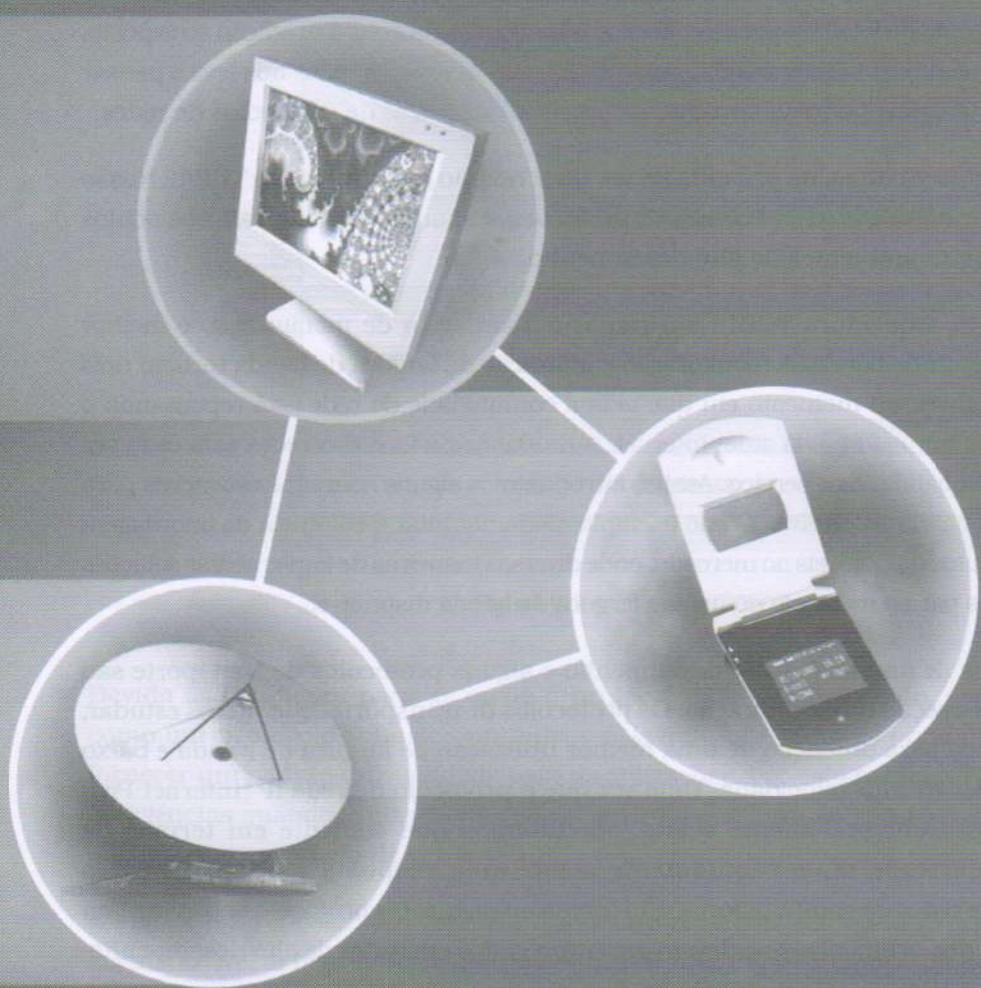


9

Capítulo

Protocolos de Alto Desempenho



Introdução

Neste capítulo, abordaremos os conceitos fundamentais para o conhecimento de um novo paradigma na área de redes de computadores denominado de *protocolos de alto desempenho*. Este novo paradigma objetiva não somente uma melhor utilização da infraestrutura das redes, mas também representa uma filosofia tecnológica voltada para a melhoria no desempenho computacional de aplicações de diferentes naturezas, quando executadas em ambientes distribuídos de rede.

A revolução contínua dos sistemas computacionais interligados através das redes tem auxiliado de forma positiva (e até então nunca imaginada) a sociedade moderna. A abrangência da utilização destes sistemas compreende um vasto espectro de áreas do conhecimento humano, tais como engenharia, biologia, medicina, educação, astronomia, música, física, química, economia e direito, entre centenas de milhares de outras áreas.

A evolução exponencial das tecnologias que compreendem a área de redes de computadores, que a cada dia nos permite a utilização de maiores larguras de bandas (*bandwidth*) para a transmissão da informação com menor retardo (*latency*), ilustra a necessidade de conhecermos com mais detalhes alguns conceitos importantes da área. Em adição ao exposto, o imenso universo de equipamentos que estão sendo conectados às redes nos leva a pensar como tais equipamentos poderão impactar as atuais arquiteturas de protocolos de redes.

Desta forma, apresentamos, na próxima seção, uma revisão da arquitetura TCP/IP. Nossa objetivo é ilustrar o contexto desta família de protocolos e apontar os principais obstáculos da arquitetura para ambientes de alto desempenho.

Após a revisão da arquitetura TCP/IP, ilustramos a abordagem de monitoração e melhor utilização da largura de banda. A monitoração e utilização da largura de banda tornam dois aspectos importantes, no momento em que toda a comunidade de rede está repensando o paradigma de melhor esforço da atual Internet. A mudança do foco das redes está cada vez mais orientada à qualidade de serviço. Assim, introduzimos alguns conceitos essenciais para o melhor uso da banda disponível. Como exemplo, podemos citar a eficiência da abordagem com alguns produtos disponíveis no mercado, onde diversas maneiras de implementar soluções são apresentadas para a melhoria no uso da largura de banda disponível.

Alguns exemplos de ambientes alto desempenho e alguns protocolos de transporte são mostrados após a seção de monitoração. Os protocolos de transporte, que vamos estudar, foram idealizados para obtenção de uma melhor utilização de largura de banda e baixo retardo nas redes de alta velocidade. Uma vez que o protocolo de rede IP (Internet Protocol) é o padrão universalmente aceito, acreditamos que somente em termos de transporte poderemos mais rapidamente achar uma resposta para melhorar o desempenho das redes de computadores. Em última análise, estes protocolos representam o estado da arte dos protocolos que devem ser empregados em redes de alto desempenho.

Finalizando este capítulo, apresentamos nossas conclusões e discutimos algumas possíveis direções futuras para os ambientes de alto desempenho.

Arquitetura TCP/IP

A Figura 9.1 ilustra a arquitetura TCP/IP e alguns dos seus principais protocolos. É relevante ressaltar a importância do protocolo da camada de rede, o IP. Este protocolo é o padrão do ambiente Internet.

Por outro lado, os protocolos de transporte, TCP (Transmission Control Protocol) e UDP (*User Datagram Protocol*), representam a forma pela qual as aplicações podem solicitar, respectivamente, serviços orientados e não-orientados à conexão. É importante recordarmos que ambos os protocolos de transporte da arquitetura TCP/IP foram projetados num passado onde as redes tinham baixa qualidade. Entendemos por baixa qualidade a constante ocorrência de erros (e sua alta freqüência) nas redes de comunicação devido às condições físicas do meio.

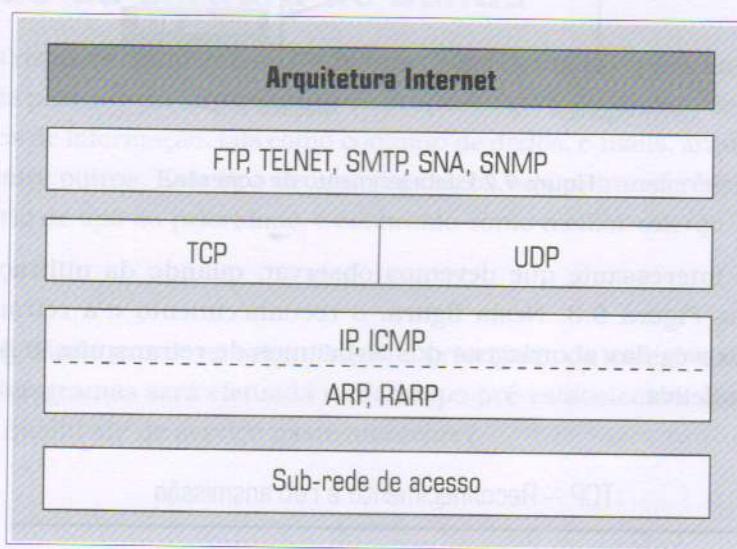


Figura 9.1 Arquitetura TCP/IP.

Devido às condições para as quais o protocolo TCP foi projetado, este foi idealizado como um protocolo robusto; em outras palavras, um protocolo que fim-a-fim deveria fornecer um padrão de qualidade para as aplicações. Por esta razão, o TCP tem algumas deficiências quando utilizado em redes com uma boa infra-estrutura para transmissão e baixíssima ocorrência de erros.

A Figura 9.2 representa o estabelecimento de uma conexão quando o protocolo TCP é acionado. Nesta figura, fica claro que o *handshake* inicial é composto pela troca de três

pacotes entre o destinatário e o remetente. Um exemplo que ilustra o alto custo desta abordagem é a troca de pequenos pacotes numa rede cuja rede de comunicação é baseada em satélite. Em outras palavras, estaremos utilizando uma largura de banda cujos custos são elevados de forma não otimizada.

TCP – Estabelecimento de conexão

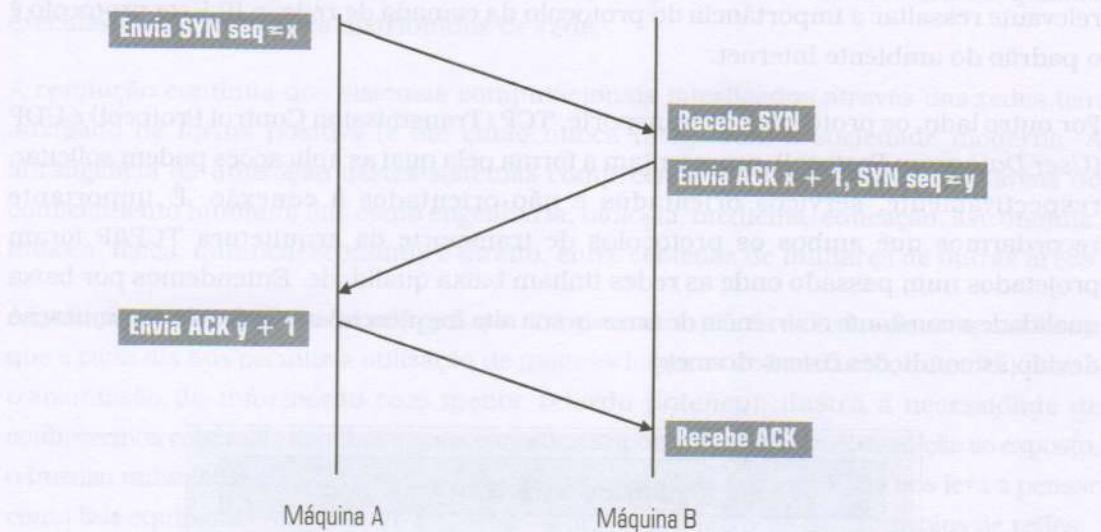


Figura 9.2 Estabelecimento de conexão.

Outro exemplo interessante que devemos observar, quando da utilização do TCP, é demonstrado na Figura 9.3. Nesta figura, o reconhecimento e a retransmissão são apresentados através das abordagens dos algoritmos de retransmissão go-back-n e da retransmissão seletiva.

TCP – Reconhecimento e retransmissão

Go-back-n x Retransmissão Seletiva

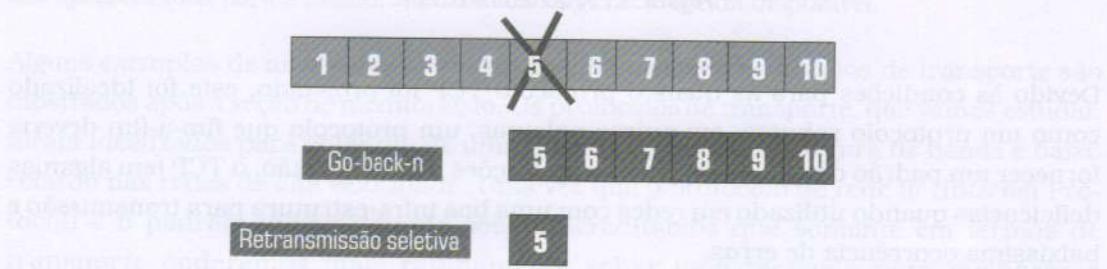


Figura 9.3 Paradigmas de retransmissão.

A retransmissão *go-back-n* é a forma implementada no protocolo TCP. Isto significa dizer que, quando uma conexão entre dois pontos quaisquer estiver em curso e quando um erro de recebimento for detectado (exemplo o pacote 5 da Figura 9.3), todos os pacotes a partir do pacote perdido serão retransmitidos. Neste procedimento, podemos detectar dois problemas. O primeiro é o desperdício desnecessário do uso da largura de banda para a transmissão dos pacotes que já haviam chegado de modo apropriado ao destinatário. O segundo problema é o desnecessário processamento dos pacotes pelo nó destinatário e o posterior descarte dos pacotes. Os dois exemplos de problemas do TCP indicam que devemos considerar soluções mais adequadas quando do uso de redes com alta taxa de transmissão, alto custo e baixa ocorrência de erros.

Na próxima seção, vamos abordar outros dois fatores importantes no sucesso num ambiente de rede: a monitoração e utilização da enlace.

Monitoração e Melhor Utilização da Largura de Banda

A maioria das redes de computadores (a Internet é um exemplo representativo da situação) é caracterizada pelo uso em larga escala e de propósito geral no sentido de transferência de diferentes tipos de informação, tais como conjunto de dados, e-mails, arquivos e aplicações multimídia, entre outros. Este tipo de abordagem, ou seja, a transferência de dados sem nenhum critério de tipo ou prioridade, é conhecido como melhor esforço.

Problema:

O modelo TCP/IP não promove, originalmente, prioridade de tráfego e não garante que a entrega dos datagramas será efetuada num tempo pré-estabelecido. Então, como obter determinada qualidade de serviço neste ambiente?

Solução (a):

Uma das possíveis soluções para o engarrafamento das informações que saem de uma LAN é a adoção de uma política de filtragem do tipo e volume de tráfego.

Solução (b):

Embora o IPv4 não tenha um conjunto de ferramentas (nativas) adequadas para a gerência da rede, a existência de ferramentas *IPv4 Network Management* auxiliam na monitoração e melhor utilização da largura de banda. Assim, vamos estudar alguns conceitos essenciais para o entendimento deste problema, e vamos apresentar, em adição, alguns exemplos de pacotes de software que podem nos auxiliar nesta tarefa.

Quality of Service (QoS)

Segundo Ferguson (1998), podemos definir qualidade de serviço (QoS) como: "Capacidade da rede de fornecer tratamento especial a certos tipos de tráfego previsivelmente".

Acreditamos que podemos definir, também, a qualidade de serviço como: "Termo empregado para definir os parâmetros específicos necessários para uma determinada aplicação do usuário. Estes parâmetros de serviço podem ser definidos em termos de largura de banda, latência e jitter, visando que a aplicação possa obter uma melhor qualidade ao longo da rede".

Para termos os requisitos de qualidade de serviço (QoS) atendidos, é necessário que tenhamos redes confiáveis. Assim, é vital a determinação das necessidades dos usuários e a geração dos parâmetros que assegurem o atendimento das aplicações dos usuários.

Protocolos de Reserva

Para que possamos garantir a manutenção da QoS numa rede IPv4, protocolos como o protocolo de reserva (exemplo: *RSVP – Resource reSerVation Protocol*) oferecem um conjunto de ferramentas para que, numa determinada rede, os roteadores trabalhem cooperativamente com mesmo objetivo: os parâmetros de qualidade.

IPv6 e QoS

A próxima geração de protocolo IP (IPng), ou seja, o IPv6, provê em seu datagrama alguns campos que permitem o gerenciamento da largura de banda. No IPv6, podemos ter qualidade de serviço como uma opção nativa do protocolo. Em outras palavras, não existe a necessidade de serem utilizados outros protocolos para prover QoS.

Gerência de Conexão

Quando, numa rede de computadores, existe a necessidade de interligação de locais remotos, nós nos deparamos com um problema de gerência da conexão. Este problema pode ser dividido de forma macro nos problemas administrativos (ou políticos) e nos tecnológicos.

Quanto aos problemas administrativos, é interessante relatar que:

- A falta de gerência de conexão, muitas vezes, leva o grupo de suporte de rede a apontar a responsabilidade pela falta como sendo da prestadora de serviço, dos equipamentos e enlaces.

- A efetiva gerência com o uso de certa limitação para alguns tipos de tráfego causam problemas para alguns usuários.
- A gerência de conexão é um jogo de perde e ganha entre usuários, posto que a largura de banda não é infinita e não tem custo zero.

Por outro lado, quanto aos problemas tecnológicos, podemos observar que:

- Para as aplicações de vídeo e áudio são desejáveis redes onde as interrupções sejam mínimas (embora estes serviços possam empregar uma rede como certo atraso).
- Para as implementações de redes de alto desempenho devemos prover serviços diferenciados para as aplicações diferentes. Um exemplo são as redes ATM, nas quais a relação de tempo comum para aplicações de áudio e vídeo é estabelecida a uma transferência constante de bits.
- A rede ATM também provê serviços de transferência a taxas diferenciadas e conexões orientadas e não-orientadas.

NOTA

Alguns profissionais consideram a gerência de largura de banda rede uma arte de magia negra. A razão para tal afirmação, muitas vezes, se baseia no fato da falta de uma coordenação global das redes para que o fluxo entre as mesmas possa ser parametrizado.

Os 8 bits do campo Service Type do datagrama IPv4 podem ser usados para prioridade. Redes ATM provêem QoS, nativamente, segundo a recomendação ITU 1350. Esta especificação serve como guia para que fabricantes e usuários usem um conjunto de parâmetros para redes de alto desempenho e para os diferentes tipos de classes de serviços. Nas redes IP que não têm originalmente tal facilidade, os administradores devem gerenciar manualmente os serviços e tipos de tráfego. É convencional a contratação do aumento da largura de banda. Canais dedicados e o uso de protocolos de reserva (exemplo: RSVP) permitem que os roteadores façam a alocação de banda devida para uma dada aplicação.

Problema:

O que fazer se o administrador da rede não tiver como especificar toda a QoS solicitada pela aplicação, pois a mesma passa pela Internet?

Solução:

Muito pouco poderá ser feito. Dentro da rede, o administrador pode empregar as políticas que quiser para atingir uma determinada QoS para determinada aplicação. Para os limites fora da sua rede e na Internet, mesmo sabendo de todos os parâmetros necessários, seus

esforços para obtenção da QoS poderão não ser bem sucedidos. Importante observar, por exemplo, a discrepância entre a largura de sua rede local (10, 100 ou até Gbps) para o enlace WAN (1, 2 ou até centenas de Mbps). Em adição, parâmetros como latência e jitter serão muito difíceis de serem negociados, uma vez que não há uma gerência unificada.

Ferramentas para Gerência de Largura de Banda num Ambiente IPv4

A versão 4 do IP tem poucas ferramentas (nativas) para a atribuição de prioridades para os pacotes IPv4. As ferramentas nativas do IPv4 são de pouca eficácia e os pacotes adicionais são, geralmente, empregados para a gerência dos pacotes IP.

Importante observar que as ferramentas existentes (um exemplo é o protocolo de reserva) para a gerência do IPv4 tem um custo alto e são muitas vezes complexas para redes pequenas. Estas são utilizadas em redes de grande escala que, usualmente, apresentam os maiores problemas no tocante à qualidade de serviço.

Resource Reservation Protocol (RSVP)

Este protocolo de reserva provê certo nível de controle sobre o fluxo de dados. O RSVP é baseado em roteamento (*routed-based*), permitindo aos roteadores fazerem solicitações a outros roteadores. O RSVP está em contínuo desenvolvimento no IETF (*Internet Engineer Task Force*).

Para maior eficiência da rede, todos os roteadores devem suportar o RSVP. O protocolo, quando faz uma solicitação, não tem garantia de atendimento, uma vez que a largura de banda já pode estar alocada para outro tráfego da rede que tem uma prioridade maior do que a do solicitante.

Os nós recebedores do protocolo RSVP podem fazer solicitações para outros roteadores, entre ele e o outro remetente, para o estabelecimento da reserva do serviço num sentido do fluxo de tráfego. Os roteadores recebedores de solicitações devem, com uma certa periodicidade, fazer reserva para garantir que os roteadores ao longo do caminho estejam cientes da reserva.

As solicitações RSVP são semelhantes aos mecanismos do ICMP usados pelo IP. À medida que uma solicitação vai passando pela rede, roteadores ao longo do caminho indicam quais os serviços que podem suportar. A solicitação também ajuda na determinação da MTU (Maximum Transfer Unit) que será empregada para o fluxo. As informações usualmente tratadas são:

- Token Bucket Rate and Token Bucket Size.
- Peak Data Rate and Minimum Policed Unit.
- Maximum Packet Size.

A solicitação RSVP está submetida às decisões de roteamento dos protocolos *Open Shortest Path First* (OSPF) e *Border Gateway Protocol* (BGP). Em outras palavras, primeiro é efetuado o algoritmo pelo qual deverá ser o caminho do pacote RSVP; só depois é que os pacotes poderão solicitar a qualidade de serviço necessária. Desta forma, o RSVP pode não representar um ganho para as redes que usam o IPv4.

Produtos de Gerência de Largura de Banda (Bandwidth Manager)

Vários produtos existem no mercado com o objetivo de auxiliar uma melhor gerência de utilização da largura de banda. Nesta seção, vamos apresentar alguns destes ambientes, que podem representar uma significativa melhora de desempenho para determinada rede.

Aponet Bandwidth Manager

As principais características do Aponet são:

- Prover canais de largura de banda disponíveis para um IP específico e uma dada porta, também provendo facilidade de monitoração de seu uso.
- O administrador pode atribuir limites de entrada e de saída para determinados limites de endereços (ou grupo de endereços).
- O pacote é bastante empregado por ISP (Internet Service Provider) para a atribuição dos limites de largura de banda de entrada e saída para seu usuários.
- Esse Bandwidth Manager vem em duas versões: a primeira, para menores quantidades de volume de informação (10 Mbps), e outra, para maiores (100 Mbps).
- Confiar no controle de fluxo do TCP para a gerência do tráfego, descartando pacotes quando ocorrer maior volume de dados.

IPATH Active Traffic Manager

Este gerenciador de largura de banda tem as seguintes facilidades:

- Dispositivo montado em rack de 10 e 100 Mbps.
- Em caso de falha, os dados passam automaticamente pelo dispositivo.
- A gerência pode ser efetuada por uma aplicação Web-based ou linha de comando.
- O produto permite a identificação de grupos de hosts e subnets para atribuição de alocação de bandwidth para o tráfego de entrada e saída para diversos protocolos.
- É permitido ao administrador auferir tipos de tráfego máximo e mínimo para um determinado critério de largura de banda.
- O produto usa uma aplicação compatível com SNMP para seu gerenciamento.

Packeteer Packetshaper

O Packeteer é caracterizado por:

- Ser um dispositivo montado em rack de 384 Kbps, 10 e 100 Mbps.
- Por sua classificação de tráfego poder ser baseada em URL.
- Prover o controle sobre uma variedade de protocolos, tipo TCP/IP, IPX, Appletalk, SNA e outros.
- Empregar o TCP Rate Control, ao invés do enfileiramento, evitando a perda de pacotes.
- Fazer uma estimativa da latência da rede e prever o tráfego que chega, ajustando a janela TCP para compensação.
- Fazer uma otimização no controle do envio de pacotes ACKs.
- Para o protocolo UDP, cria uma fila para ordenação de pacotes que chegam fora de seqüência.
- Suportar até 2.000 conexões TCP e 1.000 conexões UDP para os modelos de 10 Mbps e 384 Kbps.
- Suportar até 20.000 conexões TCP e 10.000 conexões UDP para o modelo de 100 Mbps.
- Oferecer integração com o HP OpenView e SNMP.

Checkpoint Floodgate-1

Alguns pontos interessantes deste gerenciador de largura de banda são:

- É um produto que complementa o Firewall-1.
- De forma semelhante ao pacote Firewall, o FloodGate-1 verifica quais as prioridades que os pacotes têm.
- Usando uma GUI (interface gráfica), o administrador pode determinar regras de tráfego aplicável para diferente hosts, destinos e tipos de tráfego.
- O administrador pode configurar até quatro tipos diferentes de tráfego, assim como permissões e exceções.
- Emprega diferentes ferramentas como GUI – para Windows 95, Windows NT e Solaris.
- Quando o tráfego está lento, continua a passar; mas quando ocorre um congestionamento existe um filtro pelo peso, baseado no peso relativo dos pacotes.

Sun Bandwidth Allocator

O pacote da SUN tem como facilidade:

- Prover um sistema de monitoração e gerenciamento mínimo em ambientes Sun Solaris.
- Classificar os pacotes por classes, e assim o tráfego é monitorado e gerenciado.
- Gerenciamento remoto por computador rodando Java na rede.
- Empregar SNMP para o gerenciamento e estatísticas.

Ukiahsoft Trafficware

Este ambiente provê um serviço diferenciado dos demais, sendo a capacidade do gerenciamento da largura de banda efetuada no computador do usuário (e o mesmo operando) e não apenas empregando o endereçamento IP.

O pacote executa em servidores Windows NT 4.0 e utiliza a informação de login criada no início da sessão do Windows para estabelecer uma prioridade de tráfego para a rede. Este pacote de software apresenta, também, semelhanças com os demais pacotes. A utilização de uma política na origem (baseada no endereçamento IP ou autenticação do usuário) e o tipo de tráfego (baseado em portas) são os parâmetros levados em consideração para o tráfego da rede. O Trafficware usa tanto filas, como controle de fluxo para controlar o fluxo TCP/IP. Existe uma GUI para efetuar o gerenciamento e emprega o protocolo SNMP.

Ferramentas para Gerência de Largura de Banda num Ambiente IPv6

Embora o Internet Protocol tenha se expandido de maneira nunca imaginada, não recebeu nenhuma mudança expressiva desde a RFC 791 de 1981. Todos estes anos causaram uma certa obsolescência no protocolo. Problemas como endereçamento, dificuldades de roteamento e problemas de segurança são alguns dos pontos críticos encontrados no IPv4. O IETF desenvolveu, pelas razões apresentadas, o IPv6.

O IPv6, ao contrário do IPv4, tem um conjunto de ferramentas que auxiliam de forma mais eficiente a qualidade de serviço (QoS) na rede. O cabeçalho do IPv6 dispõe de dois novos campos que auxiliam a gerência da QoS, o Flow Label e o Priority.

Flow Label

Este campo permite que os pacotes que devem ter um tratamento diferenciado sejam assim tratados. O campo tem tamanho de 20 bits, composto pelo endereço de origem e IP destino, permitindo que os roteadores mantenham o estado durante o fluxo ao invés de estimar a cada novo pacote. As aplicações são obrigadas a gerar um flow label a cada nova requisição. A reutilização do flow label é permitida quando um fluxo já está terminado ou foi fechado. A utilização de campo *flow label* provê aos roteadores uma maneira fácil de manter as conexões e manter o fluxo de tráfego numa mesma taxa.

Priority

A utilização do campo *priority* provê aos programas a facilidade de identificar os requerimentos de tráfego que estes necessitam. O uso efetivo ou normalização de como este campo, junto com o *flow label*, deve plenamente operar ainda está em discussão. O

campo de 8-bits destinado à classe está no momento em nível de desenvolvimento. Todavia, os 4 bits de prioridade ilustrados na Tabela 9.1 podem nos ajudar a imaginar o que poderemos ter pela frente.

Tabela 9.1: Prioridade no IPv6.

Valor	Descrição
0	Tráfego sem características.
1	Tráfego filler (fluxo contínuo de informação onde o tempo particularmente não interessa).
2	Transferência de dados sem supervisão (exemplo: e-mail).
3	Reservado.
4	Transferência grande de dados com supervisão (exemplo: HTTP, FTP e tráfego NFS).
5	Reservado.
6	Tráfego interativo (exemplo: telnet).
7	Tráfego de controle da Internet (informação usada por dispositivos que fazem parte da Internet, como roteadores, switches e dispositivos que empregam o SNMP para reportar estados).
8-15	Pacotes em processos que não podem controlar congestionamentos. Pacotes com valor 8 serão descartados antes dos de valor 15.

Protocolos e Ambientes de Alto Desempenho

Discutimos anteriormente, de forma geral e sucinta, a arquitetura TCP/IP e alguns dos problemas relacionados a este ambiente. Fica patente a obsolescência deste conjunto de protocolos para novas infra-estruturas de redes, onde a largura de banda cada vez maior e com latências cada vez menores requerem um melhor aproveitamento do meio.

Na seção de monitoração e melhor utilização da largura de banda, apresentamos alguns mecanismos que nos auxiliam numa melhor forma de uso da rede. Contudo, é importante observarmos que muitas das ferramentas existentes foram projetadas para as redes de comunicação e não abrangem redes locais com altas disponibilidades de largura de banda e baixo retardo. Em outras palavras, necessitamos que uma mudança em termos de protocolo sejaposta em prática.

Nesta seção, vamos discutir e apresentar alguns protocolos de transporte que foram especificamente projetados e implementados visando uma abordagem de alto desempenho. Deste modo, vamos apresentar algumas características importantes para os protocolos de transporte para alto desempenho. Depois, ilustraremos alguns protocolos que são

considerados aptos para atender os ambientes que solicitem ao desempenho, através de sua utilização direta ou modificação (um exemplo clássico é o TCP).

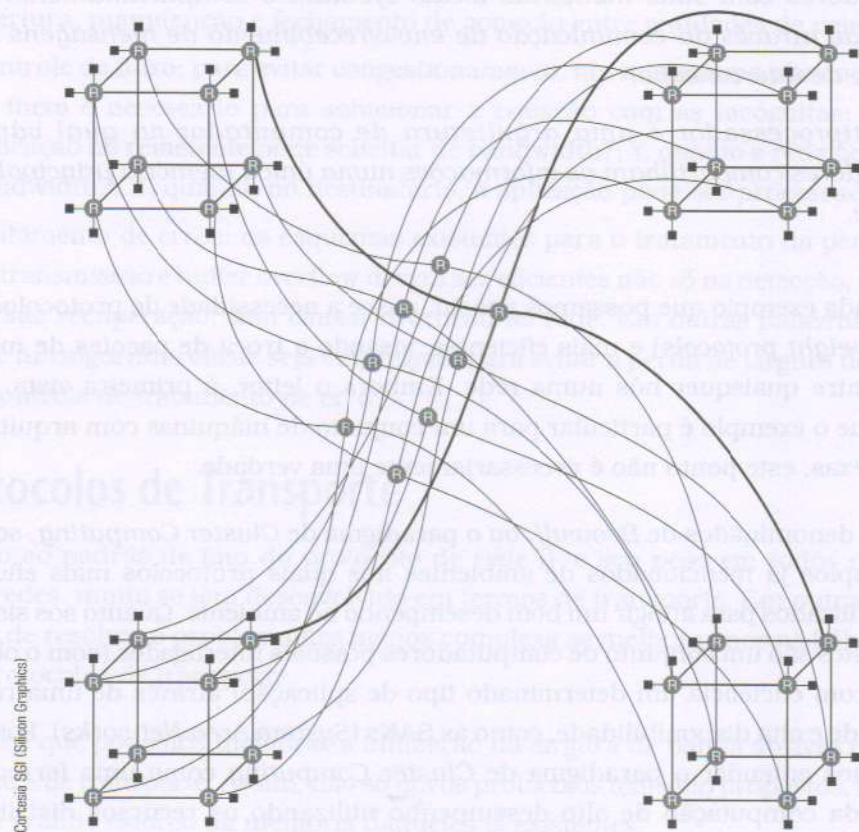


Figura 9.4 Topologia de um CRAY Origin 2000 com 128 processadores usando Cray Meta Router.

A Figura 9.4 exemplifica a necessidade de protocolos com uma abordagem diferente daquelas que eram consideradas até então. Centenas de processadores interligados através de uma rede de interconexão com um alto grau de confiabilidade. Nossa objetivo ao apresentar a figura é baseado no fato de que as arquiteturas dos *clusters* de computadores pessoais estão semelhantes aos computadores, com centenas e até milhares de computadores. Independente da abordagem, sejam estes multicomputadores ou multiprocessadores, no caso dos clusters, a rede de interconexão é representada pela tecnologia de rede local empregada na interligação das máquinas. Em outras palavras, devemos entender que a computação paralela distribuída é um problema também para a área de redes de computadores.

NOTA

Um multicomputador é uma arquitetura de computador na qual vários processadores com suas memórias locais efetuam o compartilhamento de informação através da comunicação de envio/recebimento de mensagens em uma rede de interconexão.

Um multiprocessador é uma arquitetura de computador na qual vários processadores compartilham as informações numa única memória principal do ambiente.

Assim, de cada exemplo que possamos pensar, nasce a necessidade de protocolos mais leves (*lightweight protocols*) e mais eficientes, visando a troca de pacotes de maneira otimizada entre quaisquer nós numa rede. Embora o leitor, à primeira vista, esteja pensando que o exemplo é particular para um conjunto de máquinas com arquiteturas mais complexas, este ponto não é necessariamente uma verdade.

Os sistemas denominados de *Beowulf*, ou o paradigma de *Cluster Computing*, somam-se aos exemplos já mencionados de ambientes nos quais protocolos mais eficientes devem ser utilizados para atingir um bom desempenho no ambiente. Quanto aos sistemas *Beowulfs*, estes são um conjunto de computadores pessoais interligados (com o objetivo de atender com eficiência um determinado tipo de aplicação) através de uma rede de alta velocidade e alta disponibilidade, como as SANs (*System Area Networks*). Por outro lado, podemos entender o paradigma de *Cluster Computing* como uma forma mais abrangente da computação de alto desempenho utilizando os recursos distribuídos disponíveis como se fossem um único ambiente computacional.

Em adição ao já exposto, a prática tem demonstrado que não existe uma correlação direta entre a melhoria da infra-estrutura das redes e um melhor desempenho das aplicações. Os protocolos têm sido o grande vilão deste problema. Protocolos de transporte, como o TCP, são inadequados para uma evolução das atuais configurações para os chamados ambientes de alto desempenho.

Protocolos e Melhoria de Desempenho

As três abordagens que se seguem podem melhorar o desempenho de uma determinada arquitetura de protocolo:

- Minimizar os requerimentos de processamento para redução dos custos de transmissão.
- Diminuir o controle de erros para redes que são consideradas sem erro (*error free*).
- Melhorar o algoritmo de controle de fluxo.

Novos protocolos, denominados de *lightweight protocols*, ou protocolos de alto desempenho, têm sido desenvolvidos visando:

- Melhoria na gerência da conexão: são os procedimentos de sinais necessários para a abertura, manutenção e fechamento de conexão entre entidades de comunicação.
- Controle de fluxo: para evitar congestionamento, um mecanismo eficiente de controle de fluxo é necessário para solucionar a equação com as incógnitas: X, quanto a aplicação no remetente pode solicitar de bandwidth?; Y, quanto a rede pode dispor de bandwidth?; Z, quando no destinatário, a aplicação pode ser processada?
- Tratamento de erros: os esquemas existentes para o tratamento da perda de dados na transmissão e buffer overflow devem ser eficientes não só na detecção, mas também na sua recuperação, sem causar overhead na rede. Em outras palavras, é desejável que um algoritmo eficaz seja empregado para evitar a perda de largura de banda para o controle de tratamento de erro.

Protocolos de Transporte

Devido ao padrão de fato do protocolo de rede IP e seu peso em todos os ambientes inter-redes, muito se tem desenvolvido em termos de transporte. Em outras palavras, a forma de resolver o problema fica menos complexa se melhorarmos as falhas existentes nos protocolos de transporte.

A idéia é que podemos melhorar a utilização da largura de banda através das conexões eficientes de transporte. Assim, não só novos protocolos têm sido propostos, mas também há um grande esforço na melhoria daqueles já existentes.

Exemplos de protocolos que foram projetados com o objetivo de alto desempenho são APPN, Datakit, Delta-t, NETBLT, OSI/TP4, VMTP e XTP.

APPN

O *Advanced Peer-to-Peer Networking* (APPN) é um protocolo de transporte da IBM para os sistemas S/36 e AS/400, construído para integrar a arquitetura de conexão fim-a-fim *System Area Network* (SNA). As funções de transporte foram implementadas num serviço orientado à conexão virtual baseado na total confiabilidade da conexão de enlace.

O APPN não dispõe de serviço de tratamento de mensagem. Assim, é um bom exemplo de protocolo que confia nos serviços de alta qualidade das camadas inferiores da rede.

Datakit

O protocolo foi desenvolvido como sendo um protocolo de transporte universal, independente de uma aplicação específica ou ambiente. O protocolo é baseado num serviço orientado à circuito virtual que entrega os pacotes sem erros e em seqüência,

considerando uma possível perda. A chave deste protocolo são as funções de controle fim-a-fim de controle de fluxo, detecção e retransmissão de dados perdidos.

Por observar que uns dos pontos de gargalo numa comunicação são os recebedores no seu processamento dos protocolos, o Datakit utiliza um protocolo denominado *Universal Receiver Protocol* (URP).

Na abordagem URP, somente são respondidos os comandos do transmissor que foram emitidos segundo os serviços oferecidos pelo protocolo de transporte. A unidade de transmissão é o byte; o nono bit é empregado para distinguir entre dados e sinal de controle.

Delta-t

O protocolo foi desenvolvido para prover uma comunicação eficiente no Lawrence Livermore Labs para serviços ponto-a-ponto orientados à grande quantidade de dados em redes não-orientadas à conexão.

A inovação deste protocolo foi o desenvolvimento de um sistema de gerenciamento de conexão baseado em tempo, através do qual o Delta-t suporta conexões *lightweight* com um mínimo de demora de sinalização.

NETBLT

O *NETwork BLock Transfer Protocol* (NETBLT) foi desenvolvido para a transferência de grandes quantidades de dados. O protocolo pode operar com eficiência em redes com grande latência, como redes de satélites empregando o IP.

A conexão NETBLT é unidirecional e é normalmente fechada pelo remetente. A unidade de transmissão é um largo buffer. A concorrência de diversas unidades de transmissão é o que mantém o fluxo num nível aceitável.

O controle de fluxo é efetuado na janela de transmissão através de parâmetros de tempo iniciais no intervalo de negociação. O tratamento de erros emprega uma retransmissão seletiva. Uma única solicitação de retransmissão pode ativar um número arbitrário de pacotes para serem retransmitidos. Ao final de uma transmissão, todos os NACK são enviados como um pacote único.

OSI/TP4

O TP4 é um protocolo desenvolvido sob coordenação da OSI, no qual as seguintes características interessantes existem:

- Grande número de parâmetros de QoS; entre estes existem throughput, percentual de erro, prioridade e demora de transmissão.
- Uma conexão de transporte pode ser dividida em várias conexões, ou seja, multiplexando a saída das solicitações.

VMTP

O *Versatile Message Transfer Protocol* (VMTP) foi desenvolvido para prover a infra-estrutura de comunicação para um dado sistema operacional distribuído. Em outras palavras, o foco principal do protocolo é o suporte a conexões orientadas a transações (exemplo: RPC). Estas aplicações requerem respostas rápidas para pequenas quantidades de dados.

Por outro lado, o VMTP também oferece serviço para transferência grande de dados. O controle de erro é seletivo e ainda tem uma facilidade de controle de transferência. O protocolo ainda provê um serviço de multicast.

XTP

O *Xpress Transport Protocol* (XTP) foi primeiramente desenvolvido com propósito para implementações VLSI. O protocolo foi projetado para atender, com eficiência, um grande espectro de serviços, tais como:

- Datagramas em tempo real.
- Multicasting.
- Transferência de grande quantidade de informações.

O XTP oferece em termos de controle:

- Controle de fluxo.
- Retransmissão seletiva.
- Estabelecimento de conexão explícita.

Análise das Funções do Protocolo de Transporte

Nesta seção, vamos examinar com mais detalhes os principais mecanismos do protocolo de transporte. Dentre estas funções temos:

- Gerência de conexão: estudo de início e término de uma associação de transporte.
- Fase de transferência de dados: recebimento de reconhecimento, controle de fluxo e tratamento do erro.

Gerência de Conexão

Visando compreender de maneira mais precisa como é efetivamente realizada a gerência de conexão, vamos estudar os aspectos descritos a seguir.

Sinalização

A troca de informação entre duas entidades de transporte com o propósito de gerência de conexão é conhecida como sinalização. A sinalização pode ser efetuada de maneira

que, dentro de uma mesma associação, tenhamos dados e informação. Denominamos este tipo de abordagem de *in-band*. Por outro lado, numa associação *out-of-band*, temos os dados e a informação de controle transmitidos separadamente. Considerando os protocolos de alto desempenho que foram apresentados e a taxonomia de sinalização, podemos fazer a seguinte consideração:

- In-band: TCP, NETBLT, XTP, OSI/TP4, Delta-t, VMTP.
- Out-of-band: Datakit, APPN.

O protocolo VMTP é híbrido quanto ao seu funcionamento, ou seja, efetua a verificação de conexão da forma *out-of-band*. Todavia, o estabelecimento da conexão é *in-band*. A consequência maior da sinalização *in-band* é que as entidades responsáveis pela conexão têm que resolver a cada pacote se existe, ou não, informação de controle. Este fato acarreta num aumento do processamento normal dos pacotes de dados, fato que não é desejável numa rede de alto desempenho.

De forma oposta, na sinalização *out-of-band*, temos uma desvinculação de dados e informação de controle, ocasionando num fator diferencial para redes que podem multiplexar a níveis mais baixos os pacotes. No caso da abordagem *out-of-band* é ainda permitido que diferentes tipos de dados sejam suportados na conexão. Um fator que aumenta a importância desta facilidade é a necessidade comercial de algumas aplicações de cobrança e segurança de uma única vez. A sinalização *out-of-band* parece ser a opção correta para as redes de alto desempenho, uma vez que o tempo de processamento de pacotes é reduzido.

Configuração Inicial e Fechamento

Em um estabelecimento de conexão, ou configuração inicial, e em seu fechamento estão condicionados dois mecanismos:

- Handshake: procedimento que requer explicitamente a troca de mensagens entre as entidades de comunicação.
- Implicit (ou Timer-based): neste tipo de esquema, a abertura de conexão é efetuada ao primeiro pacote recebido. O fechamento é consumado por intermédio de controle de tempo.

É importante observar que, no caso da conexão implícita, apenas no caso da abertura com garantia de controle de tempo o esquema funciona de modo apropriado. Em outras palavras, é necessário que pacotes que tenham um atraso não sejam confundidos com um fechamento. É necessário que a rede conheça os atrasos que estejam ocorrendo. Os protocolos podem ser assim classificados:

(a) Handshake

- Three-way: TCP, OSI/TP4 (setup), XTP (release).
- Two-way: NETBLT, Datakit, APPN, OSI/TP4 (release).

(b) Implicit: Delta-t, VMTP, XTP (setup).

Considerações:

- Para aplicações com granulosidade (quantidade de comunicação versus quantidade de computação) grossa, o handshake não é tão importante.
- O handshake pode ser atingido *out-of-band*; no caso do implicit, este é efetuando *in-band* quando o primeiro pacote chega.

Seleção do Serviço de Transporte

O serviço do transporte tem a responsabilidade na escolha do que prover para a aplicação dado uma determinada infra-estrutura de rede. Dependendo da rede, os serviços podem variar de maneira sensível. Os seguintes parâmetros devem ser considerados:

- Tamanho máximo de pacote.
- Valores de *timeout*.
- Contadores de tentativas.
- Tamanho de buffers.

Uma vez negociados, estes parâmetros podem ficar estaticamente estabelecidos durante a conexão. Outros parâmetros:

- Fluxo de controle.
- Número de seqüência.

Devem ser continuamente atualizados durante a transferência de dados, pelos algoritmos de controle de fluxo e sistema de recebimento de mensagem. Os seguintes parâmetros são considerados pelos protocolos:

- Parâmetros de negociação durante o estabelecimento de conexão: APPN, Datakit, NETBLT, OSI/TP4, TCP, VMTP e XTP.
- Atualização dos parâmetros durante a transferência de dados: Datakit, Delta-t, NETBLT, OSI/TP4, TCP, VMTP e XTP.
- Seleção dos modos de operação: Datakit, VMTP e XTP.

Multiplexação

A multiplexação é a combinação dos dados de mais de uma conexão em nível de protocolo para uma simples associação. A multiplexação é efetuada durante a fase de transferência de dados, todavia a facilidade é efetuada durante a fase de estabelecimento da conexão. Os protocolos podem ser classificados segundo a sua multiplexação. A multiplexação nas conexões na camada de transporte para um ponto da camada de rede é considerada um circuito virtual para as redes orientadas à conexão. No caso de uma rede não orientada à conexão, significa um par (endereço de origem e endereço destino).

Classificação dos protocolos:

- Fazem multiplexação: XTP, Delta-T, VMTP, OSI/TP4, TCP e NETBLT.
- Não fazem: APPN, Datakit.

Controle da Informação

O controle da informação é usado para o efetivo sincronismo de estado entre remetente e destinatário. Este serviço é vital para os controles de gerenciamento de conexão, recebimento de dados, reconhecimento na chegada de pacotes, fluxo de transmissão e tratamento de erro.

Exemplos são:

- XTP permite que destinatários apontem para lacunas nos dados recebidos. Desta forma, é possível a utilização do algoritmo seletivo de retransmissão.
- Uso excessivo de variáveis de controle deve ser evitado. Do protocolo HDLC, projetistas de protocolos de alto desempenho aprenderam que uma só variável para ACK e Windows não é interessante.
- Uso imediato de ACK muitas vezes leva à *síndrome da janela boba*.

Desempenho do Xpress Transport Protocol

Vamos focar nosso exemplo de desempenho no protocolo XTP (*Xpress Transport Protocol*). A razão para exemplificar através deste protocolo baseia-se no fato de termos utilizado largamente o mesmo no nosso Laboratório de Sistemas Integrados e Concorrentes (UnB/CIC-LAICO) para efeito de experimentos comparativos com o TCP.

A Figura 9.5 de XTP (1998) ilustra as possibilidades de operação do XTP. Embora possa oferecer os serviços encontrados no TCP e no UDP, além de outros, o XTP não tem como objetivo substituir estes dois protocolos. Da mesma forma que o TCP e UDP operam lado a lado sobre o IP, o XTP opera simultaneamente com qualquer outro protocolo de transporte. Algumas características interessantes do XTP quanto à utilização da camada de rede podem ser verificadas através dos seguintes pontos:

- O protocolo pode operar sobre o IP (o que o integra à Internet).
- Qualquer outro protocolo de rede é suportado.
- Pode trabalhar diretamente sobre a camada de enlace (abrangendo Ethernet, Token-ring e FDDI).
- O XTP pode, também, operar diretamente sobre a camada de adaptação em uma rede ATM.

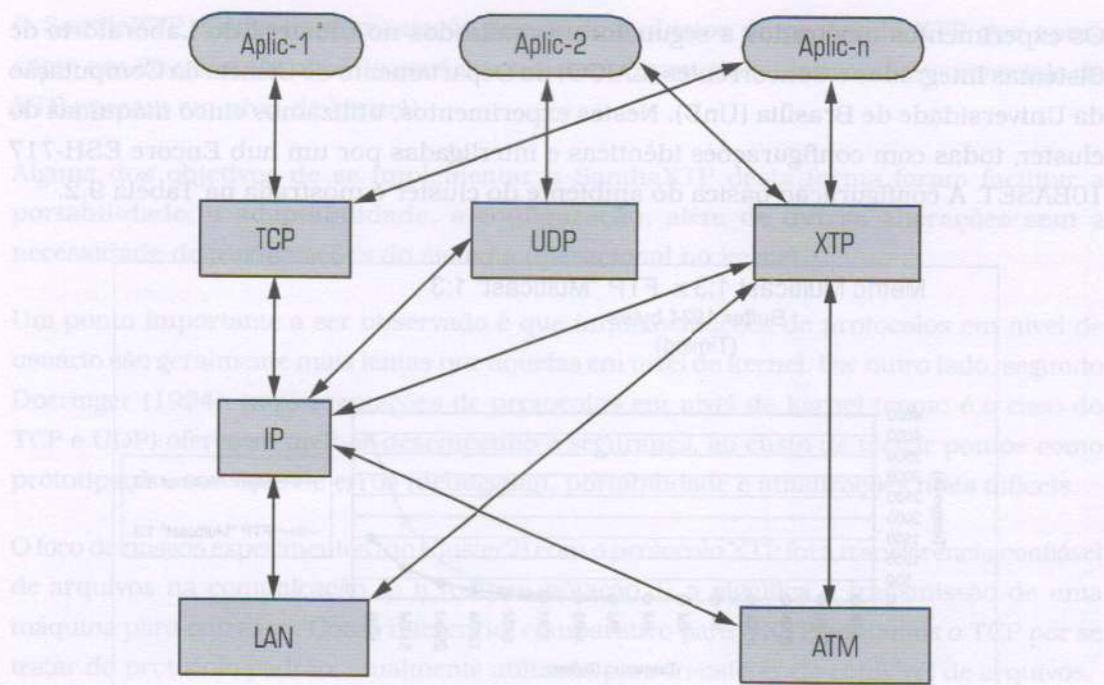


Figura 9.5 Possibilidades de operação do XTP.

Resultados Experimentais

Vamos apresentar, nesta seção, alguns resultados experimentais que foram efetuados em dois ambientes distintos. Em outras palavras, configuramos o protocolo em dois clusters semelhantes (Cluster1 e Cluster2) e com máquinas também semelhantes.

Nos gráficos que apresentamos nas figuras que seguem, ilustramos exemplos da comparação entre o protocolo XTP e o TCP (através do uso do protocolo de aplicação FTP). É importante observar, como mencionado por vários outros trabalhos, que é imperativo que tenhamos a disposição protocolos que acompanhem o melhor da infra-estrutura das redes. Desta forma, poderemos traduzir em ganho para as aplicações uma melhoria ora não existente.

A Figura 9.6 ilustra nossa primeira comparação, utilizando o Cluster1, entre o protocolo XTP (usando o protocolo de aplicação Metric) e o protocolo TCP (representado pelo multicast do FTP de uma para três máquinas). Semelhante a outros pesquisadores, provamos através de vários experimentos que o uso de um protocolo mais adequado pode representar um ganho efetivo para uma aplicação.

No caso da Figura 9.6 especificamente, simulamos uma aplicação que necessita o uso de um serviço de multicast. A facilidade de multicast é muito importante em aplicações distribuídas e paralelas. Nestas aplicações, um conjunto de processos deve se comunicar com uma certa freqüência, independente dos obstáculos que a rede represente para a granulosidade da aplicação.

Os experimentos mostrados a seguir foram realizados no Cluster2 do Laboratório de Sistemas Integrados e Concorrentes (LAICO) do Departamento de Ciência da Computação da Universidade de Brasília (UnB). Nestes experimentos, utilizamos cinco máquinas do cluster, todas com configurações idênticas e interligadas por um hub Encore ESH-717 10BASET. A configuração básica do ambiente do cluster é mostrada na Tabela 9.2.

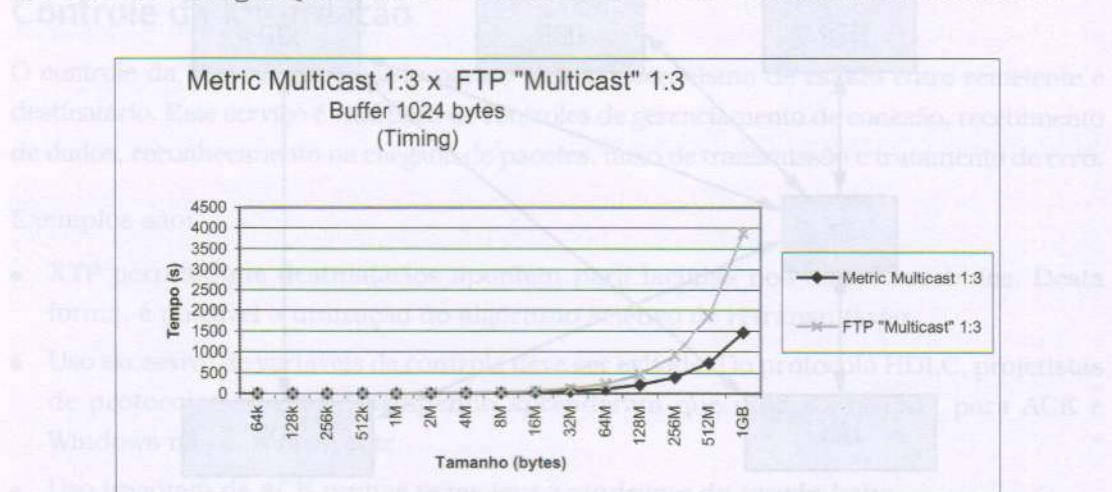


Figura 9.6 Comparação de uso do serviço Multicast

Tabela 9.2 Ambiente do cluster.

Processador	Intel Pentium II
Clock (MHz)	350
Memory (Mb)	32
Hub	Ethernet 10BASET
Sistema Operacional	Linux
Implementação do XTP	SandiaXTP 4.0

O ambiente de testes foi totalmente controlado, isolado de outras redes no momento dos testes. O ambiente foi utilizado apenas para tais propósitos, sem a interferência de qualquer outra transmissão no meio de comunicação (ou sobrecarga) nas máquinas devido a aplicações externas aos experimentos.

A implementação do XTP utilizada em nossos experimentos (assim como no primeiro experimento) foi o SandiaXTP. Este pacote é de domínio público e foi desenvolvido pelo Sandia National Laboratories (2001). Outras implementações comerciais existem e estão disponíveis nas referências Mentat (2001) e Network Xpress (2001).

O SandiaXTP é uma implementação orientada a objetos do protocolo XTP que opera como um *daemon* em nível de usuário no sistema operacional (as versões comerciais do XTP operam em nível de kernel).

Alguns dos objetivos de se implementar o SandiaXTP desta forma foram facilitar a portabilidade, a adaptabilidade, a configuração, além de outras alterações sem a necessidade de modificações do sistema operacional no kernel.

Um ponto importante a ser observado é que implementações de protocolos em nível de usuário são geralmente mais lentas que aquelas em nível de kernel. Por outro lado, segundo Doeringer (1994), implementações de protocolos em nível de kernel (como é o caso do TCP e UDP) oferecem melhor desempenho e segurança, ao custo de tornar pontos como prototipação, correção de erros (debugging), portabilidade e atualizações mais difíceis.

O foco de nossos experimentos (no Cluster2) com o protocolo XTP foi a transferência confiável de arquivos na comunicação 1: n (onde a notação 1: n significa a transmissão de uma máquina para outras n). Como referencial comparativo para o XTP, adotamos o TCP por se tratar do protocolo padrão, atualmente utilizado para transferência confiável de arquivos.

Ambos os protocolos foram submetidos aos mesmos testes, que consistiram basicamente na tarefa de transmitir quatro arquivos de tamanhos diferentes (1 Mb, 4 Mb, 16 Mb e 64 Mb) para dois, três e quatro receptores.

Como nosso principal interesse neste experimento foi avaliar o desempenho da comunicação multicast do XTP, de forma mais precisa, configuramos este protocolo de maneira idêntica ao TCP a fim de evitar que outras funcionalidades implementadas pelo XTP pudessem interferir nos resultados dos testes. Portanto, nos nossos experimentos, o XTP foi configurado para oferecer os mesmos serviços do TCP, ou seja, um serviço confiável, orientado à conexão, utilizando um controle de erros e de fluxo baseado em confirmações e janelas e, ainda, tendo como método de retransmissão o go-back-n.

Configurar o XTP para operar de maneira idêntica ao TCP nos permitiu de fato focar nossa análise nas características da comunicação multicast por ser este o único serviço extra oferecido pelo XTP.

Utilizamos duas aplicações com características semelhantes em nossos experimentos: o FTP para a transmissão com o TCP, e o Xfile para transmissão com o XTP. Ambas têm o *overhead* da manipulação, em disco, dos arquivos transmitidos, operaram com mesmo tamanho de buffer (4096) e transmitiram os bytes de cada arquivo tais como estes foram lidos, não sendo aplicado qualquer artifício (tal como compactação dos dados) que pudesse facilitar a transferência por um ou outro protocolo.

Os resultados de nossos experimentos, efetuados no Cluster2, são apresentados nas Figuras 9.7, 9.8 e 9.9. Cada uma destas figuras traz dois gráficos, um para as amostras de tempo e outro para as amostras de throughput tanto do TCP quanto do XTP.

Como o TCP não tem mecanismos de multicast, na comunicação 1: n são estabelecidas n conexões, uma para cada receptor. Desta forma, nos gráficos das Figuras 9.7, 9.8 e 9.9 apresentamos os resultados da transmissão multicast do XTP e os resultados de cada conexão TCP individualmente.

Visando facilitar a observação dos resultados do TCP nestes casos, denominaremos, de agora em diante, as máquinas do Cluster2 de c1, c2, c3, c4 e c5. Em todos os experimentos, utilizamos a máquina c1 como transmissor e como receptores sempre as mesmas máquinas tanto para o TCP quanto para o XTP.

Apresentaremos, inicialmente, os resultados para a comunicação 1:2 e, em seguida, para a comunicação 1:3 e 1:4. Ao final, faremos uma análise da escalabilidade do XTP em relação ao TCP.

Comunicação 1:2

Antes de analisarmos os resultados, é importante explicar os valores apresentados na Figura 9.7. Nossos testes avaliam o desempenho na transmissão confiável de arquivos de um transmissor a n receptores ($2 \leq n \leq 4$). Portanto, em nosso ponto de vista, o objetivo do teste é alcançado quando todos os bytes forem entregues aos receptores e o transmissor receber todas as confirmações. Neste instante é que devem ser amostrados os valores de tempo e throughput. Isto é o que ocorre no XTP multicast. Somente após o transmissor receber todas as confirmações de todos os receptores (cada receptor confirma seus dados individualmente) indicando que os dados foram recebidos corretamente é que são feitas as amostragens de tempo e throughput. Como podemos observar, o desempenho é determinado pelo receptor mais lento.

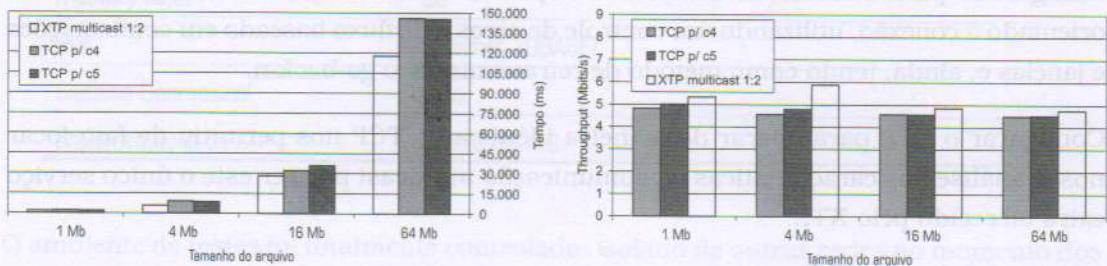


Figura 9.7 Comparação dos resultados de tempo e throughput na transferência de arquivos 1:2.

No caso do TCP, temos estabelecidas duas conexões distintas (uma para cada transmissão FTP), onde cada transmissão ocorre de forma independente (exceto pelo fato de as duas competirem pelo mesmo meio de transmissão no cluster), e os resultados de tempo e throughput são amostrados, também de forma independente, por conexão. Neste caso,

a princípio poderíamos adotar como resultado para comparação os valores da conexão TCP mais lenta, a fim de compatibilizar os resultados coletados com o XTP.

Entretanto, esta forma de análise poderia ser interpretada como uma maneira de distorcer os resultados em favor do XTP. Poderíamos também, adotar como critério a média dos resultados de cada conexão TCP. Entretanto, com este cálculo estariamos avaliando o quê? Visto que tanto a análise pelo pior caso, quanto pela média, apresentam problemas, mostramos em nossos resultados os valores coletados para cada conexão TCP juntamente com os valores da transmissão multicast do XTP. Desta forma, é possível analisar a comunicação 1:n do XTP em relação ao pior e ao melhor caso da comunicação 1:n do TCP. O critério de apresentar os valores de cada conexão TCP foi adotado em todos os testes de comunicação 1:n realizados.

Mesmo tendo a favor do TCP o fato de a aplicação que o utiliza (FTP) estabelecer suas conexões a priori, antes do envio de qualquer dado, e este tempo prévio de estabelecimento não ter sido amostrado em favor do TCP, e, ainda, o fato do TCP ser implementado em modo kernel, observamos na Figura 9.7 que o XTP obteve um desempenho melhor que o TCP em todas as transmissões na comunicação 1:2.

O bom desempenho do XTP na comunicação 1:2 deve-se basicamente ao multicast, visto que, devido à configuração do XTP em nossos testes, este é o único mecanismo presente no XTP e não encontrado no TCP. Apesar dos pontos desfavoráveis ao SandiaXTP (implementação em modo usuário, por exemplo), nosso experimento na comunicação 1:2 mostrou a eficiência da implementação de mecanismos multicast na camada de transporte.

Entre as características do multicast que contribuíram para o melhor desempenho do XTP sobre o TCP na comunicação 1:2 temos o fato que, devido ao multicast, é gerado apenas um único pacote, na camada de transporte, endereçado ao grupo multicast, independentemente do número de receptores (o TCP gera um pacote para cada receptor). Assim, de maneira simplificada, se para transmitir um arquivo de tamanho x são necessários y pacotes no nível de transporte para acomodá-lo, na comunicação 1:n o TCP gerará $y \cdot n$ pacotes, ou seja, y pacotes para cada receptor, enquanto o XTP gerará apenas y , independentemente do número de receptores.

Aliado aos mecanismos do XTP multicast, nosso ambiente de testes tem suporte para IP multicast, sendo o grupo de receptores local. Desta forma, o pacote gerado pelo XTP é encapsulado em um datagrama IP cujo campo endereço de destino tem o endereço IP classe D (endereço IP multicast) do grupo multicast (podemos observar que também na camada de rede é gerado apenas um único pacote). Este datagrama IP é encapsulado em um quadro Ethernet, cujo endereço Ethernet de destino é preenchido diretamente pelo mapeamento dos 23 bits menos significativos do endereço IP multicast nos 23 bits menos significativos do endereço Ethernet multicast 01.00.5E.00.00.00, conforme descrito por Deering (1989) (também, no nível de enlace é gerado apenas um quadro). Este quadro

Ethernet é transmitido pelo meio local (HUB, topologia em barra), onde cada receptor já conhece o endereço (tanto IP quanto Ethernet) do grupo multicast ao qual pertence. O quadro Ethernet multicast é coletado por cada receptor do grupo que o passa de camada em camada até que o dado chegue à aplicação (xFile). O menor número de pacotes gerado do XTP leva a uma melhor utilização do meio de transmissão, que implica diretamente em uma melhoria de desempenho na transmissão.

NOTA

O quadro Ethernet é transmitido em redes locais por broadcast.

Ligado ainda aos mecanismos de multicast e unicast, um fator que colabora para a degradação do desempenho do TCP em relação ao XTP é a questão da competição das conexões TCP pelo meio de transmissão. Em uma rede local Ethernet (que é o nosso caso), todas as máquinas competem pelo meio físico de transmissão utilizando como método de acesso o CSMA/CD (*Carrier-Sense Multiple Access with Collision Detection*). Portanto, como o TCP gera mais pacotes (não só em nível de transporte, mas também nas camadas inferiores) devido ao mecanismo de comunicação unicast, as chances de colisão entre os dados enviados pelo transmissor e as informações enviadas de volta pelos receptores aumentam à medida em que o número de receptores e a quantidade de dados a serem enviados aumentam. Pelo CSMA/CD, cada vez que uma colisão ocorre, cada máquina que tentou transmitir deve esperar por um intervalo aleatório de tempo antes de tentar transmitir novamente. Como nosso ambiente de testes é controlado, livre de qualquer transmissão que não esteja relacionada ao nosso trabalho, a única possibilidade de colisão é entre os receptores e o transmissor. Como o XTP, através dos mecanismos de multicast, gera menos pacotes no transmissor, a probabilidade de ocorrerem colisões diminui, diminuindo com isso a necessidade de retransmissões, o que leva a um melhor aproveitamento do meio de transmissão, melhorando o desempenho.

NOTA

A quantidade de dados a ser transmitida influencia da mesma forma as colisões no XTP.

É importante observar que as colisões ocorrem entre quadros de cada receptor e entre quadros dos receptores e do transmissor. Nunca ocorre entre os quadros gerados por cada conexão TCP no transmissor, pois ambas compartilham a mesma placa de rede para transmitir. Entretanto, o compartilhamento da placa de rede evita colisões entre os quadros gerados em cada conexão no transmissor, mas indica outro fator que prejudica o desempenho do TCP, o gargalo na placa de rede do transmissor. Como vimos, o multicast

do XTP gera apenas um pacote que, ao chegar na placa de rede, é transmitido sem nenhum problema, já que não existem outras transmissões em nosso ambiente de testes. No caso do TCP, cada conexão gera seus pacotes. Estes, ao chegarem à placa de rede, competem entre si, sendo que apenas um deles é transmitido de cada vez, enquanto os demais são enfileirados e aguardam sua vez de serem transmitidos. Da mesma forma que as colisões, este problema tende a prejudicar mais o desempenho do TCP à medida em que o número de receptores aumenta e a quantidade de dados a ser transmitida cresce.

A seguir, apresentaremos nossos experimentos para os dois protocolos na comunicação 1:3 e 1:4.

Comunicação 1:3 e 1:4

As Figuras 9.8 e 9.9 mostram os resultados amostrados em nossos experimentos na comunicação 1:3 e 1:4, respectivamente.

Pelos gráficos das Figuras 9.8 e 9.9, podemos observar mais uma vez a eficiência da transmissão multicast do XTP em relação ao unicast do TCP na comunicação 1:n.

O melhor desempenho do XTP é mais uma vez justificado pela utilização dos mecanismos de multicast. As mesmas considerações feitas anteriormente na comunicação 1:2 são pertinentes na análise da comunicação 1:3 e 1:4. Questões como o tipo de implementação, tanto do TCP quanto do SandiaXTP, o funcionamento do mecanismo de multicast, a quantidade de pacotes gerados, a questão das colisões, da competição pelo meio de transmissão e do gargalo das conexões TCP no transmissor foram amplamente discutidas na análise do teste anterior.

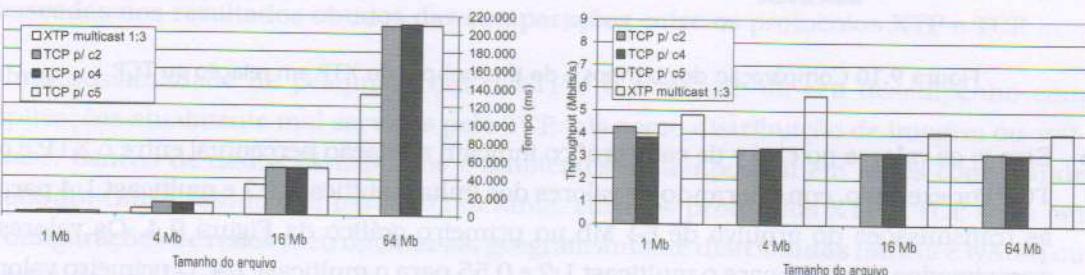


Figura 9.8 Comparação dos resultados de tempo e throughput na transferência de arquivos 1:3.

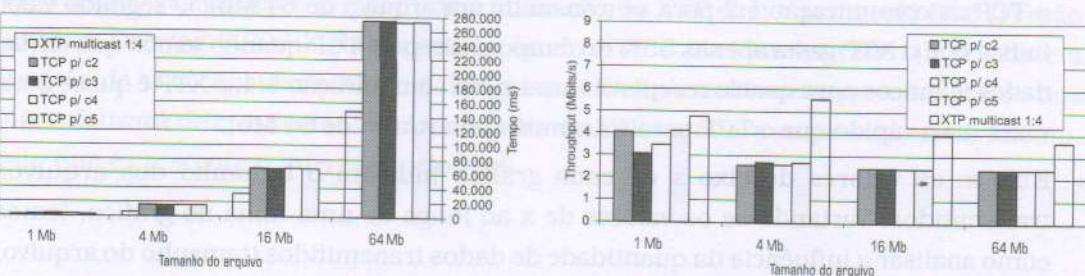


Figura 9.9 Comparação dos resultados de tempo e throughput na transferência de arquivos 1:4.

Após, apresentaremos isoladamente os resultados da comunicação 1:2, 1:3 e 1:4, e analisaremos a seguir o impacto do aumento do número de receptores no desempenho do XTP e do TCP.

Avaliação da Influência do Número de Receptores no Desempenho do XTP em Relação ao TCP

Agora, apresentaremos uma análise da escalabilidade do XTP em relação ao TCP no que diz respeito ao aumento do número de receptores. A Figura 9.10 apresenta a relação entre os resultados do XTP e os piores casos do TCP apresentados nas Figuras 9.7, 9.8 e 9.9.

A Figura 9.10 mostra o quanto (em porcentagem) o desempenho do XTP é melhor em relação ao TCP levando-se em consideração o tamanho do arquivo e o número de receptores. Os gráficos da Figura 9.10 são apresentados da seguinte forma:

- Linha: cada linha do gráfico representa a quantidade de receptores. Em cada gráfico, são apresentadas quatro linhas, sendo que três referem-se a comunicação multicast 1:2, 1:3 e 1:4 do XTP em relação ao pior caso do TCP e a quarta linha serve como referencial para as três primeiras.

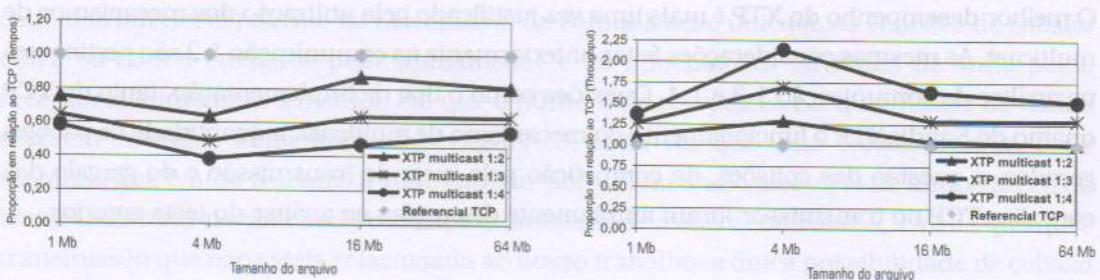


Figura 9.10 Comparação dos tempos e do throughput do XTP em relação ao TCP.

- Eixo y: os valores no eixo y de cada gráfico indicam a relação percentual entre o XTP e o TCP. Por exemplo, considerando os valores das linhas multicast 1:2 e multicast 1:4 para as transmissões do arquivo de 64 Mb no primeiro gráfico da Figura 9.4. Os valores aproximados são 0,80 para o multicast 1:2 e 0,55 para o multicast 1:4. O primeiro valor indica que o XTP multicast 1:2 gasta apenas 80% do tempo gasto pelo TCP para entregar um arquivo de 64 Mb a dois receptores (ou seja, o XTP é cerca de 20% mais rápido que o TCP na comunicação 1:2 para se transmitir um arquivo de 64 Mb). O segundo valor indica que o XTP gasta apenas 55% do tempo gasto pelo TCP quando temos que enviar dados idênticos para quatro receptores (assim, na comunicação 1:4, o XTP é quase duas vezes mais rápido que o TCP para transmitir um arquivo de 64 Mb).
- Eixo x: os valores do eixo x de cada gráfico indicam o tamanho dos arquivos transmitidos. Variando-se os valores de x ao longo de uma linha no gráfico, temos como analisar a influência da quantidade de dados transmitidos (tamanho do arquivo) no desempenho do XTP em relação ao TCP. Por exemplo, considerando os valores da

linha multicast 1:4 para a transmissão dos arquivos de 16 e 64 Mb no primeiro gráfico da Figura 9.4. Os valores aproximados são 0,45 para o arquivo de 16 Mb e 0,55 para o arquivo de 64 Mb. A análise isolada destes dois valores indica que na comunicação 1:4 o desempenho do XTP (relativamente ao TCP) é melhor para arquivos com 16M do que para arquivos com 64 Mb.

Tendo esclarecido os valores apresentados nos gráficos da Figura 9.10, podemos fazer uma análise breve de seus conteúdos.

Com relação ao número de receptores e o tempo, observamos no primeiro gráfico da Figura 9.10 que na comunicação 1:2, 1:3 e 1:4, à medida em que aumentamos o número de receptores, o XTP proporcionalmente apresenta tempos de transferência melhores que o TCP (quanto mais baixa a linha, menor o tempo gasto pelo XTP em relação ao TCP).

Com relação ao número de receptores e o throughput, no segundo gráfico da Figura 9.10, podemos observar que nas transmissões 1:2, 1:3 e 1:4, à medida em que o número de receptores cresce, o throughput do XTP melhora em relação ao throughput apresentado pelo TCP, para todos os tamanhos de arquivo (quanto mais alta a linha, melhor o throughput do XTP em relação ao TCP). Por exemplo, na transmissão do arquivo de 4 Mb para quatro receptores, o throughput do XTP chega a ser mais de duas vezes melhor que o apresentado pelo TCP para o mesmo número de receptores.

Direções Futuras e Comentários Sobre o XTP

Apresentamos inicialmente, nesta seção, algumas sugestões para direções futuras relacionadas ao XTP. Em adição, apresentaremos também nossas principais observações baseadas nos resultados obtidos das comparações entre os protocolos XTP e TCP.

Uma possibilidade de pesquisa com o XTP é a avaliação de seu desempenho com aplicações atualmente mal servidas pelo TCP, tais como distribuição de imagem ou software, bancos de dados distribuídos e ambientes de tempo real em redes com grande retardo. Outro estudo comparativo relevante, entre os protocolos XTP e TCP, seria em configurações de redes metropolitanas, geograficamente distribuídas (MANs e WANs) ou redes de satélite. Nestas configurações, seria interessante um estudo do desempenho considerando a retransmissão seletiva versus o algoritmo go-back-n.

Um trabalho relevante seria viabilizar a utilização do XTP com o protocolo IPv6 [Frazão (1997) e Deering (1998)]. Na especificação do XTP 4.0b [Sandia (2001)], já existe a previsão de se encapsular o XTP em datagramas IPv6. Acreditamos que a integração dos dois enfatizará determinadas funcionalidades do XTP e ajudará em seu desenvolvimento, visto que é crescente o interesse pelo IPv6.

Outro trabalho seria a implementação do protocolo em nível de kernel ou de projetos ligados às camadas superiores, como a implementação de novas aplicações e a integração de outras ao XTP, a fim de tirar proveito de suas funcionalidades.

Em nosso estudo, o XTP apresentou ser um protocolo flexível quanto aos mecanismos de controle, capaz de ser adaptado a vários ambientes. O XTP tem vários mecanismos, não encontrados no TCP ou UDP, que o tornam uma boa alternativa para determinados ambientes. Como exemplo podemos citar os mecanismos de retransmissão seletiva e confirmações negativas (FASTNAK), úteis para redes de satélite; os mecanismos de prioridade necessários em aplicações de tempo real; a comunicação multicast, útil para ambientes distribuídos de maneira geral, entre outros mecanismos que foram apresentados neste artigo e estão detalhados em Xpress (1998).

Nossos experimentos de avaliação de desempenho demonstraram que, a partir de dois receptores, mesmo implementado em modo usuário, o XTP através de seus mecanismos de comunicação multicast alcança melhores resultados que o TCP.

Outro fator ainda mais importante do que o XTP apresentar melhores resultados que o TCP na comunicação 1:2, 1:3 e 1:4, é que seus resultados melhoram proporcionalmente aos obtidos pelo TCP com o aumento do número de receptores. Este fato demonstra a melhor escalabilidade do XTP em relação ao TCP.

O protocolo XTP, devido a sua flexibilidade, pode ser uma boa alternativa ao protocolo TCP em uma série de ambientes, onde este último não tem atendido eficientemente aos requisitos necessários, como foi demonstrado no caso de nossos experimentos.

Ambientes

A procura por configurações com melhores desempenhos não é só uma preocupação concentrada no meio acadêmico. Nas referências [Mentat (2001), Network Xpress (2001) e Myricom (2001)], podemos encontrar um esforço da indústria já traduzido em produtos que auxiliam na implementação de ambientes de alto desempenho. A Figura 9.11 ilustra uma interface de uma rede SAN (System Area Network) denominada de Myrinet [Myricom (2001)]. Esta rede, baseada num sistema de interconexão de um computador paralelo, foi projetada para servir como um ambiente de alto desempenho para abordagens como os sistemas Beowulfs e para o paradigma de cluster computing.

A Interface Host Myrinet

Introduzimos a arquitetura da interface host Myrinet, que é baseada no conceito de interface de rede LANai.

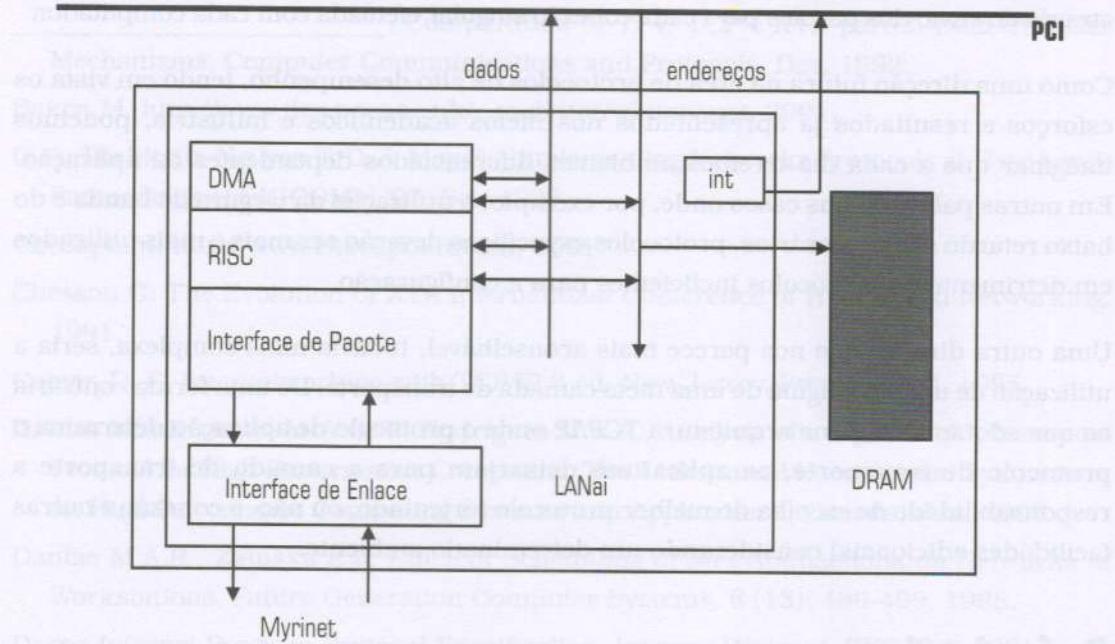


Figura 9.11 NIC Myrinet.

Conclusão

Neste capítulo, introduzimos de forma geral alguns conceitos dos protocolos de alto desempenho. Nosso objetivo foi focalizar alguns problemas existentes na arquitetura TCP/IP e indicar alguns protocolos que implementam abordagens mais eficientes para redes de alta velocidade e baixa latência. Em adição, vamos recomendar uma série de referências onde o leitor poderá aprofundar seu estudo na área.

Ilustramos como a monitoração e a melhor utilização da largura de banda podem ser atingidas com pacotes de software disponíveis no mercado. Nosso objetivo foi mostrar como atacar o problema de má utilização da infra-estrutura de rede.

Por outro lado, continuamos com o problema de desempenho de protocolos obsoletos para os ambientes onde uma grande quantidade de largura de banda e baixa latência estão presentes. Desta forma, apresentamos as principais facilidades de novos protocolos de transporte, que foram desenvolvidos sob o paradigma de alto desempenho. Em adição, mostramos uma comparação funcional entre um conjunto destes protocolos.

Finalizando a seção de protocolos de alto desempenho, mostramos, por intermédio de alguns exemplos do protocolo XTP quando comparado com o protocolo de transporte TCP, o uso mais eficaz do serviço multicast. É evidente que o protocolo XTP obteve um melhor

desempenho, uma vez que o serviço de multicast é nativo neste protocolo de transporte. Quanto ao TCP, que não dispõe do serviço de forma nativa, o serviço deve ser simulado através do envio dos pacotes para cada conexão singular efetuada com cada computador.

Como uma direção futura na área de protocolos de alto desempenho, tendo em vista os esforços e resultados já apresentados nos meios acadêmicos e indústria, podemos imaginar que a cada dia teremos ambientes diferenciados dependentes da aplicação. Em outras palavras, nos casos onde, por exemplo, a utilização da largura de banda e do baixo retardo são necessários, protocolos específicos deverão ser mais e mais utilizados em detrimento de protocolos ineficientes para a configuração.

Uma outra direção que nos parece mais aconselhável, todavia mais complexa, seria a utilização de um paradigma de uma meta camada de transporte. De uma forma contrária ao que adotamos hoje na arquitetura TCP/IP, onde o protocolo de aplicação determina o protocolo de transporte, os aplicativos deixariam para a camada de transporte a responsabilidade de escolha do melhor protocolo (orientado, ou não, à conexão e outras facilidades adicionais) considerando um determinado ambiente.

Referências

Quanto à literatura do TCP/IP são clássicas as referências Comer (1995), Darpa (1981), Tanenbaum (1996) e Postel (1980, 1981). O RSVP pode ainda não representar um ganho para as redes que usam o IPv4, assim o leitor deve consultar a referência IETF (2001) para um estudo mais detalhado sobre o assunto. Com relação ao IPv6, Frazão (1997), Deering (1989, 1998) e Lawton (2001). Quanto a QoS, é interessante ler Ferguson (1998).

Sobre os produtos de monitoração é aconselhável a leitura de Aponet (2001), IPATH (2001), Packeteer (2001), Checkpoint (2001), Sun (2001) e Ukiyahsoft (2001). Em Tantaway (1994), é apresentado um estudo de vários protocolos lightweight e suas principais características.

Quanto às referências sobre XTP, é interessante a leitura sobre o protocolo em Baguette (1992, 1992), Doeringer (1994), M.A.R (1998, 2000), Sandia (2001), Thekkath (1993), Xpress (1998) e Weaver (1999). Implementações comerciais podem ser encontradas em Mentat (2001) e Network Xpress (2001). Quanto às aplicações e o uso do protocolo XTP, o leitor pode encontrar em Dempsey (1994), Mecher (1996) e Strayer (1993, 1994).

Com relação aos ambientes Beowulfs e clusters, boas referências são Baker (2001), Sterling (1999) e Myricom (2001).

Bibliografia

Aponet, <http://www.aponet.com>, 2001.

- Baguette Y., Danthine A. Comparation of TP4, TCP e XTP, part-1: Connection Management Mechanisms, Computer Communications and Protocols, Set, 1992.
- _____, Comparation of TP4, TCP e XTP, part-2: Data Transfer Mechanisms, Computer Communications and Protocols, Dez, 1992.
- Baker, M. <http://www.des.port.ac.uk/~mab/tfcc/whitepaper>, 2001.
- C. A. Thekkath, Nguyen T. D. & Moy E., Implementing Networks Protocols at User Level, Proceedings of SIGCOMM '93, Set, 1993.
- Checkpoint, <http://www.checkpoint.com>, 2001.
- Chesson G. The Evolution of XTP, International Conference of High Speed Networking, 1991.
- Comer, D. E. Internetworking with TCP/IP, 3 ed. New Jersey, Prentice Hall, 1995.
- Dantas M.A.R., Lima M. V.G. R, Rodrigues M.R.A., Analysis of a Lightweight Transport Protocol for High-Performance Computing, The 14th Annual International Symposium on High Performance Computing Systems and Applications, Canada, Jun, 2000.
- Dantas M.A.R., Zaluska E.J. Efficient Scheduling of MPI Applications on Networks of Workstations, Future Generation Computer Systems, 6 (13): 489-499, 1998.
- Darpa Internet Program Protocol Specification, Internet Protocol, RFC, Set, 1981.
- Deering S. Host Extensions for IP Multicasting, Network Working Group, RFC 1112, August, 1989.
- _____, Hiden R., Internet Protocol – Version 6 (IPv6) Specification, The Internet Society – RFC 2460, 1998.
- Dempsey, B., Lucas, M., Weaver, A. High-Quality Video Distribution using XTP reliable multicast, Proceedings of the IWACA, Heidelberg, 1994.
- Doeringer W. A., ET AL. A Survey of Light-Weight Protocols for High-Speed Networks, High Performance Networks Technology and Protocols. Kluwer Academic, 1994.
- FERGUSON, P., HUSTON, G. Quality of Service, Wiley Computer, 1998.
- Frazão A. O que irá mudar com o IPv6, RNP News, Jul, 1997.
- IETF, <http://www.ietf.org/html.charters/rsvp-charter.html>, 2001.
- IPATH, <http://www.thestructure.com>, 2001.
- Lawton, G. Is IPv6 Finally Gaining Ground?, IEEE Computer, Ago, 2001.
- Mapp G. Preliminary Performance Evaluation of SandiaXTP on ATM, Second Workshop on Protocols for Multimedia Systems – PROMS'95, Cambridge, Outubro, 1995.
- Mechler R., Neufeld G. W., XTP Application Programming Interface, University of British Columbia, 1996.
- Mentat Inc, <http://www.mentat.com>, 2001.
- Myricom, <http://www.myri.com>, 2001.

- Network Xpress, <http://www.cs.virginia.edu/~acw/netx/>, 2001.
- Packeteer, <http://www.packeteer.com>, 2001.
- Postel J. Transmission Control Protocol, RFC 793, Set, 1981.
- Postel J. User Datagram Protocol, RFC793, Ago, 1980.
- S Gray, Cline R. E., Strayer W. T. An Object-Oriented Implementation of Xpress Transport Protocol, Proceedings of Second International Workshop on Advanced Communications and Applications for High-Speed Networks (IWACA), Set, 1994.
- Sandia National Laboratories California, <http://www.ca.sandia.gov>, 2001.
- Sterling, T. L., Salmon J., Becker D. J. e Savarese D. F , How to Build Beowulf – A Guide to Implementation and Application of PC Clusters. Massachusetts, Massachusetts Institute of Technology, 1999.
- Strayer W. T., Lewis M. J., Cline R. E. XTP as Transport Protocol for Distributed Parallel Processing. USENIX – Symposium on High-Speed Networking. California, 1- 3 de Agosto, 1994.
- _____, Weaver A. C. Is XTP Suitable for Distributed Real-Time Systems?, Department of Computer Science, University of Virginia, 1993.
- SUN, <http://www.usec.sun.com/software/band-allocator>, 2001.
- Tanenbaum, A S. Computer Networks, 3 ed. Prentice Hall, 1996.
- Tantaway, Ahmed N. High Performance Networks – Technology and Protocols, Kluwer Academic Publishers, 1994.
- Ukiahsoft, <http://www.ukiahsoft.com>, 2001.
- Weaver A. C. The Xpress Transport Protocol: Cluster Computing, Prentice Hall, 1999.
- Xpress Transport Protocol – Revision 4.0b, XTP Forum, Julho, 1998.

Bibliografia

Acesso: <http://www.scielo.br>, 2001.