recursos necessários e os disponíveis no momento da comunicação. Este mecanismo verifica, no momento da transmissão, quantos quadros estão esperando por reconhecimento positivo: caso este número seja maior que o estabelecido, a transmissão é bloqueada até que a situação se reverta.

Para se evitar ambiguidades as PDUs serão numeradas segundo uma numeração absoluta, que na prática significa trabalhar com um modulo cíclico que não se esgota durante uma sessão de trabalho. Esta solução permitirá a simplificação dos algoritmos de recepção e transmissão.

Este serviço é orientado a conexão e, portanto, pode-se dividir o período de comunicação em três fases distintas: estabelecimento da conexão, transferência de dados e liberação da conexão. As primitivas associadas a cada fase e seus parâmetros aparecem na Tabela 3.

Tabela 3: Primitivas do Serviço Ponto-a-Multiponto com canal de retorno

Serviço	**Primitiva**	**Parâmetros**
Estabelecimento de conexão	MDL_ESTABLISH.Request	-
	MDL_ESTABLISH.Indication	-
	MDL_ESTABLISH.Confirm	-
Transferência de Dados	MDL_DATA.Request	Dados do Usuário
	MDL_DATA.Indication	Dados do Usuário
Liberação de conexão	MDL_ABORT.Request	-
	MDL_ABORT.Indication	-
	MDL_ABORT.Confirm	-
	MDL_RELEASE.Request	-
	MDL_RELEASE.Indication	-

Na Figura 3 tem-se a seqüência de primitivas de um pedido de conexão bem sucedido. Ao receber um pedido de conexão (primitiva MDL_ESTABLISH.Request) difunde-se uma PDU pedindo abertura de conexão aos receptores. O provedor do serviço (nos receptores) ao receber esta PDU, envia uma primitiva MDL_ESTABLISH.Indication ao usuário receptor e uma PDU de reconhecimento ao transmissor. Este último, ao receber as PDUs deste tipo de todos os receptores, confirma a abertura de conexão ao usuário transmissor pelo envio da primitiva MDL_ESTABLISH.Confirm. É importante observar que a resposta de abertura de conexão se faz a nível de provedor do serviço sem que o usuário receptor possa interferir. Se o usuário não "quiser" a conexão ele deverá enviar um pedido de desconexão após a abertura.

A seqüência de primitivas usada na transferência de dados (difusão) do usuário está mostrada na Figura 4. A chegada de uma primitiva MDL_DATA.Request ao serviço causa o envio de uma primitiva MDL_DATA.Indication em cada receptor.

Vale ressaltar ainda o uso da primitiva MDL_RELEASE.Request por um usuário receptor, que causa a primitiva MDL_RELEASE.Indication para o usuário transmissor (Figura 5). Se este decidir terminar a conexão, enviará uma primitiva MDL_ABORT.Request que causará o envio de primitivas MDL_ABORT.Indication em todos os usuários receptores, liberando a conexão. A primitiva MDL_ABORT.Confirm é enviada ao usuário transmissor após a confirmação de todos os receptores. Essa PDU de confirmação é enviada por iniciativa do provedor do serviço pois a liberação é abrupta. No entanto, se um ou mais receptores não responderem, novas tentativas de liberação serão feitas e, se após algumas tentativas ainda assim não houver resposta, a primitiva de confirmação é enviada ao usuário, liberando-o. As demais combinações podem ser: liberação

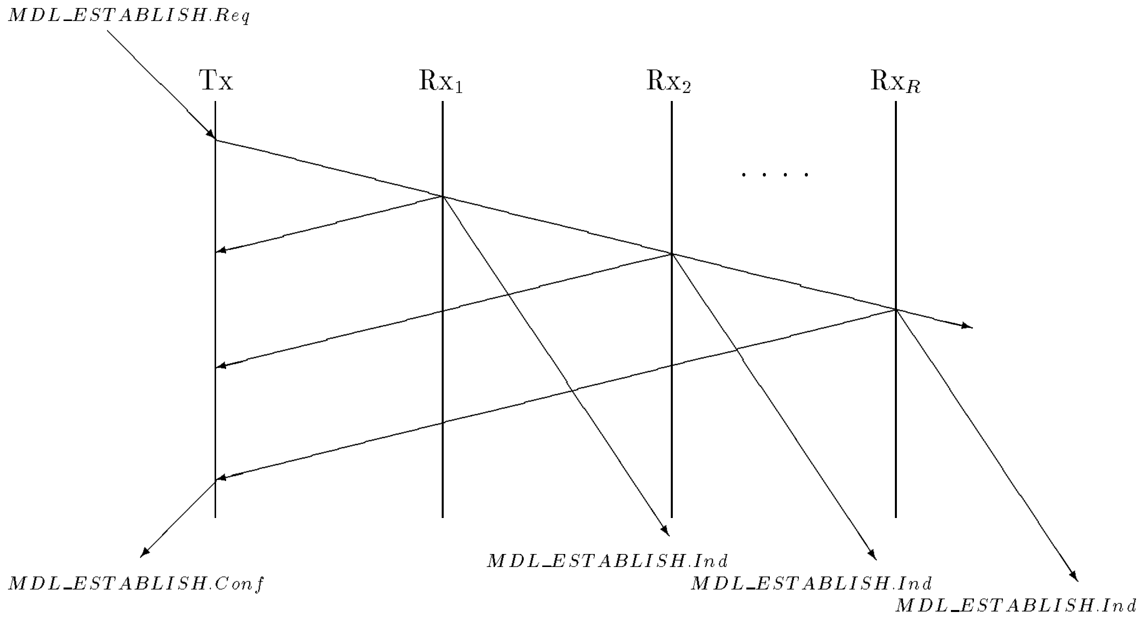

Figura 3: Estabelecimento de conexão bem sucedida

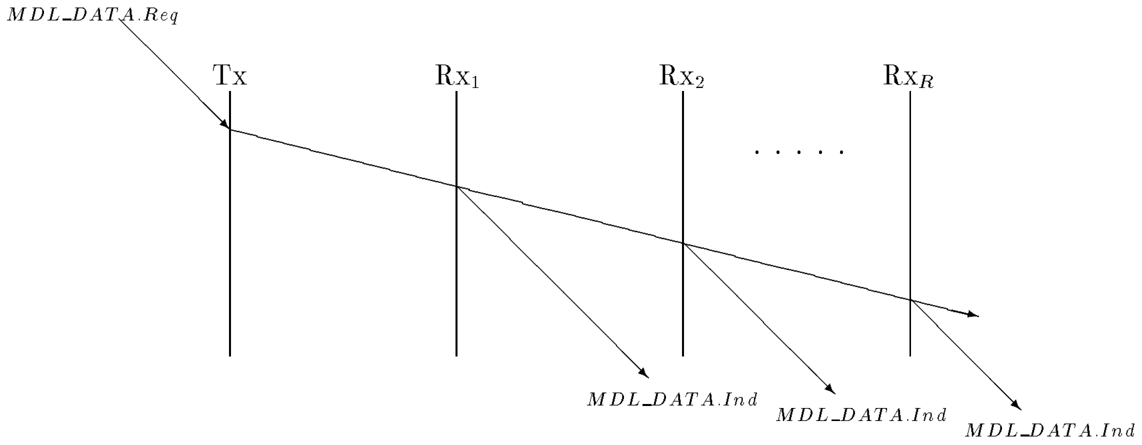

Figura 4: Seqüência de primitivas
Difusão de dados

por iniciativa do serviço; liberação por iniciativa do usuário transmissor e pedido de release por um receptor sem aceitação pelo transmissor. As seqüências de primitivas para estes casos, são análogas à Figura 5, removendo-se as primitivas não pertinentes.

Neste serviço, a PDU de informação também tem o mesmo formato que nos anteriores (Fig. 1). As demais PDUs, mostradas na Figura 6, carregam informação de: reconhecimento positivo (RR); reconhecimento negativo (SREJ); pedido de estabelecimento de conexão (SMARM); reconhecimento de quadros não numerados (UA); liberação abrupta (MDISC) e pedido de liberação (DM).

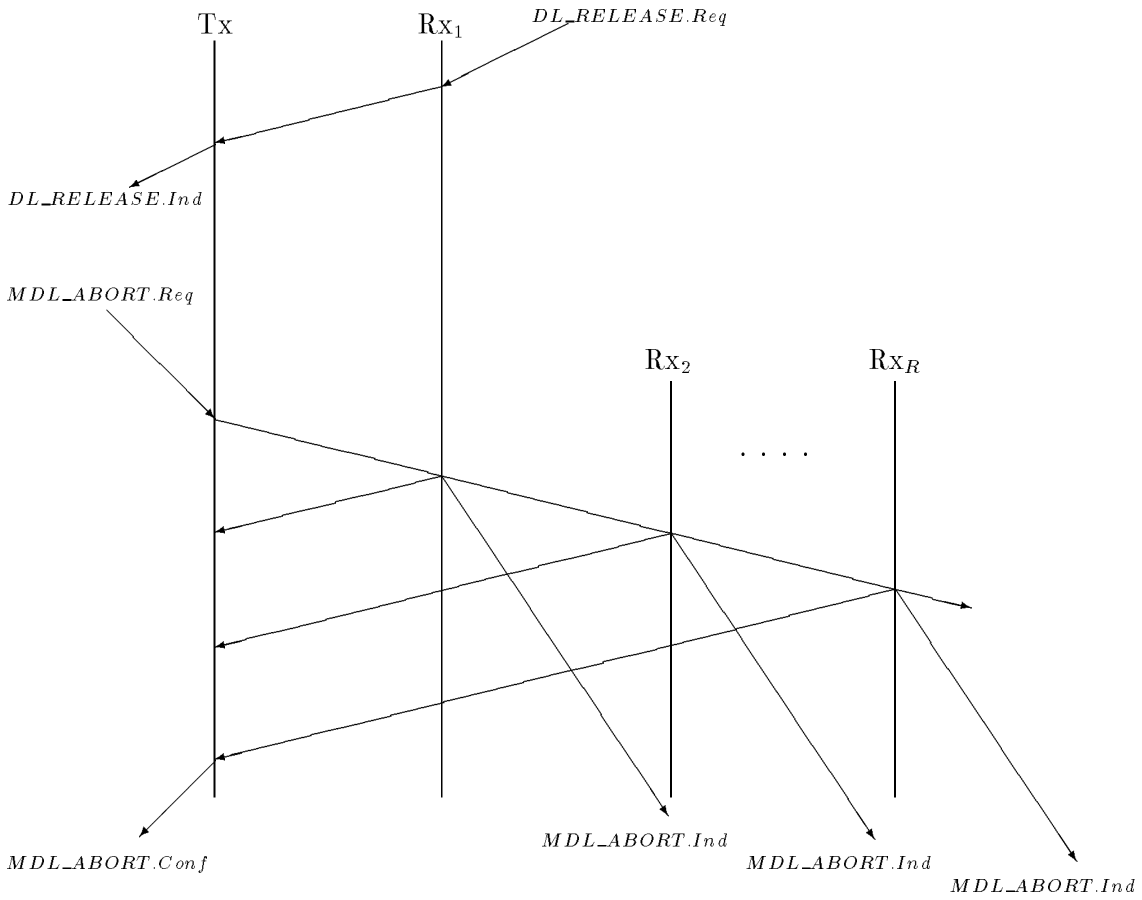

Figura 5: Seqüência de primitivas. Pedido de liberação aceito pelo usuário transmissor

3 Os Ambientes de Rede de Satélite e de Rede Local

O meio de comunicação via satélite é multi-usuário e necessita de um controle de acesso. No entanto, o serviço que é oferecido pela EMBRATEL ao usuário garante um canal de comunicação com uma banda passante fixa durante um determinado período de tempo que é contratado *a priori*.

Neste trabalho, este serviço é simulado em um único computador do tipo compatível IBM-PC. Criou-se uma interface contendo filas (de transmissão e de recepção) que doravante será chamada de interface da Camada Física e o simulador se encarrega de tratar as primitivas de serviço contidas nestas filas. Assim, a difusão de uma mensagem consiste em receber uma primitiva do transmissor (PH_DATA.request) e gerar as primitivas correspondentes nos receptores (PH_DATA.indication). No serviço com canal de retorno cada primitiva da fila de transmissão dos receptores (PH_DATA.request) gera uma primitiva na fila de recepção do transmissor (PH_DATA.indication). Para simular erros no meio de comunicação foi realizado um algoritmo que gera erros similares aos provocados pelo canal de subida (todos os receptores não recebem os dados transmitidos) e pelo canal descida (apenas um ou mais receptores não recebem os dados enviados). As taxas de erros nestes canais são diferentes como é o caso do satélite.

O meio de comunicação usado como ambiente de rede local é baseado na rede do tipo Ethernet. O serviço proposto usa o serviço oferecido pela camada MAC como é definido na norma IEEE 802.3 [6]. Esta camada usa um conjunto de rotinas *freeware* conhecidas como *Packet Driver SPEC*. O *Packet Driver* tem a finalidade de servir de interface padrão para as

RR

Delim.	End.Fonte	End.Dest.	Cód.	Ns	CRC	Delim
1	1	1	1	4	2	1

num.Octetos

SREJ

Delim.	End.Fonte	End.Dest.	Cód.	Ns	CRC	Delim
1	1	1	1	4	2	1

num.Octetos

SMARM

Delim.	End.Fonte	End.Global	Cód.	CRC	Delim
1	1	1	1	2	1

num.Octetos

UA

Delim.	End.Fonte	End.Dest.	Cód.	CRC	Delim
1	1	1	1	2	1

num.Octetos

MDISC

Delim.	End.Fonte	End.Global	Cód.	CRC	Delim
1	1	1	1	2	1

num.Octetos

DM

Delim.	End.Fonte	End.Dest.	Cód.	CRC	Delim
1	1	1	1	2	1

num.Octetos

Figura 6: Codificação de PDU - RR, SREJ, SMARM, UA, MDISC e DM

diferentes placas *Ethernet* existentes no comércio oferecidas como interface para computadores pessoais compatíveis IBM-PC.

Os serviços oferecidos pela Camada Física na rede de satélite e pela Camada MAC na rede local e utilizados nestes trabalhos, são:

- PH_DATA.Resquest
  Parâmetros: Info, Tam_Dados_Usuario;
- PH_DATA.Indication
  Parâmetros: Info, Tam_Dados_Usuario.
- MA_DATA.Request
  Parâmetros: End_Destino, Quality_of_Service, Info, Tam_Dados_Usuario;
- MA_DATA.Indication
  Parâmetros: End_Origem, End_Destino, Status, Quality_of_Service, Info, Tam_Dados_Usuario;

Segue uma breve descrição dos parâmetros existentes:

- End_Origem: endereço lógico da estação de onde partiu o quadro (inteiro sem sinal);
- End_Destino: endereço lógico da estação de destino (inteiro sem sinal);
- Quality_of_Service: Qualidade do serviço (inteiro sem sinal);
- Info: Dados do usuário (*string* de caracteres);

- Tam_Dados_Usuário: Tamanho dos dados do usuário (inteiro sem sinal).

No ambiente Satélite, a camada Física do transmissor é ligeiramente diferente do receptor, pois o transmissor só faz difusão (endereçamento global), enquanto que o receptor só envia quadros com o endereço da estação transmissora como destino.

4 Arquitetura de Implementação

Um sistema arquitetado em camadas e que segue o Modelo de Referência de Interconexão de Sistemas Abertos possui como característica básica a definição de um certo número de serviços (primitivas) e um grande número de operações de adição/remoção de PCI (*Protocol Control Information*) às/das SDUs relativo às informações entre entidades pares. O desempenho de um sistema deste tipo é muito dependente da implementação. Entre os principais fatores que limitam o desempenho podem-se citar a recópia, o gerenciamento de memória e de temporizações e também as chamadas ao sistema operacional. O GTA da UFRJ definiu uma arquitetura de implementação baseada no conceito de memória comum [7, 8] que visa a obtenção de um sistema modular, portátil e de alto desempenho. Os sistemas descritos neste trabalho seguem esta arquitetura de implementação que tem os seguintes elementos de base:

- estrutura de dados;
  Duas estruturas de dados principais foram definidas afim de permitir um encadeamento dos dados sem fazer recópias. A primeira delas, denominada "Estrutura_Primitiva", é usada para passar as primitivas entre as camadas adjacentes. A segunda estrutura, denominada "Estrutura_Parametros", é usada para o encadeamento das PCIs e também faz referência aos parâmetros da primitiva de serviço.

- interfaces;
  A comunicação entre camadas é feita por envio de mensagens (primitivas). As interfaces entre as camadas são constituídas por filas FIFO limitadas, simplesmente encadeadas, cujos elementos são "Estruturas_Primitiva". Para gerenciar a inserção e retirada dessas filas, utilizam-se variáveis e funções que permitem verificar o estado das filas e enviar/retirar primitivas destas filas.

- Gerenciador de Memória;
  O gerenciamento de memória é feito por um módulo independente a fim de melhorar o desempenho e otimizar a utilização da memória disponível. Ele divide a memória em duas regiões: transmissor e receptor; cada uma subdividida em três sub-regiões: uma para armazenar apenas os elementos da "Estrutura_Primitivas", outra para armazenar os elementos da "Estrutura_Parametros" e uma última para dados em geral. As duas primeiras sub-divisões são constituídas por blocos de tamanho fixo (9 octetos para a área de "Estrutura_Primitivas" e 15 octetos para a área de "Estrutura_Parametros"), o que elimina os problemas de fragmentação de memória e ainda diminui os tempo de alocação e liberação dessas estruturas. Já a terceira sub-área permite alocações de tamanho variável. Para gerenciar esta sub-área, optou-se por armazenar em listas à parte os ponteiros de início, os tamanhos das regiões alocadas e as informações relativas às regiões livres.

- Escalonador de Tarefas;
  A versão atual do Escalonador determina qual função deve ser executada através de uma política FIFO, ou seja, ao enviar uma primitiva de serviço coloca-se numa fila FIFO o nome da função que irá receber e tratar essa primitiva (ou no caso do estouro de uma das temporizações, coloca-se a rotina que irá tratar a expiração desse temporizador). Posteriormente, de acordo com a chegada, essas funções vão sendo tratadas uma a uma, o que

implica em que o programa principal fique em um *loop* infinito, verificando o usuário e executando essas funções.

- Gerenciamento de Temporizações.
  Este módulo específico gerencia contadores de tempo usando o esquema proposto por Varguese & Lauck [9], onde associa-se, a cada contador de tempo, um descritor a ser colocado numa posição específica de um *buffer* circular (de acordo com o tempo). A cada interrupção periódica (acionada por hardware) incrementa-se o apontador desse *buffer*. O estouro de um determinado contador de tempo ocorre quando a posição de seu descritor estiver sendo apontada pelo ponteiro do *buffer*. Nesse caso, coloca-se a rotina de tratamento de estouro de tempo da camada correspondente na Fila do Escalonador.

O *software* específico aos serviços ponto a multiponto segue uma estrutura similar para os três protocolos implementados. Pode-se dividir estas implementações em 7 partes: a parte da máquina de estados que trata os eventos gerados pelo provedor do serviço (Camada Física no ambiente de Satélite e MAC no ambiente de Rede Local), a parte dos eventos gerados pelo usuário do serviço a parte que tratam o estouro dos contadores de tempo, as rotinas que montam e enviam as primitivas, as rotinas que montam e enviam as PDUs, as rotinas de manipulação das listas internas (a Lista de Quadros Armazenados, as Listas de Quadros Esperados e as Listas de Quadros Recebidos Fora de Ordem) e as rotinas gerais (decodificação de PDUs, controle de seqüência e descarte de quadros).

As partes referentes à máquina de estado foram feitas utilizando-se uma série de sentenças *if-then-else* e/ou *switch* para definir quais ações devem ser executadas para cada interseção evento-estado válida.

Uma vez que as únicas primitivas que possuem parâmetros são as de dados e esses parâmetros são os Dados do Usuário (MDL_DATA.Request e MDL_DATA.Indication), a montagem das primitivas tornou-se bastante simples. Para montá-las, basta alocar e preencher uma "Estrutura_Primitiva", sendo necessária a alocação e o preenchimento de uma "Estrutura_Parametros" apenas no caso das primitivas de dados.

Já o envio de PDUs é um pouco mais complexo, principalmente no caso das PDUs de informação, além de montar a primitiva DL_PH, deve-se preencher o "Buffer_ Tx". Após alocar um quadro desse *buffer*, deve-se preenchê-lo com a PCI da PDU octeto a octeto.

O grupo de rotinas que manipulam as listas internas é composto de rotinas de inserção, retirada e atualização da Lista de Quadros Armazenados, das rotinas de inserção e retirada das Listas de Quadros Esperados e da rotina de inserção nas Listas de Quadros Recebidos Fora de Ordem. As inserções na Lista de Quadros Armazenados e nas Listas de Quadros Esperados são feitas como numa fila FIFO, ou seja, sempre na última posição da fila. Já a retirada dessas listas pode ser feita em qualquer posição. Por conseguinte, é necessário fazer uma busca para encontrar o elemento a ser retirado e, nesta primeira versão, optou-se por um algoritmo de busca linear. Além de liberar a memória ocupada pela estrutura e pelos campos do elemento da lista retirado, estas rotinas retornam o ponteiro para o elemento da lista apontado pelo elemento retirado, isto é, o elemento seguinte ao retirado. A atualização da Lista de Quadros Armazenados por sua vez, é feita de acordo com o conteúdo do quadro de controle recebido.

Encerrando esse grupo de rotinas de manipulação de listas internas, tem-se a inserção na lista de Quadros Recebidos Fora de Ordem. Nesta lista, os quadros devem ser inseridos em ordem crescente de seus números a fim de facilitar seu envio para o usuário. Portanto, optou-se por uma lista duplamente encadeada para simplificar o processo de inserção.

Por fim, no grupo de rotinas gerais, valem a pena ser mencionadas a existência dos algoritmos de verificação de seqüência e descarte de quadros da Lista de Quadros Armazenados. Para o protocolo com rotação de permissão o descarte de quadros é feito apenas caso todas as estações tenham enviado um reconhecimento (ACK ou NACK) e o quadro da lista não esteja selecionado para ser retransmitido. No protocolo ponto a multiponto com canal de retorno sempre que o transmissor receber um quadro RR, deve-se verificar se é possível descartar os quadros da Lista de Quadros Armazenados com número estritamente menor que Nr (número recebido no quadro RR). Isto é feito numa rotina que percorre a lista enquanto o número dos quadros armazenados for menor que Nr e testa se cada quadro foi reconhecido por todos os receptores. O acesso seqüencial à lista e, principalmente o teste do quadro ter sido reconhecido por todos acarretam num grande *overhead* para o desempenho. Por conseguinte, um dos fatores limitantes no desempenho será o tratamento dos quadros RR, principalmente com número elevado de receptores.

Afim de se poder efetuar os testes dos protocolos foi necessário a implementação, para cada serviço, de um usuário e um simulador do satélite. Os simuladores possibilitam introduzir erros através da perda de quadros, variar o tamanho destes quadros e o número de estações do sistema. Foi implementado um simulador em separado para cada protocolo. Cada um destes simuladores pode ser dividido em cinco módulos principais: Aplicação, Interface N_DL, Protocolos, Interface DL_PH e Físico.

O módulo de Aplicação é responsável pela interface com o usuário, enviando primitivas de acordo com seu comando, sendo específico para cada protocolo.

Embora existissem apenas duas filas nas interfaces entre camadas descritas anteriormente para estes serviços, o fato de se estar simulando várias estações numa mesma máquina fez com que se utilizasse duas filas para cada estação. O módulo N_DL também tem diferenças entre os três serviços, uma vez que as primitivas da interface N_DL, tipos de dados e variáveis apresentam diferenças. Já a interface DL_PH é a mesma para todos os protocolos descritos.

Para poder simular várias as estações, indexaram-se as variáveis globais da máquina de estados de cada protocolo. Assim, por exemplo, a estação número 1 recebe seus quadros pela fila PH_DL[1] compara o número do quadro recebido com Vr[1] e, no caso de haver quebra de seqüência, armazena-o na Lista de Quadros Recebidos Fora de Ordem[1].

A simulação fica patente no módulo Físico, onde os dados são copiados do "Buffer_Tx" para o "Buffer_Rx" diretamente, sem passar por nenhum canal físico real. É nesta etapa que os erros de transmissão são simulados, através da não cópia de um quadro de um *buffer* para outro. Os erros são simulados com o auxilio de uma função que gera números pseudo-aleatórios de 0 a 100. Se o número gerado for maior do que a porcentagem de erro especificada, o quadro será copiado, caso contrário não.

Embora todos os serviços possam ser adaptados para o ambiente de rede local, o serviço de maior interesse é o ponto a multiponto com canais de retorno. A exemplo dos simuladores, esta implementação também pode ser dividida nos mesmos cinco módulos já citados. Foram criados dois programas executáveis independentes, um para o transmissor e outro para cada receptor. Os módulos foram reprogramados mudando-se os nomes das interfaces, das primitivas que provem os serviços MAC e também retirou-se as indexações das filas e variáveis que simulavam a difusão. O módulo aplicação foi dividido em Tx e Rx.

5 Medidas de Desempenho

Para avaliar o desempenho dos protocolos foram feitas medidas de vazão, que correspondem à relação entre o volume de informação e o tempo necessário para transferí-lo. Entende-se por volume de informação o produto do número total de quadros de informação enviados por seus respectivos tamanhos. Estas medidas de vazão foram feitas a nível de usuário dos serviços ponto a multiponto propostos. Portanto, elas incluem o *overhead* introduzido pela criação/destruição das mensagens, envio/retirada das mensagens nas/das filas, alocação e de-alocação de memória, etc. Dois erros de medida são inerentes à metodologia empregada. Primeiro, considera-se como tempo total de transferência do volume de informações o tempo decorrido entre a transmissão da primeira mensagem e a transmissão da última mensagem, a nível da aplicação. Não se espera o reconhecimento positivo de todas mensagens enviadas. Segundo, este tempo é obtido com o auxilio de funções baseadas no relógio de um IBM-PC, o que limita sua precisão a 55 ms. Para tornar estes erros insignificantes o volume de informações a ser transferido é composto de 10.000 quadros, o que torna o tempo de transferência muitas vezes superior aos tempos não considerados.

5.1 Ambiente de Satélite

No ambiente de satélite, alguns aspectos devem ser levados em consideração. O simulador não considera o atraso do meio físico pois isto implicaria em limitar o número de receptores devido a capacidade de memória disponível. A difusão de uma mensagem para R receptores é simulada por R cópias da informação. Ainda pelo fato da estação transmissora e as estações receptoras estarem implementadas no mesmo equipamento o processador é compartilhado pelas várias estações. A cada transmissão o simulador realiza R cópias simulando as R recepções e só então uma nova transmissão é efetuada. É uma situação similar a um ambiente onde a estação transmissora ao mesmo tempo que difunde uma mensagem recebesse e tratasse R mensagens provenientes de outras estações. O tempo total de transferência considera como tempo final a ultima cópia realizada da última transmissão, ou seja, a simulação da última recepção. Estes fatores levam a resultados de medidas bastante degradados em relação a um ambiente real.

5.1.1 Difusão Simples

Para efeito de medidas neste serviço fez-se uma janela maior que os 10.000 quadros enviados, e portanto, a simulação correu sem a influência do controle de fluxo. A Tabela 4 apresenta os resultados.

Tabela 4: Vazão em função do tamanho do quadro para o protocolo de difusão simples.

Simulação
IBM-PC 286 - 8Mhz
Volume = 10.000 quadros
sem perdas
3 receptores
Limite de fluxo =
11.000 quadros / 55 ms

Tamanho (Oct.)	**Tempo** (s)	**Vazão** (oct./s)
64	96,2	6.652,8
128	98,3	13.021,4
256	102,6	24.951,3
512	110,9	46.167,7
1024	127,4	80.376,8
2048	160,9	127.284,0
4096	228,3	179.413,1

5.1.2 Rotação da Permissão de Transmissão

No serviço de difusão com rotação de permissão, as medidas se limitaram ao envio de poucos quadros (Tabela 5), tanto menos quanto maior o tamanho do quadro. Este número pequeno de quadros é devido à maior necessidade de memória (limitada a 64 koctetos), que, por sua vez, se deve ao fato da liberação dos quadros enviados ocorrer somente após o reconhecimento positivo, o que implica em pelo menos uma rotação completa da permissão de transmissão. Utilizando-se mais quadros, a transmissão destes teria que ser feita em etapas, esperando uma ou mais rotações da permissão de transmissão, o que traria a vazão a níveis bem mais baixos. Outra solução seria a utilização de área de memória para *buffer* maior do que 64 koctetos. Isto implica em manipulação de ponteiros "far" (com troca de segmento). Esta última solução se mostrou muito mais lenta, de modo que foi preferível manter esta limitação.

Tabela 5: Vazão em função do tamanho do quadro para o protocolo de difusão com rotação de permissão.

	Quantidade de quadros	**Tamanho** (Oct.)	**Tempo** (s)	**Vazão** (oct./s)
Simulação IBM-PC 286 - 8Mhz sem perdas 3 receptores Limite de fluxo = 255 quadros / 55ms	700	64	12,8	3.500,0
	440	128	5,6	10.057,1
	220	256	2,8	20.114,3
	110	512	1,0	56.320,0
	60	1024	0,7	87.771,4

Na análise destas medidas, dois fatos devem ser levados em consideração. Primeiro, o tempo medido foi apenas o da transmissão e o da recepção dos quadros por todos os receptores, sem esperar pelo envio das confirmações, conforme já mencionado. Segundo, esta transmissão foi feita em uma única fatia de tempo de permissão de transmissão.

5.1.3 Ponto a Multi-ponto com Canal de Retorno

O serviço ponto-a-multi-ponto com canal de retorno é a implementação mais elegante, devido ao controle de fluxo explícito, por janela de emissão. Na Tabela 6 são mostradas as vazões em função do tamanho do quadro. Observa-se nas três tabelas citadas até agora, uma melhora significativa do desempenho, a medida em que se aumenta o tamanho do quadro. Isto se deve a arquitetura de implementação usada que ao evitar recópias e trabalhar com ponteiros faz com que o processamento fique pouco dependente do tamanho da mensagem e portanto tem-se maiores vazões para mensagens de maior comprimento.

Na Tabela 7 são mostradas as vazões em função do número de receptores. Observa-se que os resultados obtidos são bastante prejudicados pelo fato da implementação, no simulador, ter sido realizada em um único microcomputador, pois o processador tem que manipular 16 estações, no pior caso.

A Tabela 8 mostra as vazões em função do tempo entre envio de reconhecimentos positivos. Como era de se esperar a vazão aumenta com uma maior temporização dos reconhecimentos. Isto deverá ser ainda mais significativo para um maior número de receptores e em uma implementação onde onde o transmissor e os receptores estiverem em máquina separadas, pois neste caso o tempo de processar os reconhecimentos será corresponderá mais perto do tempo de transmissão de uma mensagem.

Tabela 6: Vazão em função da variação do tamanho do quadro para o protocolo ponto-a-multi-ponto com canal de retorno.

Simulação
IBM-PC 286 8MHz
Volume=10.000 quadros
sem perdas
2 receptores
$Tack$=3x55ms
Janela=50

Tamanho (Oct.)	**Tempo** (s)	**Vazão** (oct./s)
64	126,8	5.047,3
128	130,0	9.846,2
256	135,1	18.948,9
512	142,2	36.005,6
1024	159,8	64.080,1
2048	195,2	104.918,0
4096	266,5	153.969,1

Tabela 7: Vazão em função da variação do número de receptores para o protocolo ponto-a-multi-ponto com canal de retorno.

Simulação
IBM-PC 286 - 8Mhz
Volume = 10.000 quadros
de 1024 octetos
sem perdas
T_{ack}=3x55ms
janela=50

Num. de receptores	**Tempo** (s)	**Vazão** (oct./s)
3	159,8	64.080,1
5	267,1	38.337,7
10	629,1	16.277,2
15	1.167,4	8.771,6

A tabela 9 mostra a vazão do sistema inserindo-se a simulação de erros. Pelo fato da retransmissão ser seletiva, o desempenho não é muito degradado, mesmo para taxas mais elevadas do que aquelas encontradas nos meios de comunicação existentes. Quando há algum erro no canal de difusão, os receptores tem que guardar os quadros recebidos após a detecção do erro, a espera do quadro retransmitido, para poder enviar os quadros em seqüência para o usuário. Devido à limitação de memória em 64Kb, e o fato do simulador utilizar esta memória para os três receptores, limitou-se a janela de emissão a 20. Com quadros de 1024 octetos, este valor seria o limite para não haver estouro de memória nos receptores (64Kb / 3 / 1024 $\doteq$ 20).

Tabela 8: Vazão em função do tempo de envio de reconhecimentos positivos para o protocolo ponto-a-multi-ponto com canal de retorno.

Simulação
IBM-PC 286 - 8Mhz
Volume = 10.000 quadros
de 1024 octetos
sem perdas
5 receptores
janela=50

T_{ack} (x55ms)	**Tempo** (s)	**Vazão** (oct./s)
3	159,8	64.080,1
5	156,5	65.431,3
10	153,2	66.840.7
15	164,8	62.135,9
20	244,0	41.967,2
25	304.3	33.651,0

Tabela 9: Vazão em função da porcentagem de perda de quadros na difusão e no canal de retorno para o protocolo ponto-a-multi-ponto com canal de retorno.

Simulação
IBM-PC 286 - 8Mhz
Volume = 10.000 quadros de 1024 octetos
3 receptores
T_{ack}=3x55ms
janela=20

Perda no canal de ret. (%)	**Perda na difusão** (%)	**Tempo** (s)	**Vazão** (oct./s)
0	0	159,8	64.080,1
0,01	0,1	161,3	63.484,2
0,02	0,2	162,9	62.860,7
0,03	0,3	166,8	61.390,9
0,04	0,4	167,0	61.317,4

5.2 Ambiente de Rede Local

No ambiente de rede local, utilizou-se três microcomputadores (IBM-PC 80286 8MHz) conectados a uma rede Ethernet, sendo um transmissor e dois receptores. Foram feitas medidas apenas do desempenho do protocolo ponto-a-multi-ponto com canal de retorno. As medições foram realizadas sob condições de baixa carga na rede, isto é, só estavam trafegando na rede os quadros pertinentes às medidas.

Nota-se já na Tabela 10 que há um considerável aumento no desempenho, conforme estava previsto. Isto fica claro comparando-se as Tabelas 6 e 10. Com apenas dois receptores a carga de processamento da parte transmissora é mais leve que a deles, fazendo com que o transmissor transmita mais rápido do que os receptores conseguem receber e processar. O valor 50 foi mantido para a janela devido a limitação de memória no *buffer* de recepção (64Kb), conforme descrito anteriormente. Este valor é coerente com quadros de 1024 octetos. Pode-se usar janelas maiores para os quadros menores.

Tabela 10: Vazão em função da variação do tamanho do quadro para o protocolo ponto-a-multi-ponto com canal de retorno.

Rede Ethernet
Volume=10.000 quadros
2 receptores
T_{ack}=3x55ms
janela=50
sem perdas

Tamanho (Oct.)	**Tempo** (s)	**Vazão** (oct./s)
64	41,8	15.331,0
128	43,1	29.698,4
256	44,6	57.399,1
512	49,4	103.643,7
1024	59,0	173.559,3

A Tabela 11 mostra a vazão obtida com simulação de erros. Os valores apresentados na referida tabela são de medidas específicas e não da média. Aparecem nesta Tabela (11) o número de erros efetivamente gerados (quadros perdidos) no receptor 1 e 2, e também no canal de retorno. Uma outra coluna mostra o montante de SREJ recebidos pelo transmissor. Analisando as últimas colunas, pode-se chegar a conclusões quanto a perdas de quadros pelos dois receptores ou somente por um deles, como também quanto à perda de quadros retransmitidos.

As medidas do protocolo ponto a multiponto com canal de retorno considerando o ambiente de satélite simulado foram também realizadas em um IBM-PC 386 DX com 33MHz de relógio

Tabela 11: Vazão em função da porcentagem de perda de quadros na difusão e no canal de retorno para o protocolo ponto-a-multi-ponto com canal de retorno.

Rede Ethernet / Volume = 10.000 quadros de 256 octetos
2 receptores / T_{ack}=3x55ms / janela=20.

Perda no canal ret. (%)	**Perda na difusão** (%)	**Tempo** (s)	**Vazão** (oct/s)	**Erros no no Rec. 1**	**Erros no Rec. 2**	**Erros no canal ret.**	**Num. de SREJ**
0,01	0,1	45,1	56.672,7	20	13	0	33
0,02	0,2	45,1	56.672,7	33	34	0	67
0,03	0,3	48,0	53.333,3	51	53	2	103
0,04	0,4	53,2	48.120,3	77	82	0	153

e *cache-memory* de 64 kb. Houve uma melhora de seis vezes no resultado da vazão que chegou a 3.22 Mbps para quadros de 1024 octetos, 3 receptores, sem perdas, janela de 128 e T_{ack} de 55 ms. Caso a mesma melhora se produza no ambiente de rede local, uma vazão perto de 8 Mbps será alcançada para dois receptores.

6 Conclusão

Neste trabalho três serviços de transferência de dados foram apresentados: difusão simples, difusão com rotação de permissão e ponto a multiponto com canal de retorno.

Os protocolos foram implementados e testados. Uma aplicação simples foi realizada permitindo a utilização dos serviços. As implementações seguiram uma arquitetura que visa um alto desempenho e uma fácil portatibilidade. Esta arquitetura de implementação empregada se mostrou eficaz permitindo taxas de transmissão elevadas e tornando o processamento pouco dependente do tamanho das mensagens. As diversas medidas de desempenho efetuadas considerando vários comprimentos de quadros, diversos receptores e sob condições adversas de erros indicam que mesmo com estações de baixíssimo custo (microcomputadores pessoais compatíveis IBM-PC) é possível transferir dados de uma maneira confiavel usando a capacidade máxima do serviço de difusão à alta velocidade (64 kbps) oferecido pela EMBRATEL através do BRASILSAT.

A adaptação do serviço ponto a multiponto com canal de retorno para o ambiente de rede local se mostrou bastante satisfatória e seu desempenho alcançou 1.3 Mbps para uma comunicação entre microcomputadores com processadores 80286 - 8 MHz.

Estes resultados podem ser considerados significativos e um protocolo de conferência seguindo estes conceitos está sendo especificado afim de permitir uma conferência (texto, figuras e imagens) utilizando computadores pessoais e estações de trabalho.

Ainda como resultado de um estudo mais detalhado da implementação concluiu-se que um protocolo multi-mídia em tempo real deverá buscar uma nova arquitetura mais adaptada aos dados síncronos, uma melhor eficiência no gerenciamento de memória e protocolos que diminuam (ou evitem) temporizações e reordenações de mensagens.

7 Agradecimentos

Os autores agradecem a todas as pessoas do Grupo de Teleinformática e Automação e em especial aos responsáveis e implementadores das rotinas necessárias ao Packet Driver e da Camada MAC: Prof. Marcelo Luiz Drumond Lanza, Prof. João Amaro Baptista Pereira, Frederico dos Santos Liporace e Marcelo Dias Nunes.

Referências

[1] F. M. C. de Barros, "Serviços de transferência de dados ponto a multiponto", Tese de Mestrado, Universidade Federal do Rio de Janeiro, PEL/COPPE/UFRJ, 100 p., Rio de Janeiro, janeiro 1992.

[2] C. X200, *Reference Model of Open Systems Interconnection for CCITT Applications.* CCITT Blue Book, vol. VIII, fasc. VIII.4, pp. 1-56, 1988.

[3] IS4335, *Information Processing Systems - Data Communication - High-level Data Link Control Procedures - Consolidation of Elements of Procedures.* International Standard Organization, 42 p., 1984.

[4] O. C. Duarte e H. M. de Lima, "O protocolo HDLC-Ms.SREJ+REJ Estendido - uma estratégia eficiente no ambiente de satélite", in *VIII Congresso Brasileiro de Automática*, Belem, pp. 1230–1236, setembro 1990.

[5] O. C. Duarte e H. M. de Lima, "Improving the satellite efficiency of the accumulative acknowledgement strategies", in *IEEE Global Telecommunications Conference GLOBECOM89*, Dallas, pp. 48.6.1–48.6.5, novembro 1989.

[6] I. P. 802, *Draft IEEE standard 802.3: CSMA/CD access method and physical layers specifications.* IEEE, Revision D, 1982.

[7] L. F. Baginski e O. C. Duarte, "Um modelo de implementação de alto desempenho para sistemas abertos", in *IX Congresso da Sociedade Brasileira de Telecomunicações*, São Paulo, pp. 19.3.1–19.3.5, setembro 1991.

[8] J. F. de Rezende e O. C. Duarte, "Implementação e análise de desempenho do protocolo de sessão", in *X Simpósio Brasileiro de Redes de Computadores*, Recife, pp. 289–302, abril 1992.

[9] G. Varghese e T. Lauck, "Hashed and hierarchical timing wheels: Data structures for the efficient implementation of a timer facility", in *Proccedings of the 11th ACM Symposium on Operating Systems Principles, ACM Operating Systems Review*, Austin TX, pp. 25–38, novembro 1987.