

Análise de Desempenho de TCP sobre GPRS em um Ambiente Fim a Fim

Jeísa P. Oliveira, Carlos A. Kamienski, Judith Kelner, Djamel F. H. Sadok
Centro de Informática – Universidade Federal de Pernambuco
{jpo,cak,jk,jamel}@cin.ufpe.br

Abstract — The General Packet Radio Service (GPRS) represents a packet-based extension for GSM and was developed to address the Internet wireless access demand. Wireless networks, including GPRS, suffer from high packet loss levels which lead the TCP transport to low performance due to its use of implicit congestion notification mechanism. Although there have recently been a number of publications on this topic, they present a limited scope in their analysis and since different scenarios are used the results cannot be analysed as a whole. This work presents wider performance issues related to TCP over the GPRS air interface using simulation with different TCP versions, like New-Reno, SACK and Westwood, and dynamic adaptation of GPRS coding schemes.

Key-words — GPRS, TCP, Performance Analysis

Resumo — GPRS é uma extensão da tecnologia GSM para a transmissão de dados baseada em comutação de pacotes e surgiu pela demanda por acesso sem fio à Internet. Redes sem fio, incluindo GPRS, apresentam grandes taxas de perdas de pacotes, que causam problemas de desempenho para o protocolo TCP, devido ao seu mecanismo de notificação implícita de congestionamento. Alguns trabalhos estudam o comportamento do TCP sobre o GPRS, contudo cada um com um escopo bastante limitado e com cenários distintos entre si não sendo possível analisar em conjunto os resultados. Este artigo apresenta uma análise mais abrangente do desempenho do TCP sobre GPRS, baseada em simulação, que inclui diferentes versões do TCP, como New-Reno, SACK e Westwood, e um modelo adaptativo para configurar dinamicamente cenários de codificação.

Palavras-chave — GPRS, TCP, Análise de Desempenho

I. INTRODUÇÃO

Nos últimos anos, tem havido um contínuo crescimento tanto na área de comunicação celular quanto na Internet, o que gerou a necessidade de acesso sem fio à Internet. O GPRS (*General Packet Radio Service*) [1] surgiu para atender a essa demanda, estendendo uma das tecnologias mais utilizadas na área: GSM (*Global System for Mobile communications*). Enquanto GSM utiliza comutação por circuitos, o GPRS utiliza comutação por pacotes para transmitir dados, permitindo comunicação entre as estações móveis e redes externas de pacotes de dados, incluindo a Internet.

Ao mesmo tempo em que permitem uma grande

flexibilidade de comunicação e acesso à Internet ao usuário (pela mobilidade proporcionada em grandes áreas geográficas), redes sem fio como a GPRS apresentam grandes taxas de erros na sua interface aérea, que variam significativamente de acordo com as condições do ambiente. Este aspecto causa problemas de desempenho no protocolo TCP (*Transmission Control Protocol*), que é utilizado pela maioria dos serviços de aplicação oferecidos atualmente na Internet. Em situações de congestionamento aparente, o TCP reduz a sua janela de congestionamento, causando a diminuição da taxa de transmissão. Como não possui nenhum mecanismo de notificação explícita da ocorrência de congestionamento em algum ponto da rede, o TCP transmissor o infere (implicitamente) através da perda de pacotes. Esse mecanismo foi desenvolvido para redes cabeadas, onde raramente ocorrem perdas de pacotes devido a problemas no meio físico. Em uma rede sem fio, como a GPRS, na ocorrência de perda de pacotes na interface aérea, o TCP não possui informação suficiente para distinguir entre perdas de pacote na rede sem fio e perdas devido a descartes nas filas dos roteadores. Por este motivo, é importante que o seu comportamento sobre o GPRS seja cuidadosamente analisado.

Alguns estudos sobre esse assunto têm sido realizados nos últimos anos. No entanto, cada um deles em geral avalia apenas poucos aspectos possíveis em um cenário que suporta TCP com GPRS. Em [2] é feita uma avaliação, através de simulações, da harmonia entre o mecanismo ARQ (*Automatic Repeat Request*) do TCP e do GPRS (RLC – *Radio Link Control*). Aspectos relacionados à vazão geral do sistema são avaliados em [3], através da modelagem de um tráfego Web. E em [4] medições reais são realizadas e, considerando o cenário fixo utilizado, são apresentados os problemas detectados na interação entre o TCP e o GPRS, seguidos de uma proposta para superar os mesmos. Uma vez que cada trabalho estuda um número reduzido de aspectos e utiliza cenários, ambientes e ferramentas diferentes, não se pode unir todos os resultados e estudá-los como um único conjunto para entender o comportamento do TCP interagindo com o GPRS. Há, portanto, uma necessidade de estudar vários aspectos em um mesmo cenário, variando os parâmetros e as condições desse ambiente a fim de se obter um estudo abrangente, que permita compreender melhor a interação TCP/GPRS.

Este trabalho visa analisar o desempenho do TCP sobre o

GPRS através de simulações, variando parâmetros referentes aos dois protocolos. Vários cenários distintos foram avaliados, incluindo diferentes taxas de erros (C/I) no meio físico de transmissão, quantidades de usuários e de *timeslots* utilizados por usuário no sentido *downlink*. Além disso, alguns aspectos inovadores foram avaliados, como o impacto do tamanho do segmento TCP sobre o seu desempenho e o uso de um modelo adaptativo para configurar cenários de codificação (CSs – *Coding Schemes*) da camada RLC do GPRS. Tal modelo muda dinamicamente o CS utilizado baseado na taxa de blocos RLC perdidos por janela de transmissão, permitindo uma melhor utilização dos recursos e, conseqüentemente, uma maior taxa de transmissão de dados.

O restante do artigo é organizado da seguinte forma: a Seção II apresenta as características principais do GPRS; a Seção III apresenta o TCP, descrevendo brevemente seu funcionamento; as configurações utilizadas nas simulações são expostas na Seção IV; a Seção V mostra os resultados e, finalmente, a conclusão e os trabalhos futuros são apresentados na Seção VI.

II. GPRS

A. Rede GPRS

Além dos elementos que compõem a rede GSM, a arquitetura da rede GPRS apresenta novos nós: os GPRS *Support Nodes* (GSNs). Esses nós são responsáveis pela entrega e roteamento de pacotes de dados entre as estações móveis e as redes de pacotes de dados (PDN – *Packet Data Network*) externas. Dois tipos de GSNs foram definidos, *Serving GPRS Support Nodes* (SGSN) e *Gateway GPRS Support Nodes* (GGSN). O primeiro responsabiliza-se pela entrega de pacotes de e para as estações móveis dentro de sua área de serviço. O segundo, GGSN, age como uma interface entre a rede *backbone* GPRS e as redes de pacotes de dados externas.

B. Protocolos do GPRS

A especificação do GPRS define protocolos para o plano de transmissão e para o plano de sinalização. Cinco protocolos controlam a comunicação entre a estação móvel (MS – *Mobile Station*) e o BSS (*Base Station System*): LLC (*Logical Link Control*), RLC (*Radio Link Control*), MAC (*Medium Access Control*), PLL (*Physical Link Layer*) e RFL (*Physical RF Layer*). O LLC provê um link lógico bastante confiável entre a MS e o SGSN associado a ela. Suas funcionalidades incluem controle de seqüência, entrega em ordem, detecção e correção de erros, e retransmissão. Possui os modos de transmissão de quadros com e sem reconhecimento (*acknowledged* e *unacknowledged*, respectivamente). O RLC tem como principal objetivo estabelecer um link confiável entre a MS e o BSS. Entre suas funções encontram-se fragmentação e remontagem dos quadros LLC em blocos de dados RLC; e correção de erros através de um mecanismo de retransmissão seletiva desses blocos.

O MAC controla o acesso aos canais de rádio, ou seja,

determina como os pares *frequência-timeslot* são compartilhados pelas MSs. Baseado no *slotted* ALOHA, o MAC realiza resolução de contenção entre as tentativas de acesso ao canal, multiplexação de vários usuários em um mesmo canal lógico (PDTCH), e escalonamento e priorização baseado no QoS (*Quality of Service*) negociado. A PLL e a RFL constituem a camada física. A PLL provê um canal físico entre a MS e o BSS. Entre suas funções estão codificação dos dados e detecção e correção de erros de transmissão no meio físico. A RFL realiza modulação/demodulação das ondas físicas.

C. Interface Aérea

A interface aérea do GPRS utiliza a estrutura do GSM. Na camada física, o GSM usa uma combinação de TDMA (*Time Division Multiple Access*) com FDMA (*Frequency Division Multiple Access*). O espectro reservado para o GSM, que varia de acordo com o país, é dividido em duas bandas de frequência de 25 MHz, separados por 45 MHz. Na Europa (ver Figura 1), são usadas as bandas 890 – 915 MHz para transmissão *downlink* (i.e., no sentido BSS-MS) e 935 – 960 MHz para transmissão *uplink* (i.e., no sentido MS-BSS). Cada banda é dividida em 124 portadoras de canais (de frequências) com 200 kHz de largura. Um determinado número dessas portadoras é reservado para uma célula.

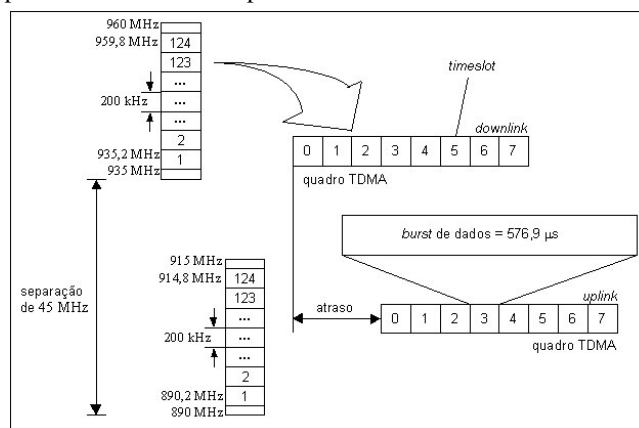


Fig. 1. Portadoras de frequência GSM, duplexação e quadros TDMA

Cada portadora é dividida em quadros (*frames*) TDMA, os quais são compostos por oito *timeslots*, ou seja, oito canais. Cada *timeslot* (TS) dura 576,9 ms, logo um quadro dura 4,615 ms. A repetição de um *timeslot* particular define um canal físico. Como pode ser observado na Figura 1, o início de um quadro TDMA no sentido *uplink* é iniciado com um atraso de três *timeslots* do início de um quadro TDMA no sentido *downlink* para evitar interferência no terminal.

Uma estação móvel GSM usa o mesmo *timeslot* nos dois sentidos: *downlink* e *uplink* [5]. GPRS apresenta um mecanismo de alocação de canais mais flexível: mais de um *timeslot* (operação multislot) pode ser reservado para uma mesma MS. Além disso, a alocação *downlink* e *uplink* pode ser assimétrica, pois é realizada separadamente. Em um sistema GSM, um canal é alocado permanentemente durante todo o período de uma chamada, havendo ou não transmissão de

dados. Em GPRS, o canal é alocado apenas quando há transmissão de pacotes, beneficiando o usuário, que não precisa se preocupar com o tempo de conexão.

D. Cenários de Codificação

A codificação de canais é usada para proteger pacotes de dados transmitidos contra erros. Há quatro cenários de codificação no GPRS, os quais são utilizados de acordo com as condições do canal (ruído). A Tabela I apresenta os valores envolvidos nos cenários de codificação. Nela pode ser observada a quantidade de dados com o cabeçalho RLC, que é o número de octetos contendo dados úteis que pode ser transmitido em cada bloco, dependendo do CS utilizado. Também são mostradas a taxa de codificação e a taxa de dados alcançada por cada cenário de codificação.

TABELA I
CENÁRIOS DE CODIFICAÇÃO

Cenário de codificação	Dados e Cabeçalhos	Taxa de codificação	Taxa de dados
CS-1	22 octetos	1/2	9,05 Kbps
CS-2	32 octetos	≈ 2/3	13,4 Kbps
CS-3	38 octetos	≈ 3/4	15,6 Kbps
CS-4	52 octetos	1	21,4 Kbps

E. Modelo Adaptativo para os Cenários de Codificação

O cenário de codificação deve ser escolhido de acordo com as condições do canal (ruído). Quanto piores as condições do ambiente de transmissão, maior deve ser a proteção dos dados. E quanto maior a proteção, menor a quantidade de dados úteis (i.e., dados do usuário). Assim, se não for utilizado o cenário adequado, haverá perda de dados ou subutilização do canal. Se, por exemplo, o CS-4 for utilizado em um ambiente com muito ruído, vários pacotes serão perdidos, pois este cenário não apresenta proteção aos dados. Por outro lado, em um ambiente bom, com pouco ruído, pacotes praticamente não são perdidos, não havendo necessidade de redundância (proteção) de dados. Se em um ambiente como esse, onde o uso do CS-4 seria apropriado, o CS-1 for escolhido, a taxa de dados transmitidos será bastante inferior a que poderia ser alcançada, uma vez que menos dados úteis serão enviados. Logo, se houvesse uma forma de mudar o CS utilizado de acordo com as condições do ambiente, a quantidade de dados transmitidos seria maior.

Em [6], um modelo de adaptação para os cenários de codificação é apresentado. Tal modelo modifica o cenário de codificação baseado no percentual de blocos descartados durante a transmissão da janela. Isto é, para cada bloco enviado, um BLER é coletado, indicando o descarte ou não do bloco. Ao término de transmissão de uma janela (ou seja, após a transmissão de 64 blocos), calcula-se o percentual de blocos descartados (%BD), dividindo-se o total de blocos descartados (BD) pelo total de blocos transmitidos (BT): $\%BD = BD / BT * 100$. De acordo com o valor desse percentual, o cenário de codificação pode ser mudado. Também é apresentado um estudo que determina os valores para a transição entre os cenários. Esses valores podem ser observados na Figura 2.

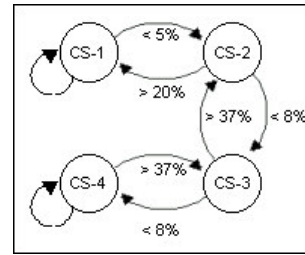


Fig. 2. Valores dos parâmetros de transição entre os CSs

O modelo de erros empregado consiste na utilização de uma tabela com valores de *block error rate* (BLER) pré-calculados, fornecida pela Ericsson [6]. O modelo funciona da seguinte forma. Cada nó possui um CS e um *C/I* (*carrier-to-interface*), os quais são determinados no início da simulação. Ao receber um pacote, o nó sorteia um valor aleatório entre 0 e 1, utilizando uma distribuição Normal. Esse valor é comparado então ao valor da tabela BLER que é obtido utilizando o CS e o *C/I* do nó. Se o valor sorteado for menor ou igual ao BLER coletado, o bloco é marcado como errado e, posteriormente, descartado.

III. TCP

O TCP [7] é um protocolo da camada de transporte, orientado à conexão, desenvolvido para oferecer funções de comunicação confiável e controle de congestionamento fim a fim. Para prover confiabilidade, o TCP utiliza o mecanismo ARQ (*Automatic Repeat Request*), que retransmite todos os segmentos que não são explicitamente confirmados. Para evitar o congestionamento da rede e a sobrecarga dos sistemas finais, o TCP realiza controle de tráfego através de um mecanismo de janela deslizante [7]. Esse mecanismo permite que o transmissor envie um determinado número de segmentos sem receber ACKs (*ACKnowledgements* – confirmações), e, ao receber um ACK, a janela “desliza” permitindo que mais segmentos sejam enviados. O tamanho da janela, indicado pelo variável CWND (*Congestion Window*), varia dinamicamente, baseado nos ACKs recebidos.

A. Controle de Congestionamento

Quatro algoritmos básicos governam o controle de congestionamento do TCP: *Slow Start*, *Congestion Avoidance*, *Fast Retransmit* e *Fast Recovery* [8]. O *Slow Start* consiste no aumento gradual da quantidade de dados inseridos na rede, através do aumento do tamanho da janela. É usado no início da conexão TCP e após a detecção de congestionamento. O algoritmo começa inicializando CWND com um valor pequeno (normalmente igual a 1) e, a cada ACK recebido, incrementa esse valor em um segmento, causando um crescimento exponencial da janela. O *Slow Start* continua até que seja atingido um limiar, denominado SSTHRESH (*Slow Start Threshold*), a partir do qual é utilizado o *Congestion Avoidance*. No *Congestion Avoidance*, o valor do CWND é maior ou igual a SSTHRESH. Este algoritmo incrementa CWND a uma taxa menor que o *Slow Start*. Para cada ACK

recebido, a janela é incrementada em $1/CWND$, produzindo um crescimento linear da janela.

Os algoritmos *Fast Retransmit* e *Fast Recovery* permitem que o TCP detecte e corrija a perda de segmentos antes que o RTO (*Retransmission Timeout*) expire. RTO consiste em um temporizador (*timer*) que é inicializado sempre que um segmento é enviado. Se um ACK não for recebido após esse tempo, considera-se que o segmento foi perdido. Assim, o segmento é retransmitido e o TCP volta ao *Slow Start*. O *Fast Retransmit* assume que um segmento foi perdido após receber três ACKs duplicados. Nesse caso, o TCP retransmite o segmento perdido e retorna ao *Slow Start* (se a versão do TCP não apresentar *Fast Recovery*, como o TCP Tahoe). O *Fast Recovery* pode ser utilizado após a retransmissão do segmento no *Fast Retransmit*, evitando que o TCP volte ao *Slow Start*. O *Fast Recovery* reduz a janela à metade, incrementando-a, em seguida, de forma linear.

Embora seu funcionamento básico esteja padronizado, o TCP apresenta diversas variações e aprimoramentos visando cada vez um melhor aproveitamento dos recursos da rede. Os algoritmos que implementa e a forma como estes são implementados diferenciam as versões do TCP. Nesse trabalho serão utilizadas quatro versões: Reno, New-Reno, SACK e Westwood.

O TCP Reno [9] implementa os quatro algoritmos sem modificações, i.e., da forma descrita em [8]. O New-Reno [10] consiste em uma versão melhorada do Reno, modificando o *Fast Recovery* para que ACKs parciais recebidos durante esse período sejam considerados um indicativo de que o segmento imediatamente após o confirmado pelo ACK (na seqüência) foi perdido e deve ser retransmitido.

O TCP SACK (*Selective ACKnowledgement*) [11] também altera o *Fast Recovery* e permite confirmar, de forma não-sequencial, mais de um segmento de uma vez. Utilizando SACK, o transmissor pode saber exatamente quais segmentos foram perdidos, evitando atrasos e retransmissões desnecessárias. O TCP Westwood também será utilizado e, por ser uma versão recente, será explicado de forma mais detalhada a seguir.

O protocolo TCP não apresenta um bom desempenho quando é utilizado em redes sem fio. Por ter sido desenvolvido para redes fixas, o TCP deduz que perda de pacotes implica em congestionamento. Contudo, em redes sem fio, perda de pacotes pode ter outros significados: ruído na interface aérea, *handoffs*, desconexões não planejadas etc. Assim, as versões mais comuns do TCP, como Tahoe e Reno, não apresentam funcionamento adequado para redes sem fio. Há várias propostas para reduzir os efeitos de perdas de pacotes não relacionadas a congestionamento em redes sem fio, como em [9], [12] e [13]. Uma das propostas mais recentes é o TCP Westwood (TCPW).

O TCPW [14] consiste em uma versão melhorada do TCP New-Reno, que modifica o esquema de controle da janela de congestionamento (CWND) do lado do transmissor usando uma estimativa da largura de banda. A fonte TCPW estima

constantemente a taxa de pacotes da conexão através do monitoramento da taxa de recepção de ACKs. Essa estimativa é usada para calcular a CWND e o SSTHRESH que podem ser utilizados após a detecção de congestionamento (i.e., após o recebimento de três ACKs duplicados ou após um *timeout*). A principal idéia é evitar que a janela seja reduzida à metade ou a um segmento desnecessariamente. Ainda não há publicações que estudem o comportamento do Westwood sobre GPRS. Este trabalho realiza uma avaliação inicial do desempenho do TCPW sobre GPRS através da vazão e comparando-o com Reno, New-Reno e SACK.

IV. CONFIGURAÇÕES DE SIMULAÇÃO

O ambiente de simulação utilizado é o *Network Simulator* (ns-2) [15], um simulador de eventos discretos muito usado no ambiente acadêmico, cujo objetivo é auxiliar em pesquisas sobre redes de comunicação. Para a simulação do GPRS, foi utilizado o módulo descrito em [16] que focaliza a interação MS-BS, implementando, assim, as camadas LLC, RLC e MAC. A implementação utilizada nesse trabalho consiste de uma versão aprimorada do módulo original, que apresentava vários problemas e limitações que tiveram que ser superados para gerar resultados significativos.

A Figura 3 ilustra a topologia utilizada nas simulações, a qual é formada por uma rede GPRS, dois roteadores intermediários e um conjunto de sistemas finais (*hosts*). A rede GPRS é constituída por um roteador GPRS, que une as funções do GGSN e do SGSN, uma estação rádio-base (BS) e n estações móveis (MSs). Os enlaces fixos apresentam capacidade de 1Mbps e atraso de 10ms. O número de sistemas finais é igual ao número de MSs, havendo uma correspondência de um-para-um entre as MSs e os *hosts* de mesmo índice. Ou seja, o *host* i se comunica com a MS i . A transmissão de dados ocorre no sentido *host* \rightarrow MS (i.e., no sentido *downlink*), simulando um usuário realizando *download* de arquivos.

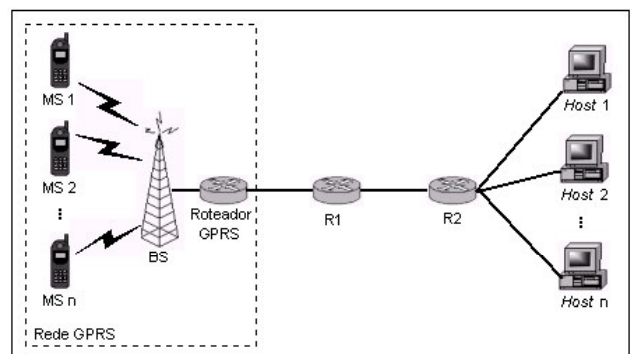


Fig. 3. Topologia de simulação

Os parâmetros utilizados na simulação são apresentados na Tabela II. O objetivo principal deste trabalho é avaliar o protocolo TCP, focando nas versões Reno, New-Reno, SACK e Westwood. Também foi avaliado o protocolo UDP (*User Datagram Protocol*), para fins de comparação, porque este permite a utilização de aplicações que geram taxas de

transmissão constantes (CBR) e não é baseado em confirmações. Foram utilizados três tamanhos diferentes de pacotes (segmentos) TCP (536, 1000 e 1500 bytes) e apenas um tamanho (200 bytes) de pacote UDP. Todos os experimentos foram realizados com 1, 5 e 50 fontes de dados (em usuários móveis). A taxa de erros da interface aérea (C/I médio) variou de 2 a 30 dB, com intervalo de 2 dB entre elas. Os experimentos incluíram os quatro cenários de codificação (CS-1, CS-2, CS-3 e CS-4), que foram configurados estaticamente, além do modelo adaptativo descrito na Seção II.E. Os modos com e sem reconhecimento da camada RLC foram avaliados. Foi utilizado um *timeslot* no sentido da MS para a BS (*uplink*) e um, dois e quatro *timeslots* no sentido contrário (configurações 1+1, 2+1 e 4+1). Para todas as simulações foi utilizado um número de frequências suficiente para todas as estações transmitirem e, em cada frequência, cinco canais de dados (PDCH – *Packet Data Channels*) foram permitidos.

TABELA II
PARÂMETROS DA SIMULAÇÃO

Parâmetro	Valor
Protocolo de transporte	TCP, UDP
Versão do TCP	Reno, New-Reno, Sack e Westwood
Tamanho do pacote TCP	536, 1000, 1500 bytes
Número de fontes/usuários	1, 5, 50
Taxa de erros (C/I médio)	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30 dB
Variância da taxa de erros	3 dB
Modelo adaptativo	sim, não
Cenário de codificação	1, 2, 3, 4
Modo RLC	com confirmação, sem confirmação
Multislot - <i>uplink</i>	1 TS
Multislot - <i>downlink</i>	1, 2, 4 TS
Número de PDCHs	5
Tempo de simulação	100 segundos

Em alguns experimentos preliminares, foi gerado tráfego de retaguarda (*background*) entre sistemas finais conectados entre os roteadores intermediários, para observar o impacto resultante no tráfego GPRS. Contudo, como não houve impacto significativo no tráfego GPRS, os resultados destes experimentos não são mostrados na Seção V.

As simulações tiveram duração de 100 segundos, que é um tempo suficiente para que os efeitos esperados pudessem ser observados. Para cada experimento, foram realizadas 100 replicações, variando as sementes com valores cuidadosamente escolhidos (sementes consideradas “boas” pelo ns-2). Esse número de replicações foi escolhido por representar um compromisso aceitável entre o custo computacional e a

confiança estatística nos resultados. Para todos os resultados, foi calculado um intervalo de confiança assintótico ao nível de 99%, que é representado através de barras horizontais nos resultados da Seção V.

Durante a execução dos experimentos, foram coletadas várias métricas, tanto relacionadas aos protocolos de transporte (TCP e UDP), quanto ao GPRS. As principais métricas TCP coletadas são vazão, tamanho da janela de congestionamento, número de segmentos transmitidos e número de segmentos retransmitidos. No que se refere ao GPRS, foram coletados dados da camada RLC, como número de blocos retransmitidos e perdidos, C/I e CS.

V. RESULTADOS

Esta seção apresenta os resultados mais significativos dos vários experimentos de simulação realizados. Devido à limitação de espaço apenas os gráficos referentes à vazão com relação a diversas condições ambientais (C/I) são mostrados. As demais métricas foram úteis para auxiliar a compreensão e a análise crítica dos dados.

A. TCP Reno sobre GPRS

Os primeiros experimentos estudam o comportamento do TCP Reno sobre o GPRS. A Figura 4 mostra a vazão atingida por uma MS submetida a diferentes condições de ambiente. Foi utilizada a configuração 1+1 (um *timeslot* no sentido *downlink* e um no sentido *uplink*) e pacote de 1500 bytes.

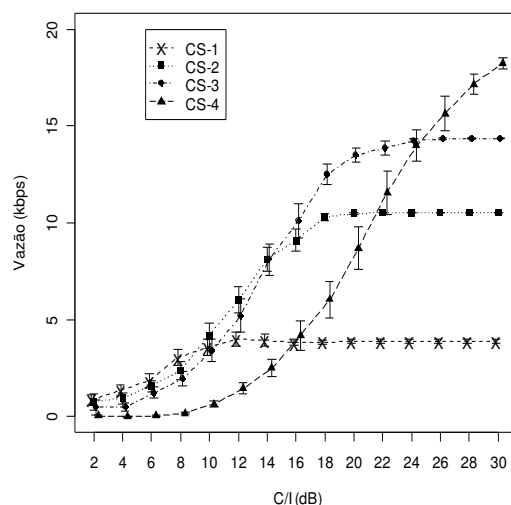


Fig. 4. Vazão do TCP Reno para os quatro cenários de codificação

Observando o gráfico, pode-se notar que há um CS adequado para cada C/I. Abaixo de 9 dB, o CS-1 obtém a maior vazão. Entre 9 e 14 dB, o CS-2 é capaz de transmitir maior quantidade de dados. O CS-3 apresenta melhor desempenho para C/Is de 15 a 24 dB, enquanto o CS-4 possui maior vazão a partir de 25 dB. Com esses resultados pode-se concluir que o uso dos CSs extremos (CS-1 e CS-4) só é justificado em condições ambientais extremas: o CS-1 para ambientes com muito ruído (C/I abaixo de 9 dB) e o CS-4 para ambientes com pouco ruído (acima de 25dB). Assim, os CSs mais apropriados são o CS-2 e o CS-3, que abrangem a faixa de 9 a 24 dB. Esta conclusão está de acordo com a experiência

das operadoras, que geralmente utilizam o CS-2 [4].

Na Figura 5 o tamanho do pacote (segmento) TCP é variado, assumindo valores de 536, 1000 e 1500 bytes. Os valores apresentados nos dois gráficos foram gerados utilizando 1 MS, com configuração 1+1.

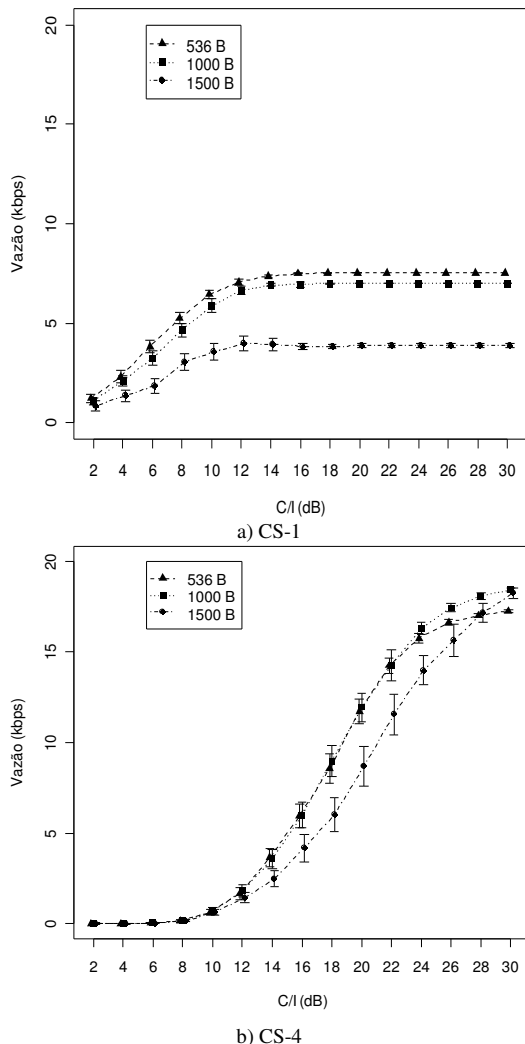


Fig. 5. Vazão do TCP Reno para diferentes tamanhos de pacote

Para o CS-1 (Figura 5a), o tamanho do pacote influencia a vazão: quanto menor o tamanho do pacote, maior a vazão. O esperado seria que a vazão fosse diretamente proporcional ao tamanho do pacote, uma vez que, quanto menor o pacote, menos dados podem ser transmitidos por segmento, sendo necessário o uso de mais pacotes e, conseqüentemente, mais cabeçalhos e maior sobrecarga (*overhead*). A diferença ocorre devido ao modo com confirmação (ACK) usado na camada RLC, que utiliza o mecanismo de janela deslizante com tamanho fixo de 64 blocos. A quantidade de dados úteis por bloco no CS-1 é reduzida (21 bytes, sem incluir cabeçalhos RLC e MAC). Assim, para transmitir um pacote de 1500 bytes (somando ainda cabeçalhos das camadas superiores), são necessários 74 blocos RLC. Mas a janela de transmissão da camada RLC só permite que sejam enviados 64 blocos sem que seja recebida uma confirmação (ACK). De acordo com a especificação [17], o receptor envia um ACK apenas quando

recebe o último pacote da seqüência, ou quando o transmissor requisita uma confirmação. Como o último bloco dessa seqüência não será transmitido dentro da janela, o receptor só enviará um ACK quando for solicitado. O transmissor solicitará um ACK quando a janela ficar completa, retransmitindo os blocos cujo recebimento ainda não foi confirmado. O tempo gasto desde a retransmissão do primeiro bloco não confirmado (com solicitação de confirmação) até a recepção do ACK e liberação da janela para novas transmissões causa a diminuição da taxa de transmissão observada no gráfico da Figura 5a. Com os demais CSs isso não é perceptível, porque, para os tamanhos de pacotes utilizados nos experimentos, o número de blocos gerados por pacote pode ser enviado em uma única janela de transmissão. O CS-4 pode ser visualizado na Figura 5b.

O número de estações móveis utilizadas nas simulações foi variado, assumindo os valores 1, 5 e 50. Os resultados obtidos (que não serão exibidos por questão de espaço) mostraram que o tráfego gerado por uma estação móvel não prejudicou as outras estações. Por este motivo, os demais resultados apresentados neste artigo foram gerados a partir de cenários com apenas uma MS.

Todos os estudos realizados sobre a interação TCP/GPRS utilizam apenas o modo com reconhecimento da camada RLC. A Figura 6 ilustra a vazão obtida por uma MS utilizando o modo sem reconhecimento da camada RLC, com configuração 1+1 e pacotes de 1500 bytes.

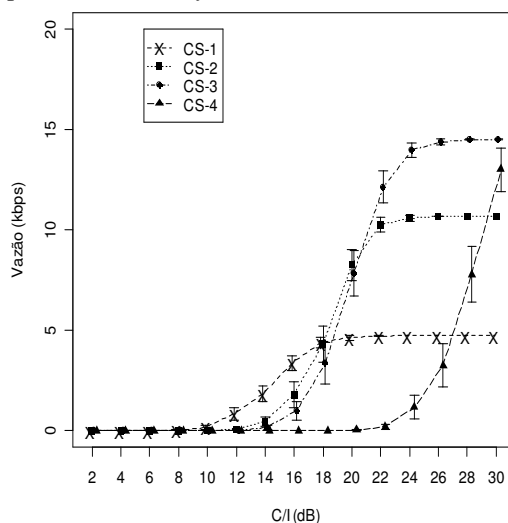


Fig. 6. Vazão do TCP Reno utilizando o modo RLC sem reconhecimento

Analisando o gráfico, pode-se perceber que em ambientes que apresentam muito ruído (C/I abaixo de 10 dB) a vazão é nula para todos os CSs. Esse resultado é bastante coerente uma vez que não ocorrem retransmissões de blocos perdidos e os erros têm que ser corrigidos utilizando a codificação oferecida pelo cenário. Como esperado, o CS-1 é o cenário que apresenta melhor desempenho por oferecer maior proteção aos dados. Enquanto o CS-4, que não proporciona proteção aos dados, consegue transmitir dados apenas em condições muito boas (depois de 22 dB), alcançando taxas bem abaixo do valor máximo teórico mesmo em ambientes com pouco ruído (30

dB). Observa-se ainda que a partir de um certo ponto (22 dB), a vazão alcançada pelos CS-1 a CS-3 é a mesma obtida pelo modo com reconhecimento (ver Figura 4). Dessa forma, os resultados mostram que seria melhor utilizar o modo sem reconhecimento para ambientes com muito boas condições, evitando o atraso e a sobrecarga (*overhead*) gerados pela troca de ACKs e pela limitação da janela de transmissão RLC. Vale ressaltar que o CS-4 não deve ser utilizado com esse modo de transmissão RLC visto que apresenta um desempenho ruim mesmo para ambientes com condições ótimas.

A Figura 7 ilustra as vazões atingidas variando o número de *timeslots* utilizados por usuário no sentido *downlink*. Além da configuração com apenas um *timeslot* (*singleslot*) utilizada para os demais resultados, foram utilizadas as configurações multislot 2+1 e 4+1.

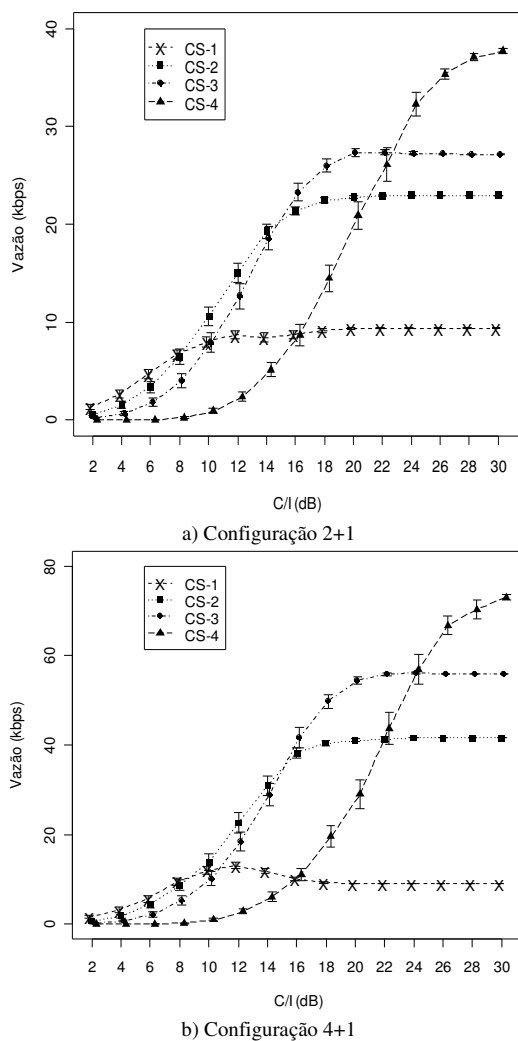


Fig. 7. Vazão do TCP com múltiplos timeslots

Comparando as Figuras 7a e 7b, observa-se que a vazão praticamente dobra, o que é absolutamente coerente visto que o número de *slots* utilizados é o dobro. Os resultados obtidos para a configuração 1+1 utilizando os mesmos parâmetros podem ser visualizados na Figura 4. Como esperado, a vazão obtida é aproximadamente igual à metade da vazão alcançada

com 2 TSs.

B. TCP X UDP

Para fins de comparação, os mesmos experimentos foram também realizados para tráfego UDP. Utilizando uma aplicação geradora de tráfego CBR (*Constant Bit Rate*) com uma taxa de transmissão suficientemente alta, é possível observar a vazão máxima atingida pelo GPRS para cada CS sob os diferentes valores de C/I. A Figura 8 apresenta os dados obtidos para o CS-1 e o CS-4 utilizando TCP Reno e UDP. Na Figura 8a o tamanho de pacote é 536 bytes. Pode-se perceber que, neste caso, o TCP apresenta um desempenho próximo do UDP. Contudo, na Figura 8b (onde o tamanho do pacote é 1500 bytes), quando a perda de pacotes torna-se desprezível para o CS-1 (acima de 12 dB), a vazão alcançada pelo TCP (≈ 4 kbps) é praticamente a metade da vazão do UDP (≈ 9 kbps). Isso se deve ao modo com confirmação da RLC e à quantidade de blocos RLC gerados nesse cenário de codificação.

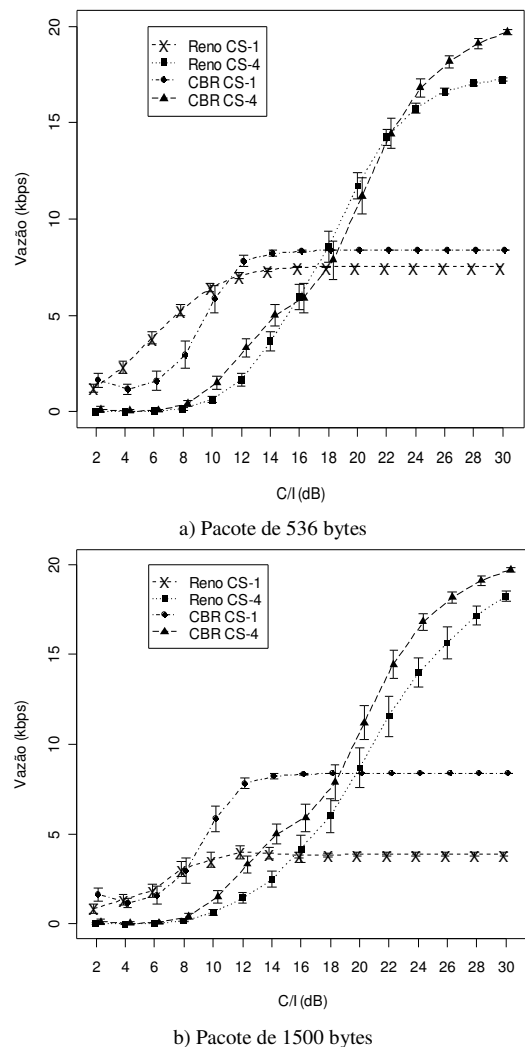


Fig. 8. Vazão do TCP Reno e do UDP utilizando CS-1 e CS-4

Novamente os resultados mostram que o tamanho do pacote influencia o desempenho do TCP sobre o GPRS. Ressalta-se que através do UDP, pôde-se validar o módulo GPRS utilizado

nos experimentos, pois as taxas de transmissão alcançadas foram bastante próximas das taxas apresentadas na Tabela I. Observando-se a Figura 8a, o CS-1 alcançou 8,38 Kbps para um valor máximo teórico de 9,05 Kbps, enquanto o CS-4 atingiu 19,7 Kbps, para um máximo de 21,4 Kbps. Em um ambiente ideal, (i.e., sem o modelo de erros) o CS-4 obtém uma taxa mais próxima da esperada (20,24 Kbps).

C. Modelo Adaptativo

Todos os experimentos realizados com CS fixo foram também realizados com o modelo adaptativo. A Figura 9 mostra a vazão alcançada para cada CS fixo (já apresentados na Figura 4) e a vazão obtida utilizando o modelo adaptativo. O CS-1 é o cenário de codificação inicial nos resultados deste gráfico. (Foram realizados experimentos iniciando em todos os CSs, que, no entanto, produziram resultados semelhantes).

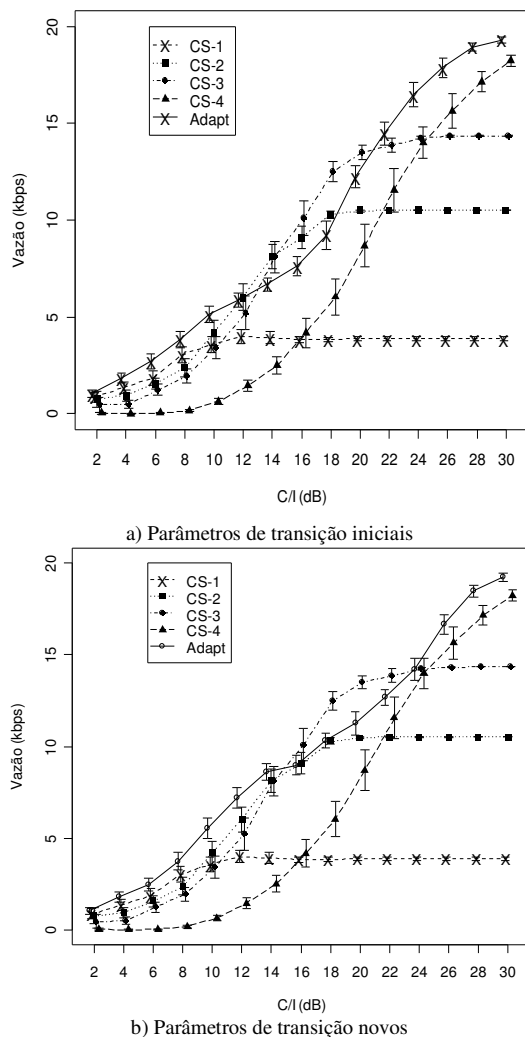


Fig. 9. Vazão do TCP utilizando o modelo adaptativo

Pode ser observado na Figura 9a que o modelo adaptativo consegue, na maioria dos casos, obter uma vazão maior do que a alcançada sem o modelo. Em, outras palavras, o modelo consegue fazer com que um cenário de codificação mais adequado ao ambiente seja utilizado, alcançando a máxima

vazão possível na maioria dos casos. Pode-se perceber também que o modelo não está devidamente ajustado, ou seja, os valores utilizados para a transição entre os cenários (Figura 2) não são adequados para o cenário utilizado. O modelo tende a migrar para os CSs extremos, i.e., CS-1 e CS-4 (ver Figura 9a). Nos casos em que o CS-2 ou o CS-3 apresentam melhor desempenho (entre 9 e 25 dB), o modelo escolhe o melhor entre o CS-1 e o CS-4.

Visando tornar o modelo adaptativo mais preciso (i.e., ajustá-lo de forma que ele migre sempre para o CS mais apropriado), novos parâmetros para os valores de transição foram testados (Figura 10). Estes foram obtidos a partir da observação do percentual de blocos descartados próximo aos pontos (C/Is) em que um cenário torna-se melhor que o anterior. Tais pontos são as extremidades dos intervalos em que cada CS alcança a maior vazão (Figura 4). Por exemplo, o CS-2 torna-se mais apropriado que o CS-1 aos 9dB. Então, foram coletados os percentuais de blocos descartados para 8 e 10 dB, utilizando CS-1 e CS-2. Para o CS-1, são descartados em média 8% dos blocos transmitidos aos 8 dB e 5% aos 10 dB. Assim, se estando no CS-1, o transmissor perder menos de 7% dos blocos transmitidos, o CS-2 pode ser utilizado. Da mesma forma, considerando que, para o CS-2 são perdidos em média 18% dos blocos aos 8 dB e 11% aos 10 dB, deve-se migrar para o CS-1 se o percentual de blocos perdidos for maior 12%.

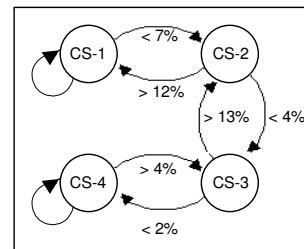


Fig. 10. Novos valores dos parâmetros de transição entre os CSs

Os resultados obtidos com os novos parâmetros podem ser visualizados na Figura 9b. Comparando-a com a Figura 9a, percebe-se que houve uma melhora nos intervalos em que o CS-2 e o CS-3 são mais apropriados (9 a 14 dB e 15 a 24 dB, respectivamente). Pode-se observar que no intervalo de 10 a 18 dB houve uma melhora na vazão, que, na maioria dos casos, ultrapassa a vazão máxima alcançada sem o modelo. Isso ocorre porque a tendência do modelo a migrar para CSs altos diminuiu. Contudo, percebe-se que entre 20 e 28 dB a vazão diminuiu, o que mostra quão difícil é encontrar valores ótimos para os parâmetros de transição. Em [18], é realizado um estudo sobre um algoritmo de gerenciamento de fila denominado RED (*Random Early Detection*). Esse algoritmo possui dois parâmetros cujos valores são essenciais para seu bom funcionamento. O estudo mostra que valores estáticos para esses parâmetros não conseguem ser considerados ótimos para todos os casos, e sim para determinados tipos de cenários. Analogamente, é demasiadamente difícil encontrar valores ótimos para os parâmetros de transição do modelo adaptativo que sirvam para todos os cenários. Modificando o cenário, os

valores que para o cenário atual são ótimos, tornam-se inadequados, precisando ser novamente ajustados.

D. Versões do TCP

O comportamento do TCP varia de acordo com os algoritmos do controle de congestionamento que implementa. Esses algoritmos são constantemente modificados para melhorarem o desempenho do TCP sobre as redes com as quais ele interage, gerando novas versões do TCP.

Os estudos realizados limitam-se a estudar o comportamento de apenas uma versão do TCP sobre redes GPRS, com exceção do [19], que analisa o SACK e o uso de *Timestamps*. Assim, este trabalho realiza um estudo do desempenho de quatro versões do TCP sobre o GPRS: Reno, New-Reno, SACK e Westwood. As versões Reno e New-Reno são estudadas por serem as mais utilizadas na Internet. SACK foi escolhido por apresentar um bom desempenho sobre redes sem fio em geral e, segundo [19], melhor performance em redes GPRS que o Reno. O TCP Westwood tem apresentado um ótimo desempenho em redes sem fio, sendo consideravelmente melhor do que outras versões do TCP, incluindo o TCP New-Reno, no qual o Westwood é baseado [20]. Contudo, nenhum estudo referente ao seu desempenho sobre redes GPRS foi ainda apresentado. Por isso, o interesse em estudá-lo neste trabalho.

A Figura 11 ilustra a vazão alcançada por uma estação móvel, considerando as quatro versões estudadas, para CS-1 e CS-4, utilizando a configuração 1+1 e pacotes (segmentos) de 1500 bytes.

Para o CS-1 (Figura 11a), pode-se notar que o SACK apresenta o melhor desempenho. As outras versões apresentam um desempenho semelhante. Para o CS-4 (Figura 11b), o New-Reno e o Westwood conseguem melhor desempenho que as versões SACK e Reno. Esses resultados são surpreendentes. Esperava-se que o SACK apresentasse sempre um desempenho melhor que o TCP Reno (no qual é baseado), como afirma [9], uma vez que sempre realiza menos retransmissões. Os experimentos realizados não forneceram informações suficientes para explicar esses resultados, sendo necessárias novas investigações.

Também era esperado que o TCP Westwood apresentasse uma melhora significativa em relação às outras versões. Contudo, além de, na maioria dos casos, conseguir uma vazão bastante próxima da alcançada pelo New-Reno (versão na qual o TCPW é baseado), em alguns casos, sua vazão é menor que a do New-Reno. Um provável motivo é a baixa taxa de transmissão do GPRS. O algoritmo que atualiza o *SSTHRESH* durante o *Fast Recovery* no Westwood é baseado na largura de banda disponível (estimada), como mostrado abaixo:

$$SSTHRESH = BWE * RTTmin / Seg_size, \quad (1)$$

onde *BWE* é a largura de banda estimada, *RTTmin* é o valor mínimo do *round-trip time* e *Seg_size* é o número de bits do segmento TCP. Quando esse valor calculado é menor que 2, *SSTHRESH* recebe o valor 2. A janela de congestionamento (*CWND*) também é baseada nesse valor:

se forem recebidos ACKs duplicados,

$$CWND = SSTHRESH \quad (2)$$

se expirar o *timeout*,

$$CWND = 1 \quad (3)$$

Considerando o melhor cenário utilizado neste estudo (CS-4 com configuração 4+1), a taxa de transmissão máxima possível para o GPRS é de 85,60 Kbps (sendo a alcançada pelo simulador um pouco menor). E, de acordo com a topologia apresentada na Figura 3, o RTT mínimo é de 80 ms (ida e volta em 4 links com atraso de 10 ms). Então, utilizando a fórmula (1), para o pacote de 1500 bytes, temos:

$$SSTHRESH = 85,60 * 80 / (1500 * 8) = 0,57$$

Como o valor calculado é menor que 2, o valor de *SSTHRESH* será 2. Assim, a janela de congestionamento não ultrapassará o valor 2 durante o *Fast Recovery*, limitando a taxa de transmissão. Para confirmar esses resultados, serão realizados novos estudos, com novos cenários.

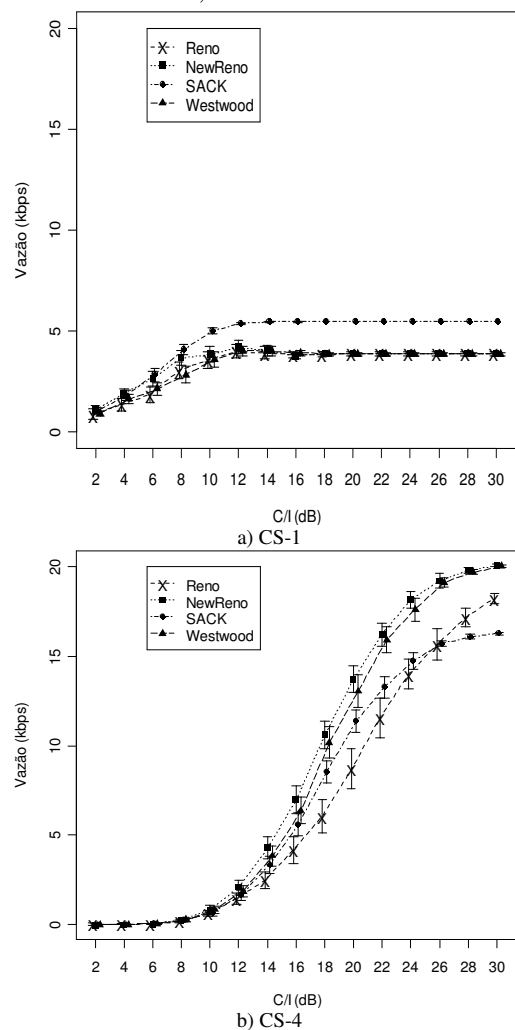


Fig. 11. Vazão do TCP Reno, New-Reno, SACK e Westwood

VI. CONCLUSÕES

Este trabalho apresentou um estudo do desempenho do TCP sobre o GPRS, um cenário comum de encapsulamento de protocolos em serviços celulares de dados 2.5G. Vários

parâmetros relacionados a TCP e GPRS foram utilizados, sendo a vazão a métrica reportada. Os resultados apresentados permitem obter faixas de valores de C/Is apropriados para o uso de cada cenário de codificação (CS), com o TCP Reno. Foi observado que os cenários mais adequados são o CS-2 e o CS-3 por abrangerem uma faixa maior de C/Is no qual apresentam melhor desempenho. Variando o tamanho do pacote (segmento), foi possível observar que quando o CS-1 é utilizado, o tamanho influencia na vazão, sendo esses valores inversamente proporcionais. Na comparação de resultados TCP com UDP, observa-se que, mais uma vez o tamanho do pacote influencia no desempenho do TCP sobre GPRS. Para o tamanho padrão (536 bytes), o TCP apresentou uma vazão próxima da alcançada pelo UDP, enquanto com pacotes de 1500 bytes, a vazão atingida utilizando o CS-1 é menor que a metade da obtida pelo UDP.

Um modelo adaptativo para os cenários de codificação, apresentado em [6], foi utilizado nos experimentos. Os resultados mostram que o modelo traz ganho na vazão, migrando para o CS mais adequado de acordo com as condições do ambiente (C/I). Também pôde-se perceber a dificuldade em se encontrar valores ótimos para os parâmetros de transição utilizados pelo modelo, uma vez que esses valores dependem do cenário estudado.

Por último, foi realizado um estudo da performance de quatro versões do TCP sobre o GPRS: Reno, New-Reno, SACK e Westwood. Os resultados mostram de forma surpreendente que, para o CS-1, o SACK apresenta melhor desempenho que as outras versões; contudo, nos outros cenários sua vazão é inferior. O Westwood, que consegue um ganho de até 500% na vazão em redes sem fio segundo [20], apresentou vazões semelhantes ao New-Reno, sendo pior em alguns casos. Esse baixo desempenho ocorre devido à baixa taxa de transmissão do GPRS, que não permite que a janela de congestionamento ultrapasse o valor 2 durante o *Fast Recovery*. Portanto, os resultados apresentados neste trabalho mostram que o Westwood não é uma versão apropriada para interagir com o GPRS.

Como trabalhos futuros, pretende-se realizar novos experimentos com o TCPW para confirmar os resultados obtidos e gerar um conhecimento abrangente sobre seu comportamento com o GPRS a fim de, possivelmente, torná-lo adequado a redes com baixas taxas de transmissão. Também serão realizados estudos sobre a combinação entre o uso do modo com confirmação e o CS utilizado na camada RLC, visando avaliar quando a presença de um torna o outro desnecessário e quando o uso de cada um é mais apropriado.

REFERÊNCIAS

- [1] GSM 03.60 version 7.4.1 Release 1998 "Digital Cellular Telecommunications System (Phase 2+); General Packet Radio Service (GPRS); Service Description; Stage 2.", ETSI, Oct. 2000.
- [2] M. Meyer, "TCP performance over GPRS", In: Proceedings IEEE Wireless Communications and Networking Conference (WCNC), Sep. 1999.
- [3] R. Kalden, I. Meirick, and M. Meyer, "Wireless Internet Access based on GPRS", IEEE Personal Communications, 7(2), Apr. 2000.
- [4] R. Chakravorty, J. Cartwright and I. Pratt, "Practical Experience with TCP over GPRS", In: Proceedings of the IEEE Global Communications Conference (IEEE GLOBECOM 2002), Taipei, Taiwan, Nov. 2002.
- [5] C. Bettstetter, H.-J. Vogel, and J. Eberspacher, "GSM Phase 2+; General Packet Radio Service GPRS: Architecture, Protocols and Air Interface", IEEE Communications Surveys, 2(3), 1999.
- [6] S. Dantas, "Avaliação de Serviços de Dados em Redes GPRS", Dissertação de Mestrado, Centro de Informática, UFPE, 2001.
- [7] J. Postel, "Transmission Control Protocol", RFC-793, USC/Information Sciences Institute, Sep. 1981.
- [8] M. Allman, V. Paxson and W. Stevens, "TCP Congestion Control", RFC 2581, Apr. 1999.
- [9] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP", ACM Computer Communications Review", vol. 26, no.3, Jul. 1996. pp.5-21.
- [10] J. Hoe, "Improving the Start-up Behavior of a Congestion Control Scheme for TCP", In: Proc. ACM SIGCOMM, Aug. 1996.
- [11] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement options", RFC-2018, Oct. 1996.
- [12] A. Bakre, and B. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts". In: Proc. 15th International Conf. on Distributed Computing Systems (ICDCS), May 1995.
- [13] H. Balakrishnan, S. Seshan, and R. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", ACM Wireless Networks, 1(4), Dec. 1995.
- [14] C. Casetti, M. Gerla, S. Mascolo, M. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", In: Proceedings of ACM Mobicom, Jul. 2001.
- [15] ns-2 "The Network Simulator", <http://www.isi.edu/nsnam/ns>
- [16] R. Jain, "GPRS Simulations using ns-Network Simulator", MSc thesis, Department of Electrical Engineering, Indian Institute of Technology – Bombay, Jun. 2001.
- [17] GSM 04.60 version 8.5.0 Release 1999. "Digital cellular telecommunications system (Phase 2+) (GSM); General Packet Radio Service (GPRS); Mobile Station (MS) - Base Station System (BSS) interface; Radio Link Control/Medium Access Control (RLC/MAC) protocol", ETSI, Oct. 2000.
- [18] M. May, J. Bolot, C. Diot, and B. Lyles, "Reasons not to deploy RED", In: Proc. IWQoS '99, mar. 1999. pp.260-262.
- [19] J. Rendón, F. Casadevall, and J.L. Faner, "Wireline TCP Options Behavior in the GPRS Network", In: Proc. IST Mobile Communications Summit, Sitges-Barcelona, Spain, Sep. 2001.
- [20] R. Wang, M. Valla, M.Y. Sanadidi, M. Gerla, "Adaptive Bandwidth Share Estimation in TCP Westwood", In: Proc. IEEE Globecom, Nov. 2002.