Serviço de transferência de dados multiponto-a-multiponto

Marcos Tadeu von Lutzow Vidal[1]

Departamento de Eng. de Telecomunicações
Universidade Federal Fluminense
E-mail: marcos@telecom.uff.br

Otto Carlos Muniz Bandeira Duarte *Aloysio de Castro Pinto Pedroza*

Univesidade Federal do Rio de Janeiro - COPPE/EE
E-mail: otto, aloysio@coe.ufrj.br

Resumo

Este trabalho apresenta o projeto completo de um novo serviço de enlace para comunicação multiponto-a-multiponto confiável. Este serviço se apoia na utilização de meios de transmissão de difusão, tais como satélite e rede local. Os elementos de serviços propostos incluem: abertura de conexão e inserção dinâmica de novos usuários, liberação abrupta e negociada, de uma forma individual ou total, da conexão; transferência de dados e gerência da conferência. As diferentes fases do projeto incluiram a especificação formal na FDT ESTELLE, o teste por simulação com geração automatizada de seqüências de teste e a obtenção de uma implementação baseada em uma arquitetura que privilegia o desempenho. A simulação e a geração de seqüências de teste foram feitas com o uso de ferramentas de um CAD para ESTELLE. A arquitetura de implementação consiste de módulos especializados de gerência de memória, gerência de temporizadores e escalonamento de tarefas, além de estruturas de dados utilizadas nas interfaces. Este trabalho foi desenvolvido usando técnicas modernas de engenharia de protocolos e os resultados obtidos foram bastante satisfatórios. Em particular, uma aplicação de texto-conferência implementada para testar o serviço apresentou um excelente desempenho.

Abstract

This work presents a complete project of a new reliable multipoint-to-multipoint data-link communication service. This service uses a diffusion transmission medium like satellite and local networks. The proposed service elements include: connection establishment and dynamic insertion of new users; abrupt and negotiated connection liberation, individual or total manner; data transfer and conference management. The different project phases included the formal specification using the FDT ESTELLE, the test through simulation based on automated test case generated and, finally, the implementation based on an architecture that focus the performance. The simulation and the test sequence generation have used tools offered by a ESTELLE CAD. The architecture for the implementation consists of a set of specialized modules for memory management, timers' management and task scheduling, and also of the structures for the interfaces. This work was developed using modern protocol engineering techniques, obtaining very interesting results. Specially, a text-conference application implemented to test the data-link service has shown an excellent performance.

---

[1] Trabalho de tese de mestrado realizado no Programa de Engenharia Elétrica da COPPE/UFRJ.

1. Introdução

Os sistemas de telecomunicações multidestinatário permitem reuniões entre membros de determinados projetos ou de empresas que necessitem tomar decisões conjuntas, sem o deslocamento físico destas pessoas, aumentam a agilidade na administração, além de diminuir os gastos com viagens e estadias [1]. Os sistemas de informações usados hoje em dia são, em sua maioria, baseados em comunicações ponto-a-ponto. Poucos sistemas oferecem serviços de transferência de dados confiáveis em configuração ponto-a-multiponto enquanto os que se servem de uma configuração multiponto-a-multiponto são quase inexistentes.

O Grupo de Teleinformática e Automação (GTA) da UFRJ desenvolve atualmente um ambiente de comunicação multiponto-a-multiponto visando aplicações multidestinatárias. O objetivo final é criar um ambiente de desenvolvimento que permita realizar aplicações multimídias, tais como: correio eletrônico multimídia, tele-conferência digital, edição cooperativa de textos multimídia, controle cooperativo de experiências remotas, projeto distribuído, etc.

Este trabalho descreve um serviço de comunicação confiável multiponto-a-multiponto. O serviço foi concebido para utilização em ambiente de satélite ou em rede local/metropolitana. O serviço e uma aplicação de texto-conferência descritas neste artigo foram implementadas [2] em um ambiente de rede local tipo Ethernet com computadores compatíveis IBM-PC. O serviço apresentado é uma evolução dos serviços ponto-a-multiponto abordados anteriormente [3,4]. A implementação, desenvolvida em linguagem C, segue uma arquitetura particular que privilegia o alto desempenho [5] e servirá de base para testes de futuras aplicações de conferência multimídia.

Na seção 2 o serviço multiponto-a-multiponto é descrito em linguagem natural procurando utilizar 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 [6]. Na seção 3 o protocolo é descrito sumariamente através de suas PDUs. A formalização da especificação é então apresentada na seção 4, juntamente com a arquitetura de especificação utilizada. A implementação é descrita na seção 5 e, na seção 6, é mostrada a aplicação criada para testar o serviço.

2. Serviço Multiponto-a-Multiponto

O serviço descrito neste trabalho é um serviço de enlace que provê a transferência bidirecional transparente e confiável de dados entre usuários, liberando-os da obrigatoriedade de possuir grande parte do conhecimento do mecanismo de comunicação. Este serviço utiliza como provedor um serviço de transferência não confiável e não orientado à conexão, onde cada estação possui canais de comunicação que difundem informação para as demais estações e também canais de comunicação ponto-a-ponto com as demais estações. O ambiente de utilização pode ser o satélite ou rede local/metropolitana. Supõe-se que no provedor de serviço (camada Física) as funções de delimitação/sincronismo de quadros, transparência (inserção/retirada de zeros) e cálculo/verificação de CRC encontram-se disponíveis, uma vez que estas funções normalmente são oferecidas em *hardware* pelos controladores de comunicação atualmente comercializados.

Exige-se que a camada Física seja composta de um canal de comunicação capaz de difundir dados de uma ou mais estações para várias outras assim como, no sentido inverso, as

estações receptoras possam enviar quadros de controle para a estação que enviou a mensagem. Devido às características do protocolo apresentadas mais adiante neste trabalho, pode-se utilizar somente um canal para difusão da informação e mais N canais para as informações de controle, para N estações que possam se conectar. Neste caso, para cada estação é alocado um canal (doravante chamado **canal secundário**) destinado à transmissão de suas informações de controle evitando, assim, colisões entre estas informações. O canal destinado à difusão de informações do usuário (doravante chamado **canal principal**) será compartilhado entre as estações (multiplexação no tempo através de *token*) de modo a não haver nenhuma colisão entre as informações. A Figura apresenta a utilização do ambiente satélite em três instantes distintos da utilização do canal principal para transmissão.

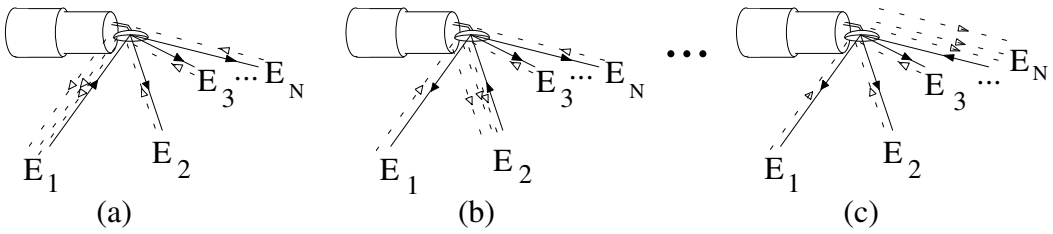


Figura 1 Exemplo de configurações de utilização do ambiente satélite (linha cheia = canal principal; linha pontilhada = canal secundário).

O ambiente de comunicação a ser utilizado também é facilmente obtido em redes locais e/ou metropolitanas ao se utilizar o serviço MAC (*Medium Access Control*), como é definido na norma ISO/IEC 8802.3 [7]. Neste caso, transmite-se informações com endereço multidestinatário para difusão de informação (*multicast*) e com endereços individuais para informações de controle.

O serviço de comunicação Multiponto-a-Multiponto irá oferecer as seguintes facilidades aos seus usuários: os meios de estabelecer uma conexão entre as estações participantes a fim de trocar unidades de dados de serviço (SDUs); os meios para inserir, negociadamente, uma ou mais estações como participante numa conexão já estabelecida; os meios para liberação negociada ou não, de uma ou mais estações; os meios para transferir SDUs de maneira confiável durante a conexão. Durante a transferência, o conteúdo das SDUs não será alterado, assim como não haverá nenhuma restrição quanto ao conteúdo das mesmas. A entrega dos dados aos usuários de destino é mantida na mesma seqüência em que foram recebidos, mesmo em caso de perdas de quadros no meio de transmissão. Isto caracteriza esta transferência como confiável e transparente para o usuário.

Uma vez que apenas um canal de difusão de alta capacidade é utilizado para transmissão de informação do usuário, em um dado momento apenas uma estação poderá estar transmitindo dados e as demais se comportarão apenas como receptoras. Para obter o efeito de transmissão multi-direcional é usada, neste canal de alta capacidade, a técnica de multiplexação assíncrona no tempo com controle distribuído, com a utilização de um *token* de permissão de transmissão. A gerência da comunicação é feita de maneira distribuída, ou seja, não existe uma estação monitora fixa: isto evita que uma pane na estação monitora venha a causar *deadlock*. Entretanto, a estação que estiver transmitindo em um dado momento (possui o *token*) possui deveres de uma estação monitora do serviço e, neste momento, é chamada de Mestra da conferência.

As exceções que podem ocorrer em um serviço multiponto-a-multiponto são variadas e em grande número, quando feita comparação com serviços ponto-a-ponto. É necessário monitorar continuamente o canal principal e também os canais secundários, de modo a manter a integridade da conexão. Estes mecanismos de garantia da integridade da conexão e das mensagens estão quase que totalmente baseados em temporizadores que, ao ter seu tempo esgotado, disparam funções que dependem basicamente dos seguintes fatores: o temporizador que estorou, o estado da conexão e, no caso do temporizador em questão tratar uma exceção, o número de reincidências desta exceção.

Como exemplo de funções de gerenciamento pode-se citar: a verificação da presença de todos os receptores, o recolhimento de eventuais pedidos de transmitissão, a inserção de novas estações que queiram participar numa conferência em andamento e a passagem do *token* para a próxima estação a transmitir. Para controlar a seqüência de estações a transmitir, a estação Mestra mantém uma fila de espera com a identidade das estações das quais recebeu um pedido de transmissão. A estação Mestra também difunde uma "lista de participantes da conferência".

A função de monitorização dos canais secundários (presença das estações em estado de recepção) está fortemente associada ao método de controle de erros implementado, que é baseado no HDLC (*High Level Data Link Control* [8]). Assim sendo, para a detecção de erros é necessário que os quadros de informação sejam numerados para permitir a detecção pela quebra de seqüência desta numeração. Os quadros de reconhecimento positivo (RR - PDU que contém o número do próximo quadro esperado, sendo que todos os quadros anteriores foram recebidos corretamente) são enviados pelas estações receptoras para a estação Mestra em uma cadência fixa no tempo que é determinada a priori. Se uma determinada estação deixar de enviar os quadros RR para a estação Mestra, esta detectará o silêncio da estação, o que causará a desconexão da estação silenciosa. É importante frisar que a conferência segue normalmente, sendo que a estação Mestra avisa às demais participantes da saída da estação, através da imediata divulgação de uma lista de participantes atualizada. O tempo máximo de silêncio permitido também é determinado a priori. No tocante ao controle de erros, existe ainda o quadro de reconhecimento negativo (SREJ) que, embora baseado no HDLC, teve a sua definição modificada de modo a permitir o envio de vários SREJ para diferentes quadros, sem que seja necessário que todos os quadros anteriores estejam corretos [9,10], ou seja, o SREJ não carrega consigo nenhuma informação de reconhecimento positivo. Ao receber um SREJ a estação Mestra retransmite imediatamente o quadro pedido. Foi utilizada a numeração absoluta (devido a simplicidade), sendo que esta numeração é individual, isto é, ela parte do zero em cada estação no estabelecimento da conexão.

Uma eventual duplicidade de Mestras (*tokens*) pode ocorrer no início da conferência e em caso de queda e posterior retorno de uma estação. Afim de evitar algoritmos complexos de votação, a prioridade para uma estação tornar-se Mestra, no caso de concorrência, é daquela de menor valor de identificação, para identificações numéricas.

Em caso de queda da atual estação Mestra, o que causa a perda do *token,* a recuperação é feita pela detecção do silêncio no canal principal, através do estouro de um contador de tempo existente em todas as estações em estado de recepção que, ao estourar causa o envio de um voto para a estação de menor identidade de sua Lista de Participantes. Uma estação que receber o convite a ser Mestra de pelo menos 50% das estações participantes assume como Mestra. Exceções como o recebimento de um Convite a ser Mestra de metade das estações participantes por uma estação em estado de Mestra também são tratadas.

Para evitar transbordo de *buffers* nos receptores, este serviço implementa um controle de fluxo no canal principal, baseado em janela deslizante de emissão. O valor desta janela é determinado *a priori* segundo os recursos disponíveis e os necessários. O bloqueio de transmissão é feito caso haja mais quadros esperando por reconhecimento positivo de alguma estação do que o número especificado para a janela.

Tabela 1 Resumo das primitivas do serviço multiponto-a-multiponto.

serviço	primitiva	parâmetros
Estabeleci-mento de conexão	MPDL_ESTABLISH.Request MPDL_ESTABLISH.Indication MPDL_ESTABLISH.Confirm	--- Identificação da estação origem. Modo da conexão e, identificação das estações conectadas.
Liberação de conexão	MPDL_ABORT.Request MPDL_ABORT.Indication MPDL_ABORT.Confirm MPDL_RELEASE_PLEASE.Request MPDL_RELEASE_PLEASE.Indication	--- Razão. --- --- Identificação da estação origem.
Transferên-cia de dados	MPDL_DATA.Request MPDL_DATA.Indication	Dados do usuário. Dados do usuário e identificação da estação origem.
Liberação de uma estação	MPDL_INDIVIDUAL_ABORT.Request MPDL_INDIVIDUAL_ABORT.Indication MPDL_INDIVIDUAL_ABORT.Confirm MPDL_INDIVIDUAL_RELEASE_PLEASE.Request MPDL_INDIVIDUAL_RELEASE_PLEASE.Indication	Identificação da estação alvo. Identificação da estação que está desconectando. Identificação da estação origem. --- Identificação da estação origem.
Gerência da conferência	MPDL_TOKEN_PLEASE.Request MPDL_TOKEN_PLEASE.Indication MPDL_TOKEN_GIVE.Request MPDL_TOKEN_GIVE.Indication MPDL_STOP_TRANSMISSION.Indication MPDL_STOP_TRANSMISSION.Response MPDL_REPLY.Request	--- Identificação da estação fonte. --- --- Razão --- Identificação da estação destino.

Sendo o serviço orientado a conexão e, devido às características multiponto-a-multiponto, pode-se dividi-lo em cinco fases distintas: estabelecimento da conexão inicial, transferência de dados, liberação da conexão, inserção de uma estação numa conexão existente e, liberação de uma estação, mantendo a conexão entre as demais. A inserção de uma nova estação numa conferência em andamento utiliza as mesmas primitivas da fase de iniciação da conferência. Durante a fase de transferência de dados pode-se identificar um grupo de primitivas relacionadas com a gerência da conferência, que foi colocado em separado por se tratar de primitivas ainda não tratadas na literatura (Tabela 1).

As primitivas MPDL_ ("*Multiple-Point Data Link*") dizem respeito, em sua maioria, às operações multiponto, e as primitivas MPDL_INDIVIDUAL_ às operações destinadas a uma só estação (Tabela 1). A seguir são mostrados três exemplos de seqüências de primitivas referentes a fase de conexão.

Na Figura 2 está ilustrada uma seqüência de conexão bem sucedida. Após receber a primitiva MPDL_ESTABLISH.Request, o serviço envia uma PDU que provoca nas estações que estiverem "ouvindo" o envio da primitiva MPDL_ESTABLISH.Indication para a camada superior. Na estação iniciante da conferência, após receber confirmação de todas as estações participantes, o serviço envia para a camada superior a primitiva MPDL_ESTABLISH.Confirm.

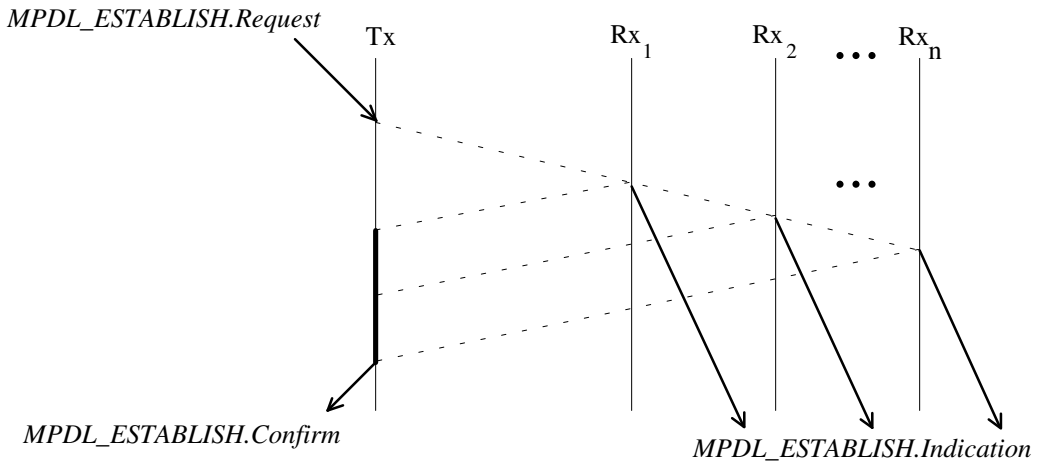


Figura 2 - Estabelecimento de conexão bem sucedida, com resposta de todas as estações.

Quando uma conexão já estiver estabelecida e uma nova estação desejar participar, a camada usuária também enviará a primitiva MPDL_ESTABLISH.Request e o serviço se encarregará de inserir a nova estação na conferência. Através de um parâmetro é possível à camada usuária que receber a confirmação, saber se sua estação realmente iniciou a conferência (Figura 2), caso em que ela inicia como Mestra, ou se está entrando em conexão já existente (Figura 3), caso em que a estação inicia como receptora e assim permanece até que a palavra lhe seja passada. Conforme mostra a Figura 3, logo após o envio da PDU que confirma a conexão para a estação que está entrando, o serviço envia uma PDU que causa o envio da primitiva MPDL_ESTABLISH.Indication nas demais estações participantes, indicando à camada usuária a entrada desta nova estação na conferência, através do parâmetro da primitiva .

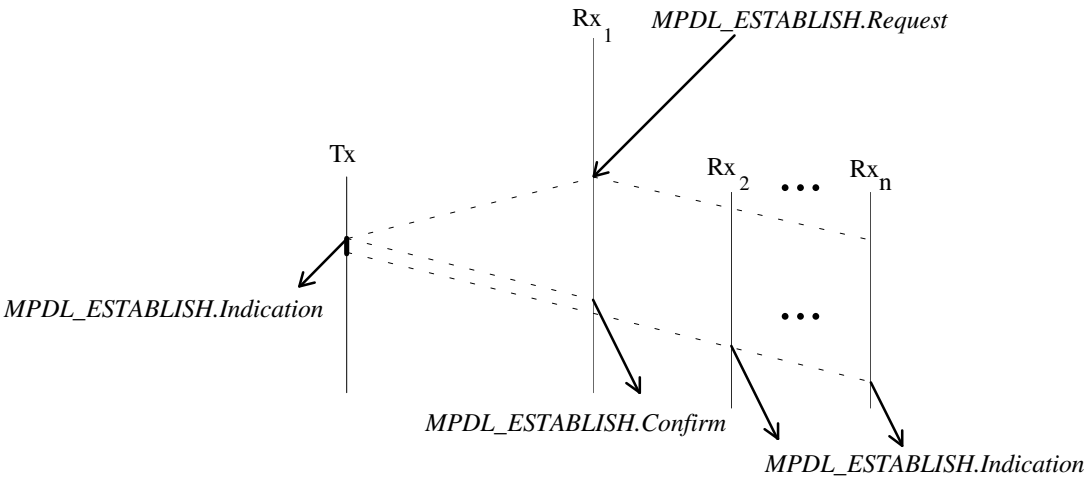


Figura 3 Entrada de uma nova estação em uma conexão já existente.

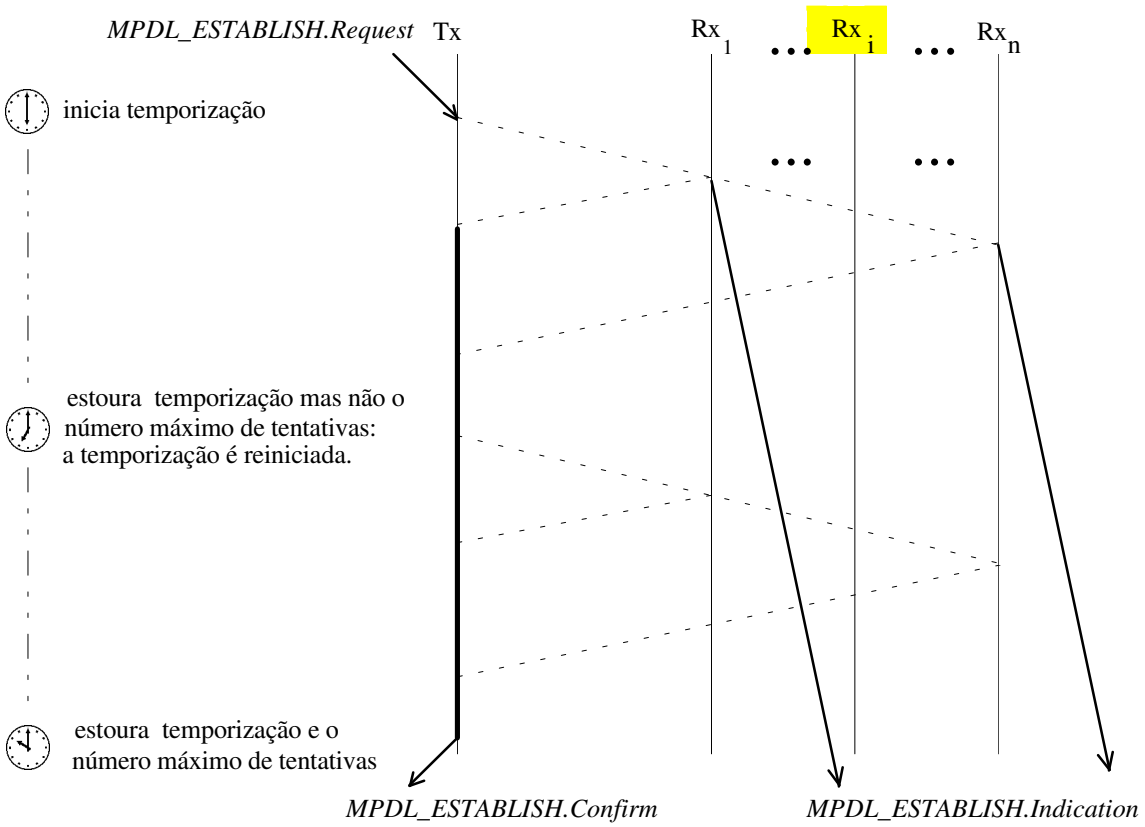


Figura 4 Estabelecimento de conexão bem sucedida, com resposta de algumas estações, mas não de todas.

Na Figura 4, a estação "Rx2" está desligada e não responde ao pedido de conexão enviado pela estação "Tx". O serviço realiza uma temporização e estabelece a conexão com as

estações que responderam. Neste caso, o envio do pedido de conexão multiponto pode ser enviado mais de uma vez, sendo o número máximo de tentativas estabelecido *a priori*.

3. O Protocolo

A máquina de estados finita do protocolo foi representada usando uma tabela evento/estado. Esta tabela possui 8 estados e 34 eventos. Estes eventos correspondem ao tratamento da chegada de 10 primitivas (*request* e *response*), 16 PDUs e ao tratamento de 8 temporizadores. Assim, esta tabela possui 272 interseções, onde 74 delas são tratadas pelo protocolo.

A Tabela 2 apresenta as PDUs separadas por unidades funcionais. Os quatro primeiros grupos funcionais apresentam PDUs que se interrelacionam dentro de cada grupo no protocolo, como, por exemplo, MPDISC - um pedido de desconexão - e MPDISCAC - o reconhecimento positivo correspondente. É importante notar que, para cada PDU enviada pela estação Mestra (difusão), devem ser recebidas e tratadas PDUs de reconhecimento de todas as demais estações participantes da conferência. Isto ocorre, por exemplo, com os quadros de informação. Para que um quadro transmitido possa ser descartado da Fila de Quadros a Espera de Confirmação, devem ser recebidos reconhecimentos positivos (RR) de todas as demais estações conectadas.

Tabela 2 Unidades funcionais e suas PDUs

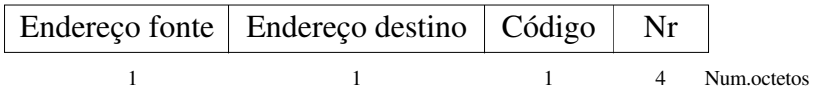
Unidade	Código da MPPDU	Nome da MPPDU
Estabelecimento de conexão	SMPARM	*Set MultiPoint Asynchronous Response Mode*
	MPARMAC	*MultiPoint Asynchronous Response Mode ACcept*
Liberação de conexão	MPDISC	*MultiPoint DISconnect Command*
	MPDISCAC	*MultiPoint DISconnect Command ACcept*
	MPRP	*MultiPoint Release Please*
Transferência de dados	MPI	*MultiPoint Information*
	RR	*Receive Ready*
	SREJ	*Selective REJect*
Liberação individual de uma estação	DISC	*DISconnect Command*
	DISCAC	*DISconnect Command ACcept*
	RP	*Release Please*
Gerência da conferência	PLIST	*Present LIST*
	TP	*Token Please*
	TG	*Token Give*
	BM	*Become Master*
	CM	*Convocation to become Master*

Formatos das PDUs

O formato das PDUs e uma breve descrição de seus campos é dada a seguir. PDUs com formatos idênticos são agrupadas. Imediatamente abaixo do formato existe uma indicação do

número de octetos que cada campo ocupa. Após cada formato há uma breve descrição dos campos.

SREJ



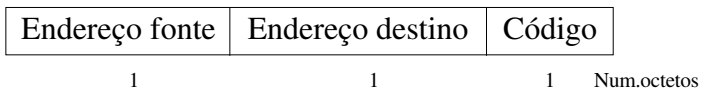
Endereço fonte - Endereço da estação que enviou o quadro;

Endereço destino - Endereço da estação ao qual se destina o quadro;

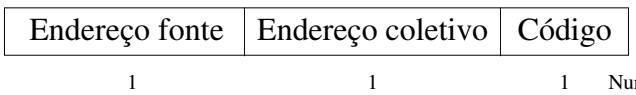
Código - Código que identifica o tipo de PDU;

Nr - Número do próximo quadro esperado. O último quadro recebido correto é o de número Nr - 1;

DISC, MPDISCAC, MPARMAC, MPRP, DISCAC, TP, CM, RP

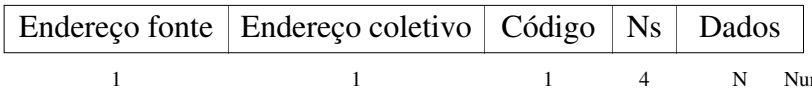


SMPARM, BM, MPDISC



Endereço coletivo - Endereço coletivo - o quadro será difundido para todas as estações. No caso de satélite é um endereço global e no caso de redes locais um endereço de grupo;

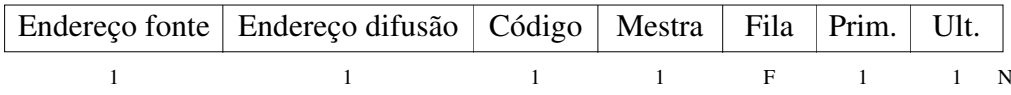
MPI



Ns - Número deste quadro. Usado em quadros numerados;

Dados - Dados do usuário.

TG



Mestra - Identidade da nova estação Mestra;

Fila - Fila com a identidade das demais estações que pediram o *token.* Esta fila é na verdade um array de "F" octetos, onde "F" é o número máximo de estações participantes. Neste array é formada uma fila circular que tem seu inicio e fim delimitados pelos campos "Prim." e "Ult.", ambos descritos a seguir.

Prim. - indexador do elemento do array que contém a identidade da primeira estação da fila de espera.

Ult. - indexador do elemento onde será inserida a identidade da próxima estação que pedir o *token*. O elemento anterior do array contém a identidade da estação que ocupa a última posição na fila. A fila está vazia quando os valores de "Prim." e "Ult." forem iguais.

PLIST

Endereço fonte	Endereço coletivo	Código	Ns	Lista	
1	1	1	4	L	Num.octetos

Lista - Lista com a identidade das estações que estão participando da conferência, onde L é o número máximo de estações que podem participar.

4. Especificação Formal

O uso de Técnicas de Descrição Formal (FDT - *Formal Description Techniques*) de protocolos permite a obtenção de uma especificação clara, concisa e sem ambigüidades. Além disso, as ferramenteas que normalmente estão associadas a uma FDT permitem a verificação, a simulação, o teste de forma automática e até a obtenção de uma implementação de forma automática a partir da especificação.

Dentre estas FDTs, destacam-se aquelas padronizadas pela ISO: **ESTELLE** (*Extended State Transition Language*) [11] e **LOTOS** (*Language of Temporal Ordering Specifications*) [12], que são dois formalismos usados para descrições formais de sistemas distribuídos e, em particular, de protocolos de comunicação, assim como dos serviços oferecidos por um protocolo.

Neste trabalho foi usado o formalismo definido pela linguagem ESTELLE. Esta escolha se deve à facilidade de modelagem da máquina de estados finita do protocolo, aliada à disponibilidade de ferramentas para ESTELLE desenvolvidas no GTA da UFRJ.

As ferramentas desenvolvidas [13] permitem desde um auxílio à edição da especificação até a validação com o uso de seqüências de testes geradas de forma automática. São elas:

- Editor de ESTELLE: permite a edição orientada de especificações ESTELLE, inclusive com recursos gráficos [14];
- Compilador [15] e Simulador ESTELLE [16]: estes dois módulos operam de forma integrada, permitindo a compilação de uma especificação ESTELLE para uma Forma

Intermediária (FI) em linguagem módula 2, que pode ser executada com intervenção do usuário. Existe ainda a geração de um relatório de acompanhamento (*Trace*) para o usuário, do qual constam cada transição disparada, na ordem cronológica em que ocorreu;

- Gerador de Seqüência de Testes [17]: permite a geração automática de seqüências de teste a partir da especificação dos estados e das transições de uma máquina de estados finita. Este gerador foi utilizado para gerar seqüências de primitivas do usuário do serviço proposto neste trabalho. As seqüências geradas foram utilizadas como entradas para o simulador ESTELLE.

4.1. Arquitetura da especificação

A arquitetura adotada para a especificação é mostrada na Figura 5. Esta arquitetura modela a troca de dados entre quatro usuários da camada de enlace (UA a UD). A troca de dados é controlada por quatro entidades Multiponto-a-Multiponto (ENL_MPMPa a ENL_MPMPd), conectadas através do meio de comunicação MACPHY.

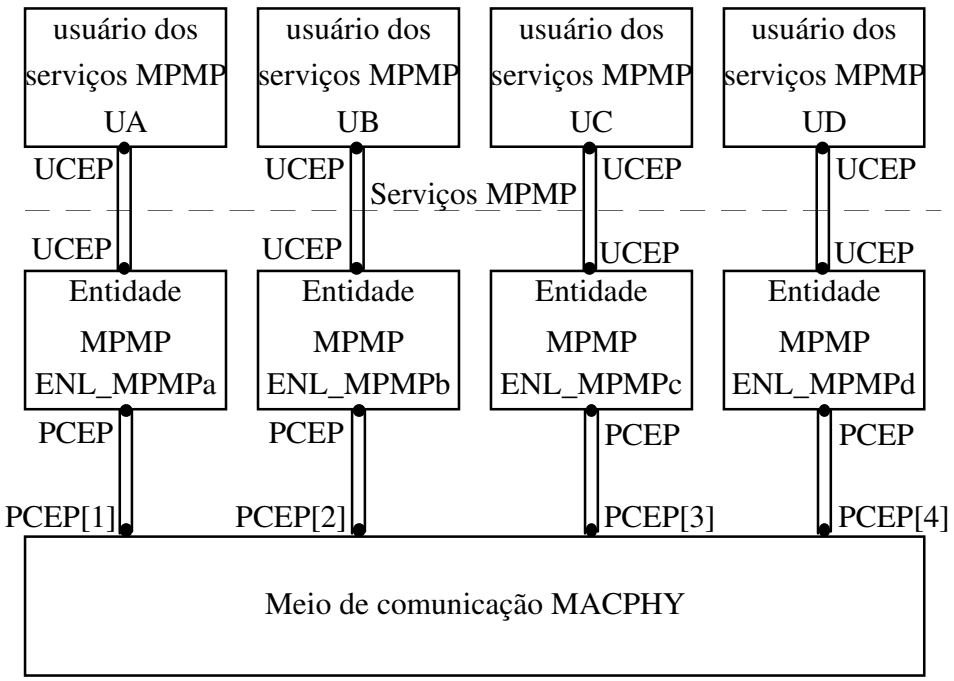


Figura 5 - Arquitetura da especificação do protocolo Multiponto-a-Multiponto.

A tradução da máquina de estados para a especificação formal foi realizada em conjunto com a simulação, ou seja, para cada grupo de primitivas afins o protocolo foi simulado e, se necessário, corrigido. Vale frisar a inserção de uma primitiva não constante da especificação do serviço: a primitiva DERRUBA. Esta primitiva foi inserida com a finalidade de tirar uma estação da conexão abruptamente, simulando um defeito grave ou uma eventual falta de luz. Esta primitiva permitiu avaliar o protocolo nestes casos de falha grave. Também foram inseridas transições que perdem quadros na entidade "meio de comunicação" (MACPHY),

para simular condições de erro. Realizando interações manuais com seqüências aleatórias de primitivas (sem uso do gerador de seqüência de teste), foram possíveis correções de alguns erros relacionados ao tratamento de exceções, como queda de estações durante a fase de transmissão de dados e, com isso, a inclusão de transições não previstas inicialmente, mas necessárias.

4.2. Testes com geração automática de seqüências

As seqüências de teste foram obtidas com o uso do Gerador de Seqüência de Testes do CAD de ESTELLE. O Gerador de seqüência de teste foi alimentado com uma máquina de estados criada a partir do serviço multiponto-a-multiponto (primitivas de serviço). Como resultado, foi gerado um relatório com diversas seqüências de envio/recebimento de primitivas de serviço. As seqüências de testes obtidas no Gerador foram inseridas, uma a uma, no simulador ESTELLE.

Através deste procedimento foi detectada a falta de duas transições existentes na máquina de estados do protocolo, mas que não haviam sido especificadas em ESTELLE. Estas transições são as que manipulam as primitivas MPDL_RELEASE_PLEASE.Request / MPDL_RELEASE_PLEASE.Indication e MPDL_INDIVIDUAL_RELEASE_PLEASE.Request / MPDL_INDIVIDUAL_RELEASE_PLEASE.Indication. A seguir são mostradas as duas seqüências de teste que detectaram a falha:

```
10 = r/N,MPDL_Establish_Req/MPDL_Establish_Conf,
     MPDL_Individual_Release_Please_Req/Null, Null/MPDL_Abort_Ind

11 = r/N,MPDL_Establish_Req/MPDL_Establish_Conf,
     MPDL_Release_Please_Req/Null, Null/MPDL_Abort_Ind
```

5. Implementação

O Modelo de Referência de Interconexão de Sistemas Abertos é arquitetado em camadas. Portanto, um sistema que segue este modelo tem como características básicas a definição de um certo número de serviços (primitivas) e um grande número de operações de adição/remoção de PCIs (*Protocol Control Information*) às/das SDU (relativo às informações trocadas entre entidades pares). O desempenho de um sistema como este é muito dependente da implementação [4].

O Grupo de Teleinformática e Automação (GTA) da UFRJ definiu uma arquitetura de implementação [5, 18, 19] que atende as exigências das normas internacionais e, ao mesmo tempo, privilegia o alto desempenho através da minimização da movimentação de dados e da carga de processamento. Além disto, esta arquitetura foi definida de forma a favorecer a modularidade e a migração para outras plataformas (código portátil).

Na definição de uma arquitetura de implementação para o modelo OSI, existem cinco aspectos que devem ser levados em consideração: os dois primeiros, diretamente relacionados

com o modelo OSI, são a estrutura de dados utilizada (interface entre camadas adjacentes ou entre camadas pares) e a forma de ativação dos serviços de cada camada (troca de primitivas). Os demais aspectos, que estão intrinsecamente ligados ao processamento do protocolo, são: a utilização da memória e seu gerenciamento, a maneira de ativar as diversas subrotinas e, a implementação dos temporizadores, fartamente utilizados nos protocolos que seguem o modelo OSI. O GTA definiu estruturas de dados para as interfaces entre camadas ("Estrutura_Primitiva" e a "Estrutura_Parâmetros"), módulos de gerência e funções dedicadas, a saber: gerência de temporizadores; gerência de tarefas (escalonador de tarefas) e funções de manutenção de interface entre camadas.

O serviço apresentado neste trabalho procura seguir os padrões definidos pela arquitetura de implementação definida pelo GTA. Algumas estruturas sofreram acréscimos para aumentar a velocidade de processamento, a exemplo da Lista de Quadros a Espera de Confirmação, que teve uma variável de contagem inserida de modo a não ser necessário percorrer toda a Lista a cada chegada de confirmação.

A implementação visa um ambiente de micro-computadores IBM-PC compatíveis, utilizando linguagem de programação C, partindo da especificação da máquina de estados e da especificação formal em Estelle.

Duas novas estruturas de dados foram criadas para gerenciamento de uma conexão. A Lista de Participantes e a Fila de Espera. A simplicidade e rapidez de acesso foram os alvos na escolha da implementação destas duas estruturas, para que seu gerenciamento não acarretasse nenhum aumento de processamento que pudesse ser relevante.

As rotinas referentes à implementação do Protocolo Multiponto-a-Multiponto foram colocadas em um único módulo de código, totalizando 4600 linhas de programação, na versão final. Pode-se dividir as rotinas em oito grupos funcionais. São eles:

1. Tratamento dos eventos gerados pelo usuário do serviço (Camada Rede);

2. Tratamento dos eventos gerados pelo provedor do serviço (Camada Física);

3. Tratamento dos temporizadores (*time-out*);

4. Montagem e envio de primitivas;

5. Montagem e envio de PDUs;

6. Gerenciamento das listas internas referentes a manutenção da qualidade da comunicação (Listas de Quadros Armazenados, Listas de Quadros Esperados e Listas de Quadros Recebidos Fora de Ordem) e controle de seqüenciamento, retransmissão e descarte de quadros recebidos com sucesso;

7. Gerenciamento da Lista de Participantes e da Fila de Espera e;

8. Rotinas gerais (iniciação, finalização, decodificação de PDU).

Os dois primeiros grupos citados referem-se à implementação da máquina de estados propriamente dita, sendo o primeiro relativo às ações de transmissão (sentido usuário para meio físico) e o segundo às ações de recepção. Para cada grupo utilizou-se um esquema com

declarações *switch ... case*, em dois níveis. O primeiro nível identifica o estado corrente da máquina de estados e chama a rotina de tratamento correspondente. Dentro da rotina chamada, uma nova declaração *switch ... case* determina a seqüência de operações com base na primitiva recebida (primeiro grupo) ou, com base na PDU recebida (segundo grupo) definindo assim, um ponto de interseção evento-estado.

As rotinas responsáveis pela integridade dos dados (grupo 6) são sorvedouras de uma parcela considerável do processamento do protocolo, principalmente em estações receptoras quando a comunicação está sujeita a elevadas taxas de erros. Estas rotinas foram, portanto, otimizadas para não se tornarem "gargalos" no processamento.

6. A Aplicação de texto-conferência

Esta aplicação foi idealizada para testar e avaliar o desempenho do Protocolo de Comunicações Multiponto-a-Multiponto desenvolvido [2]. Ela permite uma conferência através de texto entre até quatro estações. Para os usuários, esta aplicação assemelha-se ao *talk* do ambiente UNIX, mostrando-se útil como ferramenta que permite rápidas reuniões via texto, como se fosse um *talk* entre os vários usuários. Tendo sido desenvolvida para testar o protocolo, através desta aplicação é permitido ao usuário verificar e alterar parâmetros do protocolo, como os valores de contadores de tempo, a janela de emissão usada no controle de fluxo, a probabilidade de perda de quadros (simulação de erros de transmissão), etc. Além disso, a aplicação possui um algoritmo que permite o envio de uma seqüência de N quadros de tamanho T, com finalidade de testes de vazão, onde tanto N como T podem ser escolhidos pelo usuário.

Ao ser invocada através da linha de comando DOS, a aplicação desenvolvida divide a tela em quatro janelas, uma destinada a cada estação. Uma quinta janela é aberta por cima destas mostrando os valores atuais dos parâmetros do protocolo, com possibilidade de modificação de seus valores. As teclas chaves desta aplicação são algumas das teclas de função. A tecla F1 abre uma janela com um *Help* sensitivo ao contexto: nesta tela inicial, a tecla F1 abre uma nova uma janela com explicações sobre cada um dos parâmetros. Esta tela de edição de parâmetros pode ser aberta sempre que não houver conferência em andamento (conexão fechada) e, enquanto esta janela estiver aberta, a estação não responderá a pedidos de conexão, ou seja, não participará de conferências que estejam iniciando, podendo, porém, entrar posteriormente em uma conferência já em andamento.

As demais teclas de função permitem: estabelecer uma conexão (o que também é feito automaticamente se algo for digitado), pedir ou enviar o *token*, sair de uma conferência em andamento e encerrar uma conferência.

As condições da conferência são sinalizadas pela cor da linha título de cada janela: quando não há conferência em andamento, todas as janelas tem título em fundo preto, exceto a janela pertencente à estação local, que possui fundo azul. Ao ser iniciada uma conferência as estações que responderam ao pedido de conexão aparecem com o título em fundo vermelho. O usuário também fica sabendo quando seus dados estão sendo imediatamente transmitidos através da cor do título da janela de sua estação: na fase de transmissão o fundo é verde.

7. Conclusões

Este trabalho apresentou um serviço de comunicação multiponto-a-multiponto confiável. O projeto do serviço foi realizado segundo técnicas avançadas de projeto e implementação de protocolos que incluiram a especificação completa em linguagem formal e a simulação com seqüências de teste geradas automaticamente através de ferramentas de CAD-ESTELLE. O serviço e a aplicação apresentados foram implementados e testados em um ambiente real de trabalho.

O serviço oferece uma transferência confiável e transparente de dados dos usuários com facilidades de conexão e desconexão total ou individual. A gerência da conferência é feita por um algoritmo distribuído que utiliza um *token* de permissão de transmissão e uma lista de participantes como principais agentes de administração. O serviço apresentado é bastante robusto com previsão de perda de quadros, perda do *token* e de queda e retorno de estações. O serviço se mostou estável mesmo com elevadas taxas de erros no meio de transmissão (BER $> 10^{-3}$).

Foram realizadas medidas de vazão (bps) com 3 microcomputadores IBM-PC 286, sendo um com *clock* de 10 MHz e dois com *clock* de 12MHz, todos com placas de rede padrão Ethernet, compatíveis com a placa NE1000 (barramento de 8 bits). Nestas condições, pode-se citar o caso mais severo, onde as três estações possuem dados a transmitir: obteve-se uma vazão maior que 1,8 Mbps.

A aplicação de texto-conferência apresentada se mostrou bastante satisfatória. Outras facilidades podem ser incluidas no serviço para que este possa atender requisitos necessários às aplicações multimídias.

8. Referências

[1] R. C. Harkness, "Teleconferencing - A Key Subsystem in the Office-of-the-Future", *Office Automation Conference 1982 Digest*, San Francisco, abril de 1982.

[2] M. T. L. Vidal, "Especificação Formal, Verificação e Implementação de um Protocolo de Comunicação Multiponto-a-Multiponto e Aplicação de Texto-conferência", Tese de Mestrado, Univesidade Federal do Rio de Janeiro, PEL/COPPE/UFRJ, 192 p., Rio de Janeiro, outubro de 1994.

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

[4] M. T. L. Vidal, F. M. C. de Barros e O. C. Duarte, "Serviços de transferência de dados ponto a multiponto", *XI Simpósio Brasileiro de Redes de Computadores,* Campinas, pp. 97-116, maio de 1993.

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

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

[7] ISO/IEC 8802.3, *Information Processing Systems - Local Area Network Part 3: CSMA/CD Access Method and Physical Layers Specifications.* ISO/IEC, setembro de 1990.

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

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

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

[11] ISO TC 97/SC 21/WG1 - FDT/B IS9074, *ESTELLE - A Formal Description Technique Based on an Extended State Transition Model*, 1990.

[12] ISO IS8807, *LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour*, 1990.

[13] A. C. P. Pedroza, P. R. O. Valim, C. C. Goulart e R. C. Oliveira, "Um Sistema de Auxílio ao Projeto de Protocolos de Comunicação para Redes de Computadores." Anais do Seminário Franco-Brasileiro Sistemas Informáticos Distribuídos, setembro de 1989.

[14] C. C. Goulart e A. C. P. Pedroza, "Proposta de uma Forma Gráfica para a Linguagem ESTELLE", *VII Simpósio Brasileiro de Telecomunicações*, Florianópolis, 1989.

[15] R. C. de Oliveira Jr, "Um Compilador para a Linguagem ESTELLE", Tese de Mestrado, Univesidade Federal do Rio de Janeiro, PEL/COPPE/UFRJ, 135 p., Rio de Janeiro, agosto de 1991.

[16] P. R. O. Valim, "Um simulador de Protocolos de Comunicação para a Linguagem ESTELLE", Tese de Mestrado, Univesidade Federal do Rio de Janeiro, PEL/COPPE/UFRJ, 130 p., Rio de Janeiro, abril de 1990.

[17] G. S. Silva, A. C. P. Pedroza, "Um Sistema para Geração Automática de Seqüências de Teste para Protocolos de Comunicação"., VIII Simpósio Brasileiro de Redes de Computadores (VIII SBRC), 1990

[18] M. L. D. Lanza, "Especificação Formal e Implementação da Camada Enlace", Master's thesis, Univesidade Federal do Rio de Janeiro, PEL/COPPE/UFRJ, Rio de Janeiro, novembro de 1991.

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