A INFLUÊNCIA DO ALGORITMO DE NAGLE NO DESEMPENHO EM REDES IP/ATM

Paulo C. A. Antunes, Angelino C. Oliveira Walter Godoy Júnior, E.Nisenbaun
Banco do Brasil - Gerência Tecnologia, Software e Hardware
Brasília, DF
E-mail{pauloantunes,angelinoc}@bb.com.br
Centro Federal de Educação Tecnológica do Paraná - Centro de Pós Graduação em Engenharia Elétrica e Informática Industrial
Curitiba, PR
E-mail: {godoy,baun}@cpgei.cefetpr.br

RESUMO

Este artigo avalia o desempenho dos protocolos TCP/IP tendo como rede de suporte a tecnologia ATM. É analisada a influência do algoritmo de Nagle quando aplicado ao "IP Clássico e ARP sobre ATM". Os desempenhos teóricos foram calculados e comparados aos experimentos práticos. É mostrado que para aplicações no ambiente IP sobre ATM, quando se usa buffers de sockets desbalanceados no transmissor e receptor, ocorrem condições de bloqueio e degradam sensivelmente o desempenho. Também é analisado analiticamente e comparado aos experimentos práticos que o limitante superior de desempenho que se aproxima de 88% da capacidade de linha.

1. INTRODUÇÃO

O Modo de Transferência Assíncrono (ATM-*Asynchronous Transfer Mode*) é uma tecnologia projetada para integrar serviços de voz, vídeo e tráfego de dados de computadores, fornecendo individualmente a cada serviço e aplicação a qualidade de serviço(QoS-*Quality of Service*) requerida. O ATM é uma técnica orientada a conexão que usa pequenas unidades de tamanho fixo, denominadas células, no transporte de informações, o que permite a rápida comutação em *hardware*[1]. Devido a ampla utilização do protocolo de rede IP(*Internet Protocol*), dominante na interconexão de redes e que apresenta contínua expansão na rede mundial (Internet), analisaremos o ATM como estrutura principal de rede para o IP. Objetivando estudar os impactos de desempenhos que esta integração apresenta, neste artigo analisamos o modelo proposto pelo IETF "*The Classical IP and ARP over ATM*"[2]. A tecnologia ATM é visto como uma camada de enlace, da mesma forma como ocorre em redes Ethernet, Token Ring, FDDI, Frame-Relay, etc. O IP quando usa ATM, requer que seus pacotes de dados de tamanho variável, sejam adaptados, através de protocolo nas camadas de adaptação (AAL - ATM *Adaptation Layer*), em células ATM. Porém tal procedimento impõe *overhead* e custo de processamento. Tais características também são enfocadas neste trabalho.

Este artigo está organizado em seis seções. A segunda refere-se à topologia e equipamentos utilizados no experimento. Na terceira seção são abordados os fatores limitantes de desempenho na integração do protocolo IP em ambiente ATM. Na quarta é analisado o modelo proposto pelo IETF e ATM Forum e são quantificadas suas fontes de *overhead*. Na seção cinco são mostrados os experimentos realizados e faz uma revisão do Algoritmo de Nagle. Finalmente na seis são apresentadas as conclusões.

2. TOPOLOGIA DA REDE DE TESTES

Os testes foram efetuados nos laboratórios da Rede Metropolitana de Alta Velocidade(REMAV), em particular nas instituições: Centro Federal de Educação Tecnológica do Paraná(CEFET), Universidade Federal do Paraná(UFPR) e Universidade Católica do Paraná(PUC-PR), interligadas por cabo de fibras ópticas monomodo. A topologia da rede é mostrada na Fig. 1.

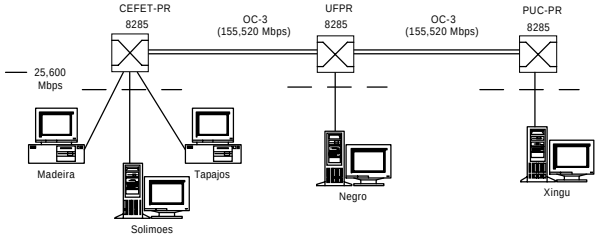

Figura 1: Topologia da rede

Utilizou-se comutadores (*switches*) ATM (IBM-8285) com *backplane* de 4,5 Gbps que usam a técnica de EPD(*Early Packet Discard*)[3]. EPD é a técnica que, em casos de congestionamentos, em vez de descartar células aleatórias de cada pacote, são descartadas todas as células pertencentes ao mesmo pacote com exceção da última célula, isto evita o desnecessário envio de células que não poderão ser remontadas no receptor. As interfaces de rede utilizadas possuem arquitetura PCI(*Peripheral Component Interconnect*) com capacidade nominal de 25,600 Mbps(Megabits por segundo). As estações de trabalho, CPUs, sistemas operacionais e interfaces de rede utilizadas são descritas na Tabela 1.

Estação	Processador//Sistema Operacional	Interface
Solimoes	Power PC 233 MHz, AIX 4.2.1	Turboways
Negro	Power PC 233 MHz, AIX 4.2.1	Turboways
Xingu	Power PC 233 MHz, AIX 4.2.1	Turboways
Madeira	Pentium 166 MHz, NT 4.0	Turboways
Tapajos	Pentium 166 MHz, Linux	ForeRunner

Tabela 1: Estações, sistemas operacionais e interfaces de rede utilizados

Na interface física de conexão local, entre a estação e o comutador, foi utilizado cabo de par trançado categoria 5 (UTP-5) com taxa de linha de 25,600 Mbps [4][5]. Na conexão remota da rede metropolitana (MAN), os comutadores foram interligados através de enlace óptico com fibras monomodo, com taxa de linha de 155,520 Mbps. Na camada física a interface de 25,600 Mbps não possui *overhead* de quadro(*frame*), são transmitidas células em fluxo direto e o delineamento entre elas é feito através do campo GFC(*Generic Flow Control*) contido na célula ATM[6]. As interfaces físicas entre comutadores foram configuradas para utilizarem quadros SONET (*Synchronous Optical Network*). A Fig. 2 [7] mostra um quadro SONET composto de três sinais STS-1 concatenados formando um sinal STS-3c/OC-3c. Cada quadro SONET se repete a cada 125 μs (8000 quadros por segundo) = 8000 x [9 x (261+9)] x 8 bits], fornecendo a taxa de transmissão de 155,520 Mbps.

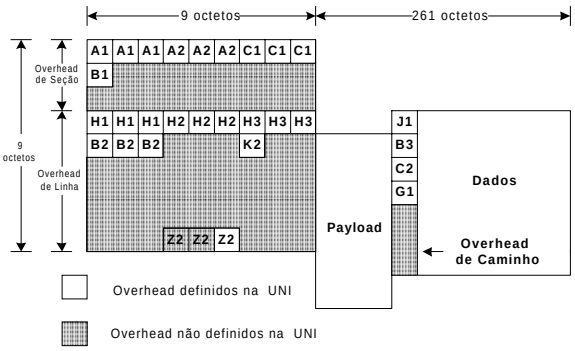

Figura 2: Quadro SONET STS-3c/OC-3c

O tamanho total do quadro SONET STS-3c/OC-3c é 2430 octetos (9 x(9 +261)). Dos quais 90 octetos são de *overhead*, divididos em :

- 27 octetos de *overhead* de seção;
- 54 octetos de *overhead* de linha;
- 9 octetos de *overhead* de caminho.

Nota:
1octeto= 1 *byte*

3. FATORES LIMITANTES DO DESEMPENHO

A aplicação, quando transfere dados para a rede, envolve a interação de protocolos em diversas camadas. Cada protocolo pode inserir informações de controle a esses dados para identificá-lo, seja na forma de cabeçalho (*header*), fecho(*trailer*) ou ambos. Estas informações aumentam a confiabilidade no transporte dos dados, como é o caso, por exemplo, do uso dos campos: HEC(*Header Error Control*) da célula ATM, CRC (*Cyclic Redundancy Check*) da AAL5 (ATM *Adaptation Layer 5*),soma de integridade dos dados (*checksum*) nos protocolos IP, TCP (*Transmission Control Protocol*) e UDP(*User Datagram Protocol*). As informações também identificam a origem e destino dos parceiros da comunicação, como é o caso: em ATM, das identificações do caminho e canal virtual das conexões(VPI-*Virtual Path Identifier*, VCI-*Virtual Channel Identifier*); em IP, do endereço origem e destino; em TCP e UDP, da indicação da porta de comunicação, tamanho da PDU (*Protocol Data Unit*). Estas informações adicionais são denominadas *overhead* de protocolo e é um dos fatores que limitam à aplicação ao tentar utilizar a total capacidade da linha.

Outras fontes, apesar de não serem consideradas *overhead*, influenciam de forma não linear, no desempenho e serão analisadas em conjunto, destacam-se:

- Arquitetura e velocidade da CPU;
- Arquitetura do barramento da estaçao de trabalho;
- Velocidade de acesso ao disco rígido;
- Interface física e de software de dispositivo ATM;
- Sistema operacional.

4. MODELO DO IP CLÁSSICO

O protocolo IP no modelo do IP Clássico utiliza o protocolo da camada de adaptação AAL5. O tamanho máximo permitido da PDU na CS(*Convergence Sublayer*) é de 65.535 octetos[8]. O formato dos quadros das camadas AAL5 e ATM é mostrado na Figura 3, onde CPCS é a parte comum na subcamada de convergência (*Common Part Convergence Sublayer*) e SAR é a subcamada de segmentação e remontagem (*Segmentation and Reassembly Sublayer*).

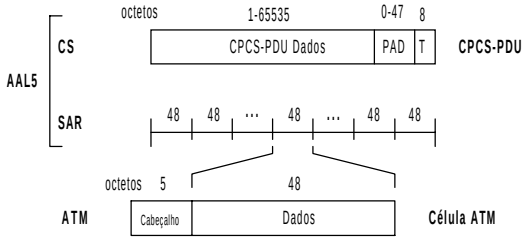

Figura 3: Quadros das camadas AAL5 e ATM

Na transmissão, a SAR segmenta a PDU em blocos de 48 octetos, nenhuma identificação de protocolo é inserida nesta subcamada. A SAR entrega os blocos, agora denominados ATM-SDU(ATM-*Service Data Unit*) à camada ATM. Essa camada insere o cabeçalho em cada PDU, agora denominada célula, marca a última unidade do quadro, e entrega a camada física para transmissão. Na recepção, a SAR extrai o bloco de dados(*payload*) das células, converte-as novamente em PDU e entrega ao protocolo superior.

4.1 IP Clássico sobre ATM

O valor genérico (*default*) da MTU (*Maximum Transmission Unit*) para o protocolo IP na camada AAL5 é de 9180 octetos[2], nos experimentos além desse valor, também foi usada MTU de 1500 octetos. A MTU indica a quantidade máxima de dados que podem ser inseridas em uma PDU sem que ocorra fragmentação. A Fig. 4 mostra o formato das PDUs nas camadas dos protocolos de transporte TCP e UDP para o modelo do IP Clássico sobre ATM, bem como a quantidade de *overhead* e o número correspondente de octetos e células.
Cada segmento TCP ou UDP tem relação direta com pacote (*packet*) IP, entretanto o protocolo de transporte pode incluir vários blocos de dados num único segmento[8].

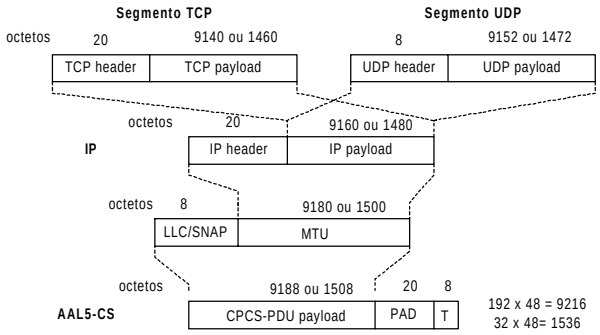

Figura 4: Formato das PDU para os protocolos TCP/UDP e IP sobre AAL5

4.2 Cálculos Teóricos da Banda Máxima Disponível para o Modelo do IP Clássico sobre ATM

A fim de avaliar os resultados obtidos experimentalmente, inicialmente foi calculada a banda teórica máxima disponível à aplicação abrangendo todas as camadas de protocolo. Nos cálculos foi considerado apenas o *overhead* imposto pela unidade de dado de protocolo em cada camada.

Através das expressões abaixo foram calculadas a banda máxima disponível a cada camada quando se usa MTU de 1500 octetos. Os resultados para as demais MTUs estão sumarizados na Tabela 2:

a) a banda disponível para a camada ATM entre comutadores

$$BD_{ATM} = \frac{Dados\_Quadro}{Tamanho\_Quadro}.Taxa\_Bits$$

$$BD_{ATM} = \frac{2430-90}{2430}.155{,}520.10^6 = 149{,}760 Mbits/s$$

b) a banda disponível para a camada de adaptação

$$BD_{AAL} = \frac{Dados\_Célula}{Tam\_Célula}.BD_{ATM}$$

$$BD_{AAL} = \frac{48}{53}.25{,}600.10^6 = 23{,}185 Mbits/s$$

c) a banda disponível para o protocolo IP

$$BD_{IP} = \frac{MTU\_IP}{CPCS\_PDU}.BD_{AAL}$$

$$BD_{IP(1500)} = \frac{1500}{1536}.23{,}185.10^6 = 22{,}642 Mbits/s$$

d) a banda disponível para os protocolos TCP

$$BD_{TCP} = \frac{Dados\_IP}{MTU\_IP}.BD_{IP}$$

$$BD_{TCP(1500)} = \frac{1480}{1500}.22{,}642.10^6 = 22{,}340 Mbits/s$$

e) a banda disponível para a aplicação, quando se usa o protocolo de transporte TCP

$$BD_{apl} = \frac{Dados\_TCP}{Dados\_IP}.BD_{TCP}$$

$$BD_{apl(1500)} = \frac{1460}{1480}.22{,}340.10^6 = 22{,}038 Mbits/s$$

A Tabela 2 sumariza os valores calculados.

Em Mbps	**SONET OC-3c**		**UTP – 25,600**	
Enlace	155,520		25,600	
ATM	149,760		25,600	
AAL	(*)		23,185	
	IP Clássico MTU(em octetos)			
	1500	**4060**	**6620**	**9180**
IP	22,642	23,071	23,004	23,094
TCP	22,340	22,957	22,934	23,044
Aplicação	22,038	22,843	22,864	22,994

Tabela 2: Banda máxima disponível a cada camada.
(*) Na interface NNI - SONET OC-3 não há camada de adaptação dos dados.

Na Tabela 2 foi mostrado o máximo desempenho teórico disponível a cada camada, porém ela não reflete o desempenho quando as aplicações usam mensagens de tamanho variável. Na Fig. 5 é mostrado esta variação. Pode ser observado nessa figura que para mensagens pequenas, o *overhead* é predominante na SDU. Quando se usa, por exemplo, mensagens de 112 octetos, são geradas 4 células(53x4=212 octetos) que correspondem a 89,28% de *overhead*, obtendo desempenho teórico de 13,524 Mbps. A Tabela 3 mostra a distribuição deste *overhead*.

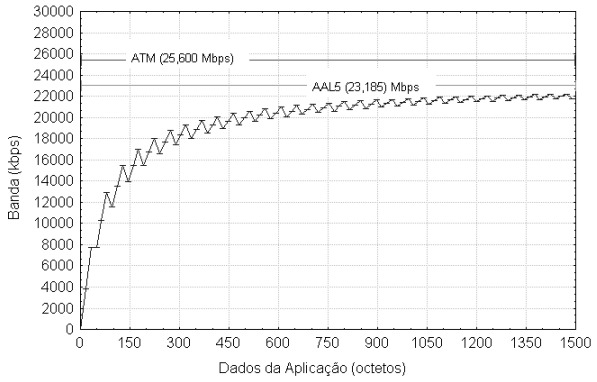

Figura 5: Desempenho Teórico para IP/ATM, baseado em mensagens recebidas da aplicação

O efeito escada(*ripple*) contido na Fig. 5 refere-se aos octetos de preenchimento necessários para tornar a PDU na AAL5 múltipla de 48 octetos.

Overhead Das Camadas	*Overhead* (octetos)	Acumulado (octetos)	*Overhead* Para à Aplicação (%)
Aplicação	0	112	0
Transporte - TCP	20	132	17,86
Rede - IP	20	152	35,71
LLC/SNAP	8	160	42,86
AAL5 (PAD)	32	192	71,43
ATM (4 células)	20	212	89,28

Tabela 3: Distribuição de overhead para mensagem da aplicação com 112 octetos

À medida que o tamanho da mensagem da aplicação aumenta o desempenho também aumenta, até atingir o valor máximo. O valor máximo é obtido quando a mensagem corresponde ao MSS(*Maximum Segment Size*) e gera o mínimo de octetos de preenchimento(PAD) na camada de adaptação (AAL5) para uma determinada MTU. Neste caso foi usada MTU de 1500 octetos. O valor encontrado foi de 22,038 Mbps para mensagens de 1460 octetos que tiveram 20 octetos de PAD, dando um *overhead* total de 16,16%, considerando todas as camadas (TCP,IP,LLC/SNAP,AAL5 e ATM).

5. MEDIÇÕES

Nas medidas de desempenho foi utilizado o programa de domínio público chamado Netperf[9], desenvolvido pela empresa Hewlett-Packard. O Netperf é composto por dois módulos, o módulo servidor e o módulo cliente. Emprega *sockets* na comunicação entre processos (IPC-*Interprocess Communication*) e entre máquinas e processos. Um *socket* estabelece a associação entre a fonte e o destino para um determinado protocolo, interface ou meio de comunicação e é identificado pelo endereço IP e número da porta de comunicação [10].

Os testes consistiram em enviar blocos de dados, de tamanho pré-fixado, continuamente. Para obter índice de confiabilidade de 99%(±3%), foram necessários vinte interações com períodos de quarenta segundos para cada variação individual de parâmetro. Devido ao ATM ser uma técnica orientada à conexão, antes das medições iniciarem, foram estabelecidas as SVC na LIS(*Logical IP Subnetwork*), necessárias para transportar o tráfego TCP/IP, entre os parceiros de comunicação. Dessa forma os resultados não serão influenciados pelo tempo despendido nessa operação.

5.1 Teste Local de Desempenho

O teste local avalia a capacidade de processamento da estação, interagindo com a pilha de protocolos. Consistiu em enviar e receber fluxos de dados tabulados na mesma porta de comunicação. Para estimar a capacidade de processamento, quando se usa o protocolo de transporte TCP, de acordo com a recomendação em [11], duplica-se o valor obtido na transmissão. A Tabela 5 mostra os resultados obtidos em cada estação. As estações foram configuradas com MTU de 1500 octetos e *buffers de sockets* de envio e recepção de tamanhos iguais a 4096 octetos. Estes parâmetros correspondem à maior carga imposta de processamento nos testes analisados.

As medidas foram colhidas no transmissor. O ambiente foi controlado, não houve disputa pela banda, uma vez que o modelo analisado utiliza tráfego UBR (*Unspecified Bit Rate*) e em caso de congestionamento na rede tem preferência no descarte de células.

Estação	TX = RX (Mbps)	Capacidade Estimada (Mbps)
Solimoes	115,567	231,124
Negro	114,231	228,462
Xingu	115,584	231,168
Madeira	41,480	82,960
Tapajos	111,782	223,564

Tabela 5: Capacidade das estações de trabalho

5.2 Desempenho em Relação à MTU

O protocolo TCP, no estabelecimento da conexão, negocia o MSS a ser usado, o qual é dependente da MTU. Se a rede pertence à mesma LIS, o MSS é dado pela Expressão 1 e cada *host* inclui no campo *option*(opção) do cabeçalho TCP o seu valor e anuncia ao parceiro o MSS proposto. O MSS escolhido é o menor valor entre os anunciados. Quando os *hosts* pertencem a LIS diferentes, o valor do MSS anunciado pelo TCP corresponde ao parâmetro configurado no *kernel*(núcleo do sistema operacional). Nos testes os *hosts* pertencem à mesma LIS, portanto o valor do MSS é negociado.

$$MSS_{TCP} = (MTU - Cabeçalho_{(TCP)} - Cabeçalho_{(IP)}$$

Expressão 1: Normalmente o tamanho dos cabeçalhos somam 40 octetos, sendo 20 octetos do TCP e 20 octetos do IP, porém quando se usa o fator de escala, o cabeçalho do TCP tem um acréscimo de 12 octetos[12].

Neste experimento foram utilizadas: MTU de 1500 octetos, por ser o valor básico das redes tradicionais IP; MTU de 9180 octetos que corresponde ao valor sugerido na RFC-2225 para redes ATM e MTUs de 4060 e 6620 por corresponder a intervalos regulares entre os valores extremos. As mensagens foram fixadas em 64 k octetos e os *buffers de sockets* variaram igualmente de 8 k octetos a 64 k octetos, em passos de 8 k octetos.

Pode ser observado pelos resultados contidos na Fig. 6, que não houve predomínio de algum valor específico de MTU ou de *buffer* de *sockets*, porém o melhor desempenho médio está para as MTUs de maior valor, independente do tamanho do *buffer* de *sockets*. Como não foi analisado ou medido o tempo gasto de CPU para efetuar as tarefas, pode-se inferir que é mais efetivo para o processamento utilizar *buffer* de *sockets* e MTUs com valores mais elevados. Uma maior MTU diminui a fragmentação de pacotes na camada IP e pode gerar, quando submetido a alto tráfego, melhor desempenho[13].

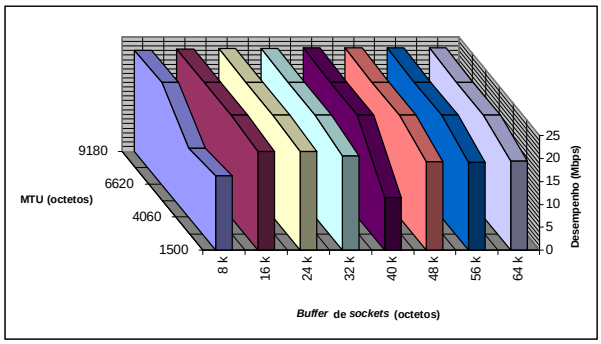

Figura 6: Desempenho da rede em relação a MTU e ao tamanho de buffer de sockets

5.3 Influência no Tamanho de *Buffers* de *Sockets*

Nesta avaliação a MTU foi configurada com o valor *default* (9180 octetos) e os *buffers* de *sockets* variaram independentemente entre 4k octetos a 64k octetos, com incrementos de 4k octetos. A Fig. 7 mostra os resultados obtidos, sendo os melhores encontrados para *buffers* de *sockets* maiores ou iguais a 28k octetos; apresentando em média desempenho(*throughput*) de 22,53 Mbps e máximo de 22,57 Mbps, correspondendo a 97,98% e 98,15%, respectivamente, do inicialmente previsto. Todavia, nos demais valores de *buffers*, era de se esperar um acréscimo no desempenho à medida em que os *buffers* aumentassem, porém, como pode ser observado na área hachurada da Tabela 6, quando foi mantido o valor do *buffer* de transmissão e acrescentado *buffers* na recepção houve um significante decréscimo.

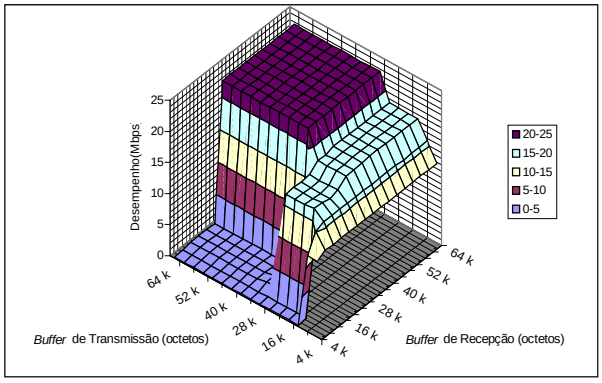

Figura 7: Desempenho da rede em relação ao tamanho de *buffers* de *sockets* na transmissão e recepção

Especialmente pode ser observado que quando se utilizou *buffer* de transmissão e recepção com tamanhos iguais a 4k octetos obteve-se 12,14 Mbps e na transição no *buffer* de recepção para 12k octetos, obteve-se apenas 0,16 Mbps de desempenho, que corresponde a 1,32% do resultado anterior. Em [14] obteve fenômeno semelhante quando comparou o desempenho, em ambiente local, entre uma rede Ethernet a 10 Mbps e uma rede ATM a 100 Mbps. Enquanto a primeira obteve desempenho médio de 1,313 Mbps, a segunda obteve desempenho médio de 0,366 Mbps.

T/R	4k	12k	20k	28k	36k	44k	52k	60k
4k	**12,14**	0,16	0,16	0,16	0,16	0,16	0,16	0,16
12k	12,79	**16,94**	16,69	0,91	0,56	0,56	0,58	0,56
20k	12,79	16,56	**16,54**	0,93	0,56	0,56	0,56	0,56
28k	12,84	18,07	18,13	**22,52**	22,55	22,55	22,57	22,54
36k	12,8	18,32	18,38	22,44	**22,54**	22,54	22,57	22,57
44k	12,81	18,37	18,39	22,57	22,45	**22,57**	22,57	22,47
52k	12,8	18,39	18,37	22,57	22,57	22,46	**22,54**	22,54
60k	12,84	18,31	18,41	22,57	22,47	22,54	22,57	**22,47**

Tabela 6: Desempenho(Mbps) da rede em relação ao tamanho(octetos) de *buffers de sockets* na transmissão(T) e recepção(R)

Este alto decréscimo de desempenho foi devido ao atraso gerado no receptor em encaminhar a confirmação do recebimento dos dados(ACK) ao transmissor por, implementar no protocolo TCP o algoritmo de Nagle, descrito na RFC-1122[15], que será demonstrado abaixo.

5.4 Algoritmo de Nagle

O algoritmo de Nagle objetiva otimizar o desempenho na rede e evitar situações de bloqueio(*deadlock*) quando as aplicações enviam ao protocolo TCP pequenas unidades de dados para transmissão. O algoritmo de Nagle é utilizado em conexões TCP e aplicado tanto no transmissor como no receptor[10]. Inicialmente vamos fazer uma breve descrição do algoritmo de Nagle

No transmissor:
Se existirem dados em *buffer* esperando pela transmissão e também dados transmitidos e não confirmados pelo host destino, o TCP só transmitirá os novos dados se atendidas uma das seguintes condições:

Condições
S1: Se há dados pendentes de recebimento de confirmação(ACK) e se o Mínimo(D,U) ≥ 1 x MSS, então transmite um segmento com 1 x MSS de dados. Onde, D é a quantidade de dados a ser transmitido e U é o tamanho da janela (window size) do receptor.

S2: Se há dados pendentes de recebimento de confirmação(ACK) e logo após for recebido ACK dos dados pendentes e existam X octetos de dados esperando no *buffer* do transmissor, então transmite-se um segmento com o Mínimo(X,U) de dados.

Notas:
1) No critério S1, se o *host* origem tem no mínimo um MSS de dados a transmitir e o *host* destino tenha espaço em *buffer* para receber um MSS, então o TCP transmite um MSS.

2) No critério S2, para que o *host* origem transmita novos dados recebidos da aplicação, será necessário de antemão que ele receba ACK dos dados pendentes.

3) O TCP sempre verifica a condição S1 antes da S2. Sendo falsa a condição S1, X sempre será menor do que um MSS. Quando existem dados pendentes, diante das condições S1 e S2, permite-se ao protocolo TCP armazenar em *buffers* os pequenos segmentos de dados recebidos da aplicação, até que ele possa enviar um MSS. Pode ser observado pela condição S2, que quando não existam dados pendentes o protocolo TCP transmite os dados enfileirados no *buffer de sockets* envio, mesmo sendo menor que um MSS.

4) Pode ser observado pelos critérios S1 e S2 que, quando não há espaço disponível em *buffer* no *host* origem e há espaço em *buffer* no *host* destino e ainda a aplicação continua a gerar novos dados, o protocolo TCP entrará em estado de bloqueio.

No receptor:

O *host* destino da conexão TCP pode otimizar o desempenho do TCP, reduzindo o processamento do protocolo em ambos os lados, e gerar menos tráfego, atrasando a confirmação (*delayed* ACK) dos segmentos recebidos[16]. Um receptor TCP também pode implementar *delay* ACK gerando menos ACK por segmento recebido e é responsabilidade de cada implementação do protocolo TCP utilizar o *delay* ACK[RFC-1122]. A RFC recomenda que quando o TCP recebe segmentos completos(MSS) correspondentes ao negociado na fase de estabelecimento da conexão, ele deve, no mínimo, encaminhar um ACK para o segundo segmento recebido. Os sistemas operacionais utilizados nos testes utilizam a implementação do *delay* ACK, da seguinte forma:

Condições:

R1: Se o *buffer* do receptor está vazio e a janela avançar em no mínimo em duas vezes o MSS, então transmite um ACK.

R2: Se a janela do receptor avançar em no mínimo 50% do total de espaço do *buffer*, então transmite um ACK.

Notas:

1) A janela do receptor avança na medida que a aplicação retira os dados do *buffer* de recepção. O protocolo TCP verifica a condição R1 antes da condição R2. A condição R1 garante ACK do protocolo TCP ao *host* remoto após cada dois MSS completos recebidos. Na recepção TCP com um pequeno *buffer*, quando comparado ao MSS do TCP, a condição R2 gera ACKs para permitir ao *host* origem transmitir mais dados.

2) Na RFC-1122 estipula que o TCP deve enviar um ACK ao *host* destino no máximo em 500 ms(milisegundos) após a recepção dos dados. O sistema operacional utilizado nos testes enviam ACK em 200 ms. Observa-se que as duas condições R1 e R2 não garantem que um *host* no destino sempre atenderá tais requisitos. É o caso em que a CPU não tenha capacidade suficiente de processamento, ou encontra-se sobrecarregada e retira os dados do *buffer* muito lentamente.

Agora voltando à análise do baixo desempenho quando houve alteração do tamanho do *buffer* de *sockets*, no receptor para 12k e mantido o tamanho de 4k no *buffer* de *sockets* do transmissor. O tamanho do bloco da mensagem era de 64k octetos; a MTU de 9180 octetos e a MSS de 9140 octetos. A Fig. 8 mostra a ocorrência de bloqueio no transmissor.

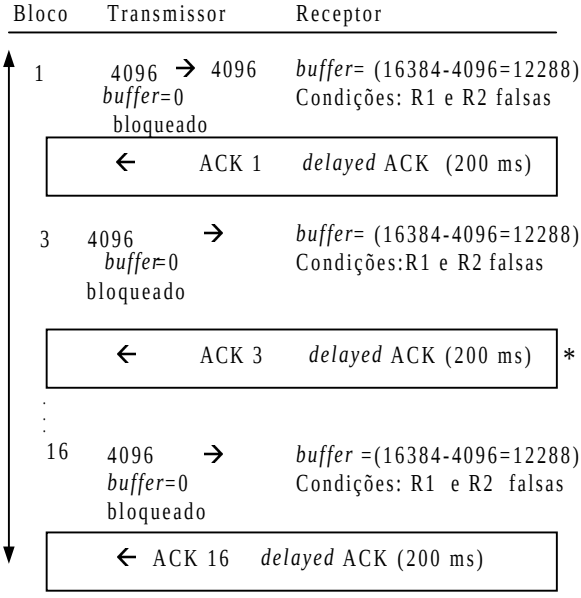

Figura 8: Desempenho da rede em relação ao tamanho de *buffers* de *sockets* na transmissão e recepção.

(*) Situação anormal - Condição de bloqueio

Nesse caso, o transmissor só consegue transmitir 1 segmento de 4096 octetos a cada vez, pois não há mais espaço em seu *buffer*. Apesar do espaço do *buffer* de *sockets* no receptor, ele ficará bloqueado por 200 ms devido ao atraso gerado no receptor no envio do ACK. Conseqüentemente o desempenho não deve ser superior a 163,292 kbps [(4096 octetos X 8 bits) / (200 ms + RTT)], o que se aproxima do valor encontrado 0,16 Mbps. O RTT(*Roudin Trip Time*) calculado para a rede em questão corresponde a 670 μs (micro segundos)

De acordo com [14] e constatadas através dos experimentos na rede metropolitana, cujos resultados foram mostrados na Tabela 6, para evitar as situações de bloqueio do protocolo TCP, as seguintes medidas devem ser tomadas:

- Desabilitar o algoritmo de Nagle;
- Configurar o tamanho do *buffer de sockets* no transmissor maior que 3 X MSS;
- Configurar o tamanho do *buffer de sockets* no transmissor de forma que ele seja maior ou igual ao *buffer de sockets* do receptor.

As duas últimas medidas são difíceis de colocá-las em prática, visto o ambiente heterogêneo de configurações de máquinas participantes de uma rede tradicional. Aliado ainda ao fato de que incrementar *buffers* pode ultrapassar a quantidade de memória, de acesso rápido, disponível na estação e levá-la a utilizar a memória em disco, cujo acesso de leitura e escrita é mais lento, gerando maior atraso na remessa de dados e por conseqüência degradando o desempenho.

5.5 Influência no Desempenho em Relação ao Fator de Escala

No cabeçalho do TCP, o campo *window size*(tamanho da janela) é de 16 bits, significando que 2^{16} *bits* podem ser transmitidos sem recebimento de ACK, porém esse tamanho apresenta limitação em redes com alta velocidade ou ainda redes que apresentam um alto *delay* (satélites geoestacionários). Para suprir essa deficiência na RFC-1323[12] foi proposto e já adotado em vários sistemas operacionais para estender este campo, permitindo enviar maior quantidade de segmentos sem recebimento de ACK. Essa extensão suporta atualmente 2^{30} bits.

Neste experimento a MTU foi configurada com o valor *default* (9180 octetos), mensagens com tamanho de 64k octetos, *buffer* de *sockets* iguais no transmissor e receptor, porém variaram de 64 a 128k octetos em passos de 1k octeto e habilitada a RFC-1323. Como pode ser observado na Fig. 9, o desempenho manteve-se praticamente regular enquanto se variava o tamanho de *buffer* de *sockets*.

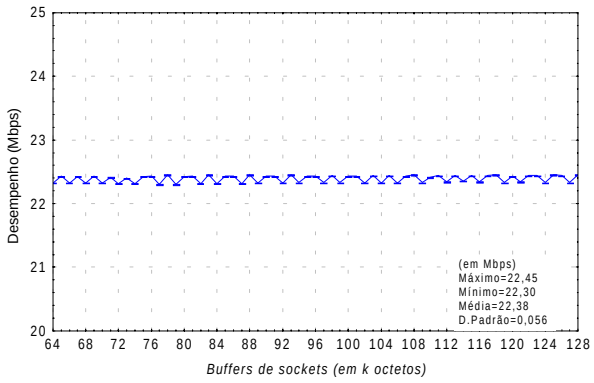

Figura 9: Desempenho da rede em relação ao tamanho do *buffer de sockets* utilizando o fator de escala do TCP.

6. CONCLUSÕES

Os experimentos permitiram identificar os fatores que contribuem, degradam e bloqueiam o desempenho do protocolo TCP/IP quando se usa ATM como rede básica. O desempenho obtido para o modelo do IP Clássico apresentou desempenho(*throughput*) médio de 22,53 Mbps e máximo de 22,57 Mbps, que corresponde a 97,98% e 98,15% do inicialmente previsto, o que corresponde a 88% da capacidade da linha de transmissão. Essas taxas de transmissão foram obtidas otimizando o tamanho de mensagens aliado a MTU. Muito embora esses valores sejam os melhores obtidos, a presença de *buffers de sockets* desbalanceados levaram o desempenho a cair para apenas a algumas centenas de kbps. Entretanto foi mostrado neste artigo como se pode evitar tais condições. São elas: Desabilitar o algoritmo de Nagle; configurar o tamanho do *buffer de sockets* no transmissor maior que 3 X MSS; configurar o tamanho do *buffer de sockets* no transmissor de forma que ele seja maior ou igual ao *buffer de sockets* do receptor.

6. REFERÊNCIAS

[1] PARTRIDGE, Craig, "Gigabit networking" Addison-Wesley, 1994.

[2] LAUBACH, M., J. Halpern, "Classical IP and ARP over ATM". RFC 2225, Newbridge Networks, april, 1998.

[3] IBM 8285 Nways ATM Workgroup Switch,Installation and Users's Guide,1996.

[4] ITU-T Recommendation I.432.5 : "B-ISDN user-network interface – Physical layer specification: 25,600 kbit/s operation", june, 1997.

[5] ATM Forum, "Physical Interface Specification for 25.6 Mb/s over Twisted Pair Cable, af-phy-0040.000". november, 1995.

[6] ATM Forum, "User-Network Interface Version 3.1 Specification", september, 1994.

[7] ITU-T Recommendation I.363.5 – "B-ISDN ATM Adaptation Layer specification: Type 5 AAL". Agosto de 1996.

[8] STALLINGS, William, "High-speed networks –TCP/IP and ATM design principles", Prentice-Hall, Inc., 1998.

[9] Hewlett-Packard Company, "Netperf: A network performance benchmark", revision 2.1, Information Network Division Hewlett-Packard Company, 1996.

[10] COMER, E. Douglas, Internetworking with TCP/IP. vol. 1,Prentice-Hall, Inc., 1995.

[11] ANDRIKOPOULOS, T. Örs I et all, "TCP/IP throughput performance evaluation for ATM local area networks" , Proceedings of 4th IFIP Workshop on Performance Modeling and Evaluation of ATM Networks, Ilkley, UK, june, 1996.

[12] JACOBSON, V, BRADEN, R, BORMAN,D., TCP extensions for high performance, RFC-1323, mai. 1992.

[13] ANTUNES, Paulo C. A, et al, IP over ATM – performance evaluation, experiments and results, Proceeding of 5th IFIP International Conference on Hong-Kong, november 1999.

[14] MOLDEKLEV, K. GUNNINGBERG, P., Deadlock situations in TCP over ATM, Proceeding of 4th IFIP workshop on protocols for high speed networks, Vancouver, B.C. Canada, ago. 1994.

[15] BRADEN, Robert, Requirements for internet hosts-communication layers, RFC-1122, out. 1989.

[16] CLARK, David D., Window and acknowledgement strategy in TCP, RFC-813, jul. 1982.