Capítulo I

Introdução

Nos anos 60, a banda-passante dos meios de comunicação era um recurso caro e escasso em redes de longa distância. Especificamente, taxas de transmissão de centenas, no máximo milhares, de bites por segundo eram disponíveis. Além disso, o serviço oferecido pela rede era de baixa qualidade, com uma grande freqüência de erros de transmissão.

Nos últimos anos, com o desenvolvimento das tecnologias VLSI e de transmissão por fibra ótica, a velocidade das redes de comunicação aumentou significativamente, atingindo taxas de transmissão da ordem de 100 Mbps. Para os anos 90, espera-se que as redes de longa distância (WANs) alcancem velocidades de gigabites por segundo. O fator limitante tranferiu-se da banda-passante para o processamento do protocolo, que impede uma aplicação de utilizar uma fração razoável da banda disponível.

Além disso, as futuras Redes Digitais de Serviços Integrados em Banda Larga (RDSI-BL) deverão oferecer serviços que suportem uma vasta gama de aplicações: transferência de arquivos, transações cliente/servidor, datagrama, transferências multidestinatárias e aplicações de tempo real.

Para se atingir alto desempenho, várias estratégias têm sido propostas, tais como: arquiteturas de implementação eficazes, utilização de processamento paralelo [Kouf92] [Papa93] [Zitt90] e implementação em *hardware* do conjunto de protocolos.

Este trabalho apresenta uma proposta e a implementação de um protocolo multidestinatário estatisticamente confiável, configurável e com gerenciamento de acesso por fichas de permissão. Entretanto, uma das etapas fundamentais para se definir um novo protocolo é o estudo dos mecanismos de comunicação usados nos demais protocolos de alto desempenho citados na literatura especializada. Portanto, neste capítulo será feita uma análise de diferentes mecanismos necessários para o suporte dos seguintes serviços:

- **conexões**, incluindo os métodos de abertura, manutenção e liberação;

- **reconhecimentos**, usados pelo receptor para informar ao transmissor o sucesso ou fracasso da recepção correta dos dados;
- **controle de fluxo**, para compatibilizar as taxas de transmissão e recepção e, com isso, evitar congestionamentos;
- **controle de erros**, incluindo detecção e recuperação de dados que possam ser perdidos por erros de transmissão e transbordo de memória.

No Capítulo II, são apresentados vários protocolos que influenciaram o projeto do protocolo apresentado neste trabalho. O Capítulo III discute e compara os mecanismos que serão apresentados nas seções seguintes, bem como as razões que levaram à escolha de um ou outro em detrimento dos demais. Este capítulo também contém uma descrição do protocolo proposto, detalhando cada um de seus procedimentos. O Capítulo IV apresenta a arquitetura de implementação desenvolvida pelo Grupo de Teleinformática e Automação (GTA), bem como a modelagem do protocolo implementado no contexto desta arquitetura. O Capítulo V expõe os resultados das análises de desempenho realizadas, enfatizando medidas de vazão. Finalmente, conclusões e propostas para trabalhos futuros são apresentadas no Capítulo VI.

I.1 - Conexões

I.1.1 - Gerenciamento

De uma maneira geral, pode-se dizer que existe uma **conexão** entre duas entidades pares enquanto **informações de estado** suficientes para uma transferência confiável dos dados forem mantidas em ambas as entidades. Tais informações de estado são armazenadas em **descritores de conexão**, que consistem de um conjunto de características e variáveis relacionadas à comunicação entre entidades (por exemplo, identificadores, números de seqüência de transmissão e recepção).

Uma conexão entre duas entidades pares passa por três fases distintas:

- **estabelecimento** ou **abertura** da conexão, com a alocação e inicialização não ambígua das informações de estado em cada entidade. Para prevenir ambigüidade, algumas condições devem ser observadas [Wats81]: se nenhuma conexão existir, nenhuma mensagem de uma conexão previamente terminada deve causar a abertura de uma nova conexão; e se já existe uma conexão, nenhuma mensagem de alguma conexão anterior deve ser aceita na conexão atual. É ainda na fase de estabelecimento que alguns parâmetros, como modos de operação, recursos de memória ou CPU e tamanho de *buffers*, podem ser negociados para aumentar a eficiência da transferência de dados;

- **manutenção** da conexão ou fase de transferência de dados, durante a qual mudanças ocorrem nas informações de estado devido à recepção de dados ou reconhecimentos. Nesta fase, são trocadas também **informações de controle** entre as entidades, necessárias à **sincronização** entre os estados do transmissor e do receptor. Essa troca de informações de controle é fundamental à realização das funções básicas de um protocolo: reconhecimento de mensagens e controles de fluxo e erros. Todas as variáveis de controle a serem atualizadas devem ser incluídas em uma **mensagem de controle** sempre que uma troca de informações de controle for necessária. A utilização de uma mensagem exclusiva evita cabeçalhos com campos variáveis;

- **liberação**, **fechamento** ou **término** da conexão, em que as informações de estado são dealocadas também de maneira não ambígua. A ambigüidade nesta fase é evitada desde que as seguintes condições sejam atendidas [Wats81]: o receptor não pode fechar a conexão até que tenha recebido corretamente e reconhecido todas as possíveis retransmissões efetuadas pelo transmissor; analogamente, o transmissor não pode fechar a conexão até ter recebido todos os reconhecimentos pelos dados transmitidos ou tenha esperado, por um intervalo de tempo suficiente, o reconhecimento de sua última retransmissão.

As fases de estabelecimento e liberação da conexão são realizadas através de dois mecanismos principais: **negociação**, ou *handshake*, e gerenciamento **implícito** ou **baseado em temporizadores**.

- **Conexões negociadas**: neste mecanismo, uma troca de mensagens, normalmente específicas, é utilizada nas fases de abertura e fechamento da conexão. Uma abertura negociada evita problemas com mensagens duplicadas e permite um acordo prévio entre as entidades envolvidas a respeito de alguns parâmetros relevantes à conexão. Uma liberação negociada assegura a recepção completa e correta dos dados e que ambas as entidades estão preparadas para o fechamento da conexão. Porém, o gerenciamento por *handshake* introduz um *overhead* de processamento devido à essa troca de pacotes específicos e a novos estados que devem ser incorporados à máquina de estados do protocolo, além de atrasos. Esse *overhead* é entretanto perfeitamente aceitável desde que a duração das fases de estabelecimento e liberação sejam desprezíveis quando comparadas à duração da fase de transferência de dados e que o desempenho, como um todo, seja beneficiado pela pré-negociação de parâmetros. A Figura I.1 ilustra um exemplo de estabelecimento negociado, realizado pelo protocolo de Transporte Classe 4 da ISO [ISO8073]. Neste exemplo, é importante salientar que a conexão não é considerada aberta até que uma terceira

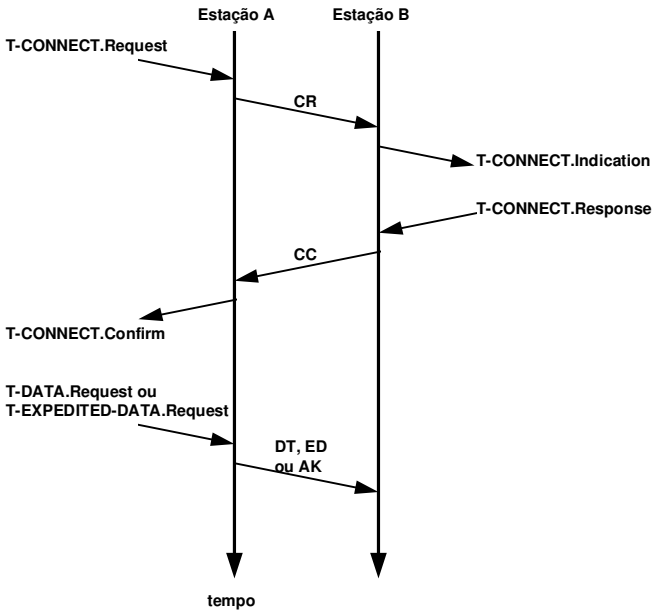

Figura I.1: Estabelecimento de conexão no TP4.

mensagem (a primeira TPDU DT, uma TPDU ED ou uma TPDU AK) tenha sido transmitida pelo iniciador (*three-way handshake*).

- **Conexões implícitas ou baseadas em temporizadores**: neste mecanismo, a conexão é automaticamente aberta mediante a recepção da primeira mensagem de dados. Temporizadores, em ambas as entidades, são utilizados para terminar a conexão também de forma automática. Por não haver mensagens específicas nem fases explícitas de estabelecimento e liberação de conexão, o gerenciamento da conexão é denominado **implícito**. Logo, a transferência de dados não precisa esperar pelo estabelecimento de uma conexão para ser iniciada, minimizando dessa forma a troca de mensagens. Num mecanismo baseado em temporizadores, o receptor não libera os descritores da conexão por um intervalo de tempo suficiente até que todas as mensagens duplicadas, incluindo retransmissões, tenham "morrido" na rede. Para isso, é imprescindível limitar o **tempo de vida** de uma mensagem, o que é feito através de um campo especial no cabeçalho da mensagem. Esse campo nada mais é do que um contador, inicializado com um certo valor máximo, baseado no tempo de ida e volta de uma mensagem sob pesadas condições de tráfego na rede e decrementado à medida que a mensagem atravessa a rede. Cada estação deve necessariamente ter a capacidade de receber todas as mensagens, decrementar seus campos de tempo de vida e devolvê-las para a rede. Entretanto, se após o decremento o campo de tempo de vida de uma mensagem atingir zero, a mensagem não é devolvida, sendo descartada pela estação. O transmissor, por sua vez, mantém os descritores da conexão associados a um temporizador cuja duração garanta a recepção de todos os possíveis reconhecimentos e a geração de números de seqüência e/ou identificadores de

conexão válidos. Conexões baseadas em temporizadores atendem as condições mencionadas anteriormente e garantem um gerenciamento livre de riscos [Wats87] [Wats89b].

I.1.2 - Tipos de conexão

No item anterior, os mecanismos de gerenciamento de conexão foram apresentados levando-se em consideração apenas conexões **ponto-a-ponto**, ou entre duas entidades pares. Além desse tipo de conexão, entretanto, existem três outras categorias diferentes:

- conexões **ponto-a-multiponto**: em um grupo de entidades, apenas uma pode operar como transmissor, enquanto todas as outras são receptores (*1*→*N*);

- conexões **multiponto-a-ponto** (ou **concentrações**): neste tipo de conexão, apenas uma entidade do grupo pode operar como receptor, aceitando mensagens (por exemplo, reconhecimentos) das demais entidades (*N*→*1*);

- conexões **multiponto-a-multiponto** (ou de conferência): todas as entidades do grupo podem operar como transmissores e receptores (*N*→*N*);

A Figura I.2 ilustra esses três tipos de conexão.

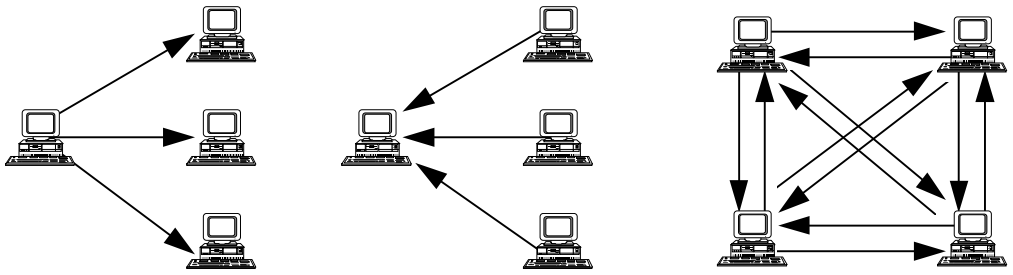

Figura I.2: Tipos de conexões para comunicações de grupo.

Doravante, as conexões ponto-a-multiponto e multiponto-a-multiponto serão chamadas de conexões **multidestinatárias**.

I.2 - Reconhecimentos

Um **reconhecimento (***acknowledgement* ou ACK) é uma indicação de que uma certa informação foi corretamente (ou não) recebida. Os reconhecimentos podem ser mensagens

específicas ou informações de controle constantes do cabeçalho de outros tipos de mensagem. Os reconhecimentos são normalmente associados aos mecanismos de controle de fluxo ou de erros, como será visto adiante. Entre as ações tomadas pelo transmissor mediante a recepção de um reconhecimento, estão a atualização de variáveis de estado, a liberação de dados que tenham sido porventura armazenados para retransmissão, ou o próprio disparo de retransmissões.

Existem basicamente dois mecanismos de geração de reconhecimentos: **submetidos ao transmissor** e **independentes do transmissor**.

- Geração de reconhecimentos **submetida ao transmissor**: na grande maioria dos protocolos, o receptor envia reconhecimentos de acordo com eventos implícitos ou explícitos desencadeados pelo transmissor. Um exemplo de evento implícito pode ser o reconhecimento de um certo número de mensagens; um evento explícito pode ser a utilização, pelo transmissor, de uma solicitação explícita de reconhecimento, através de uma mensagem específica ou de um campo no cabeçalho da mensagem de dados. Assim, o receptor somente enviará um reconhecimento quando for instruído neste sentido pelo transmissor.

- Geração de reconhecimentos **independente do transmissor**: reconhecimentos são gerados periodicamente pelo receptor, utilizando-se um temporizador, independentemente de qualquer ação do transmissor.

Uma vez que a freqüência dos reconhecimentos está diretamente relacionada à confiabilidade do protocolo, as conexões multidestinatárias podem receber diferentes graus de confiabilidade [Diot], a saber:

- conexões **não confiáveis**: o transmissor apenas difunde suas mensagens, sem se preocupar com a recepção pelos demais membros do grupo, ou seja, mensagens corrompidas ou perdidas não são retransmitidas ou recuperadas;

- conexões **atômicas** ou **totalmente confiáveis**: o transmissor deve retransmitir todas as mensagens que tenham sido porventura perdidas ou danificadas até ter a certeza de que **todos** os demais membros do grupo as tenham recebido corretamente. Isso implica, evidentemente, que o transmissor conheça *a priori* o tamanho do grupo e deva esperar necessariamente os reconhecimentos de todos os membros.

- conexões **k-confiáveis** ou **estatisticamente confiáveis**: considerando um grupo de N membros, a transmissão será considerada bem sucedida se nenhum ($k = 0$), alguns ($k < N$) ou todos ($k = N$) os membros do grupo receberem corretamente uma mensagem (ou um conjunto de mensagens). Os dados devem ser mantidos internamente em *buffers* durante um certo intervalo de tempo, ao fim do qual o transmissor decidirá (baseado nas informações acumuladas dos reconhecimentos recebidos) pela retransmissão ou não de uma determinada mensagem. Essa retransmissão, dependendo do protocolo que a implementa, pode ser feita ponto-a-ponto ou ponto-a-multiponto, conforme será visto no Capítulo II. Nesse tipo de conexão multidestinatária, normalmente é a aplicação que decide o grau de confiabilidade mais apropriado.

I.3 - Controle de fluxo

O desempenho de um sistema pode ser seriamente comprometido se a taxa com que o transmissor envia seus dados à rede subjacente não for compatível com a capacidade do receptor em processar e armazenar os dados recebidos. Além disso, ao se lidar com meios físicos de alta velocidade, a latência dos dados na rede aumenta significativamente. Logo, a taxa de transmissão não deve jamais exceder a capacidade da rede e a taxa com que a entidade destinatária recebe, processa e entrega os dados ao usuário, sob pena de causar congestionamentos e possíveis perdas de mensagens. O controle de fluxo destina-se exatamente a aplicar essas restrições ao transmissor, impedindo a ocorrência de tais problemas.

Há três mecanismos principais de controle de fluxo:

- Mecanismos de "**parada e espera**": o transmissor deve parar e esperar o reconhecimento após transmitir uma mensagem ou um grupo de mensagens.

- Controle de fluxo por **janela**: uma janela representa o número máximo de mensagens que podem ser transmitidas até que uma atualização de seus limites (inferior e superior) seja recebida em um reconhecimento. Neste caso, há duas abordagens: **janelas deslizantes** e sistema de **créditos**. Na primeira abordagem, uma janela de números de seqüência é mantida no transmissor e no receptor. No receptor, a janela representa o intervalo de números de seqüência das mensagens que podem ser aceitas, sendo que qualquer

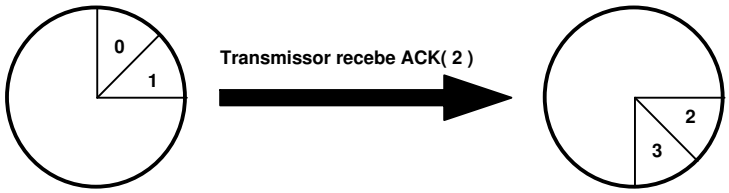

Figura I.3: Controle de fluxo por janela de tamanho fixo.

mensagem recebida com número de seqüência fora da janela deve ser descartada. É importante ressaltar que, neste mecanismo, o tamanho da janela é **fixo**; os reconhecimentos recebidos pelo transmissor simplesmente fazem a janela avançar sobre o espaço de números de seqüência, informando o número que servirá como limite inferior. A Figura I.3 exemplifica este método para um espaço de números de seqüência de módulo 8 e uma janela de valor 2. Inicialmente, a janela permite a transmissão das mensagens de números 0 e 1. Após a atualização recebida no ACK, a janela é avançada para os números 2 e 3. Na segunda abordagem, o sistema de créditos permite que a janela tenha tamanho **variável**. O reconhecimento informa então, além da atualização do limite inferior da janela, o parâmetro *crédito*, utilizado para determinar o limite superior. Dessa forma, se o receptor não tiver mais espaço em *buffer* e não puder receber mais uma mensagem sequer, ele pode efetivamente "fechar" a janela. Observe-se que, no método anterior, o limite superior da janela era sempre implícito, uma vez que seu tamanho era fixo. Um exemplo é mostrado na Figura I.4, em que o receptor aumenta a janela para três mensagens.

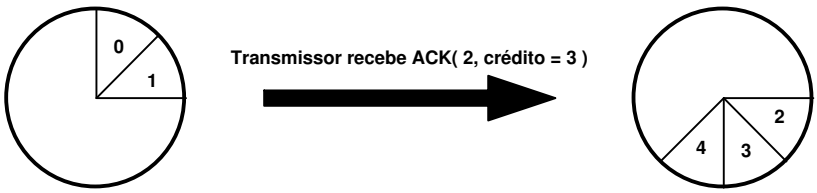

Figura I.4: Controle de fluxo por janela de tamanho variável.

- Controle de fluxo por **taxa de transmissão**: neste tipo de controle de fluxo, a taxa de transmissão, em bites por segundo, pode ser efetivamente controlada, ou limitada, pois existe um processamento mínimo necessário a ser seguido, limitando a vazão a um máximo. O sistema usualmente utiliza a taxa desejada, em octetos por segundo, e a **rajada**, ou seja, o número de octetos que podem ser transmitidos em uma unidade de tempo. Essa unidade de tempo é obtida dividindo-se a rajada pela taxa. Uma segunda alternativa consiste em se introduzir um **atraso** entre duas transmissões consecutivas (*interpacket gap*).

I.4 - Controle de erros

Dependendo da qualidade da rede subjacente, um protocolo deve implementar funções que garantam a integridade e o seqüenciamento das mensagens de uma transferência de dados. A **detecção** de erros se dá através de números de seqüência, campos de comprimento e *checksums*. Atribuindo-se **números de seqüência** a mensagens, é possível detectar-se perdas, duplicações ou entregas fora de ordem. Por exemplo, durante uma conexão, as mensagens cujos números de seqüência forem menores que o limite superior da janela de recepção são consideradas duplicatas e, conseqüentemente, descartadas. Internamente a uma mensagem, é

possível identificar perdas de dados comparando-se o número de octetos recebidos com um **campo de comprimento**, que pode estar localizado no cabeçalho ou na cauda de uma mensagem. Mas o método mais utilizado de verificação da integridade de uma mensagem é o *checksum*, que consiste em uma soma realizada sobre todo o comprimento de dados de uma mensagem. Na transmissão, essa soma é feita e colocada no cabeçalho (ou na cauda) da mensagem. Na recepção, o mesmo procedimento de soma deve ser realizado e o resultado comparado com o valor do campo *checksum*. Se forem diferentes, pode-se afirmar que os dados da mensagem foram corrompidos. Inclusive, em uma mesma mensagem, pode haver um *checksum* sobre os dados e outro sobre o cabeçalho.

O transmissor toma conhecimento de erros detectados pelo receptor através da recepção de **reconhecimentos negativos**, contendo o número de seqüência da última mensagem recebida corretamente (não fora de ordem e não corrompida) ou um intervalo de números de seqüência de mensagens perdidas (ou corrompidas), ou através de **temporização**, baseando-se em **reconhecimentos positivos** pelos quais o receptor reconhece apenas dados corretamente recebidos.

Entretanto, existe apenas uma maneira de se recuperar dados perdidos ou incorretos: a **retransmissão** dos dados. Existem dois mecanismos principais de disparo de retransmissões:

- **Retransmissão por temporização**: neste mecanismo, o transmissor deve manter um temporizador que estourará caso um determinado reconhecimento positivo não seja recebido, desencadeando a retransmissão das mensagens não reconhecidas. Este mecanismo recebe o nome de PAR (P*ositive* A*cknowledgement with* R*etransmission*).

- **Retransmissão por pedido do receptor**: neste mecanismo, as retransmissões são baseadas em informações de controle ou pedidos explícitos enviados pelo receptor (através de reconhecimentos negativos), sendo denominado ARQ (A*utomatic* R*epeat re*Q*uest*).

Quanto à retransmissão propriamente dita, pode ser por *go-back-N*, no qual o transmissor retorna ao número de seqüência imediatamente superior àquele da última mensagem corretamente recebida e retransmite todas as mensagens a partir daquele ponto, ou **seletiva**, em que o transmissor retransmite apenas as mensagens pertencentes a *gaps*, ou falhas, reportados. Entretanto, ambas as técnicas podem ser combinadas, como por exemplo em um mecanismo onde retransmissão seletiva seria usada na primeira retransmissão de uma mensagem e *go-back-N* seria usado nas retransmissões seguintes (da mesma mensagem) [Lima94a] [Lima94b] [Lima94c], pois a probabilidade da mensagem chegar com erro pela segunda vez é muito pequena.